EVALUATING SINGING FOR COMPUTER INPUT USING PITCH, INTERVAL, AND MELODY

by

GRAEME ZINCK

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2022

AUTHOR'S DECLARATION

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

STATEMENT OF CONTRIBUTIONS

This thesis is adapted from a manuscript written for publication [38]. The research was conducted at the University of Waterloo by Graeme Zinck under the supervision of Dr. Daniel Vogel. Graeme Zinck conducted the reported research and drafted the manuscript. Daniel Vogel provided feedback on the research and manuscript drafts.

ABSTRACT

In voice-based interfaces, non-verbal features represent a simple and underutilized design space for hands-free, language-agnostic interactions. This work evaluates the performance of three fundamental types of voice-based musical interactions: pitch, interval, and melody. These interactions involve singing or humming a sequence of one or more notes. A 21-person study evaluates the feasibility and enjoyability of these interactions. The top performing participants were able to perform all interactions reasonably quickly (<5s) with average error rates between 1.3% and 8.6% after training. Others improved with training but still had error rates as high as 46% for pitch and melody interactions. The majority of participants found all tasks enjoyable. Using these results, we propose design considerations for using singing interactions as well as potential use cases for both standard computers and augmented reality glasses.

ACKNOWLEDGMENTS

This work made possible by NSERC Discovery Grant 2018-05187, Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces," and Ontario Early Researcher Award ER16-12-184.

DEDICATION

To those who gave their time to sing in the name of science.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

If I cannot fly, let me sing.

– Stephen Sondheim

1 INTRODUCTION

With the proliferation of voice-activated devices, people are increasingly using their voice to perform hands-free interactions: as of 2018, 27% of the global online population used voice search on mobile devices [15]. Most of these systems use verbal interactions that translate spoken words and phrases into commands. On the other hand, non-verbal vocal interactions provide opportunities for language-independent, simple, continuous controls [32]. Instead of recognizing words, these techniques detect properties of vocal sound, such as vowel sounds or pitch, making them simple enough to process locally on consumer hardware in real time. Despite its simplicity, much less work has explored the non-verbal vocal input design space.

Past research has evaluated how we can use various vocal features, such as blowing puffs of air [6, 37], hissing [25], producing vowel sounds [9, 11], and singing or humming pitches. Within this last category, past work has used pitch for continuous control by sliding pitch up and down [18, 31] or discrete control by assigning portions of the frequency space to different functionality [5, 24, 26, 29]. In these systems, users can either hum or produce sustained sounds like "la" and "do." However, most work does not leverage musical concepts like semitones and intervals to expand these vocal interactions, and those that do have not performed formal evaluations.

Unlike past work in non-verbal vocal interactions, this work leverages musical concepts to design interactions that make computer use more musical and potentially more enjoyable. The presented techniques are versatile: by using pitches from the major scale, a larger number of distinct interaction variations are possible. Furthermore, frequently using these interactions could help users improve their vocal ability. We envision using singing interactions as a way to replace or supplement current interface methods for functions like command shortcuts, mode switching, and parameter control by singing melodies from popular music. These interactions could also provide a hands-free input method for augmented reality glasses. Before considering such applications, this work first evaluates the feasibility of fundamental types of singing interactions.

This work explores pitch, interval, and melody interactions, corresponding to sequences of one, two, and three distinct musical notes, respectively (Fig. 1.1). A 21-person study evaluates their feasibility and enjoyability for participants who self-identified as singers ($n = 10$) and non-singers ($n = 11$). For each technique, participants were prompted to sing or hum 7 possible sequences of notes, both with and without background music. We hypothesized that background music would improve performance by providing a reference pitch for the notes to sing. The findings demonstrate that top performers are able to perform all interaction types sufficiently fast (3.5–5s) with low error rates (<10%) after training. The lowest performers had comparable

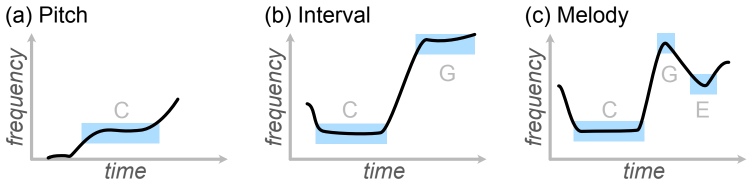


Figure 1.1: Examples of fundamental singing interaction techniques, visualized as vocal frequency over time: (a) pitch is a single note; (b) interval is a sequence of two notes; and (c) melody is a sequence of three notes.

speeds (4.5–5.5s), but error rates as high as 46% after training. Of the three techniques, pitch interactions are the easiest to learn and interval interactions are the hardest to master. Participants found background music helpful in all tasks, despite having a limited effect on measured performance. Overall, the majority of participants found all interactions enjoyable.

Our work makes three contributions: (1) experimental results validating the effectiveness of pitch, interval, and melody interactions as a form of computer input; (2) an evaluation of how background music impacts performance; and (3) a set of design considerations and potential use cases that leverage these interactions.

2 RELATED WORK

Pitch, interval, and melody interactions are types of non-verbal vocal interactions: they facilitate vocal control without the use of words. This chapter provides an overview of recent and current non-verbal vocal input mechanisms, with a focus on those incorporating vocal pitch.

2.1 NON-VERBAL VOCAL INTERACTIONS

The simplest non-verbal vocal interactions are binary in nature. For example, with Pufftext [6], users blow into a microphone to select characters on a hands-free spinning keyboard for mobile phones. Similarly, with Blowclick [37], users blow into a microphone to make selections with low latency. Poláček et al. [25] leverage hissing to select characters on a virtual keyboard. These on/off interactions enable fast, precise, hands-free control, but they are limited: the only degree of freedom is the duration of the interaction.

Other techniques have more degrees of freedom by interpreting a wider range of vocal sounds. In particular, some systems map vowel sounds like "a," "e," "i," "o," "u," and their intermediaries to a single continuous parameter. For example, Vocal Joystick [8] is a voice-controlled mouse that moves based on the shape of a vowel, as classified by a multi-layer perceptron. A small Fitts' Law study with four expert users showed movement time was comparable to using a joystick, but much slower than using a mouse. The same strategy has been applied to drawing applications with VoiceDraw [10] and video games with Harada et al.'s Pacman [11]. Unlike our work, these techniques attempt to replace traditional inputs with more accessible non-verbal vocal inputs.

Some work has considered how non-verbal commands can work in tandem with other interactions for general computing. Igarashi and Hughes [14] use verbal commands followed by vowel sounds to support more precise control of various parameters. Sakamoto et al. [27] augment touch input with vowel sounds to allow continuous control of scrolling and zooming. VoicePen [9] augments pen input with vowel sounds to configure drawing parameters in an enjoyable manner. In a similar manner, we envision our singing interactions as augmenting standard user input.

2.2 PITCH FOR CONTINUOUS CONTROL

Our work focuses on non-verbal vocal interactions that leverage pitch. While these typically still involve producing vowel sounds, the interaction is independent of the vowel used. Instead, the fundamental frequency of the voice determines the interaction. Such interactions can be mapped to a single continuous parameter, in which users can sing any pitch to continuously adjust the parameter, or multiple discrete parameters, in which the sound frequency space is partitioned into ranges that trigger different functionality. Unlike the detection of vowel sounds, continuous pitch-based interactions intuitively map to relative movement control by singing higher or lower. Voodle [18] uses pitch and rhythm to control a one-dimensional robot, and VoiceBot [12] uses pitch and vowels to control a robotic arm. Similarly, Sporka et al. [31] use pitch to determine movement direction of a mouse pointer. Peixoto et al. [22] use pitch for smooth control of wheelchair speed. These works have evaluated application-specific task performance, but the results are not generalizable. Furthermore, in all cases, pitch is only mapped to a single parameter. Our work discretizes the spectrum of pitches to facilitate a wider variety of functionality.

2.3 PITCH FOR DISCRETE CONTROL

Instead of directly using the continuous frequency space, discrete pitch-based interactions divide it into two or more ranges that map to different parameters. Some techniques use a single threshold pitch to determine when the voice is above, below, or crossing the threshold. Chanjaradwichai et al. [5] detect pitches above and below a threshold to select one of four cells in a grid. A small 6-person study indicated that short pitch-based vocalizations were faster than using speech commands and had error rates around 12%. Poláček and Míkovec [24] use hummed commands above and below a threshold for mouse clicks, finding the approach faster but more error-prone than speech commands (6% vs. 3.5%). This approach has also been adapted to the more complex context of text entry. Humsher [26] detects frequencies above and below a threshold to select characters for text entry and CHANTI [30] uses similar interactions with an ambiguous keyboard, where each keyboard key is associated with multiple possible letters. In all cases, a single threshold pitch still restricts control to a small number of parameters.

Other work has divided the frequency space with multiple threshold pitches, enabling control of more parameters. For instance, Sporka et al. [28] distinguish between four pitches for keyboard input, where each pitch sung divides the number of possible letters by 4 until only one possibility remains. More related is the work of Hämäläinen et al. [13], which uses many threshold pitches for music education video games. The pitch thresholds align with the twelve semitones of Western music, making this interaction technique more musical. However, their system is designed for showing visual pitch feedback when singing songs instead of providing an interaction mechanism for general use. They do not perform any formal evaluation of their technique.

Sporka [29] evaluates how effectively people can sing discrete pitches in general. The author compares performance when there are different sized ranges of pitches to sing, using ranges two, four, and eight times larger than a baseline size of one semitone. The findings demonstrate that non-singers have a lower error rate when the range is four times larger than the baseline, with error rates of 23% and 52%, respectively. Our work differs in four ways. First, the pitch recognition method is different: instead of determining pitch based on the last frequency detected, which might slide off the desired pitch, our technique determines it by holding a pitch for 200ms. Second, our experiment considers sequences of one, two, and three notes instead of just one. Third, our analysis explores in what ways background music can affect performance, which was not considered in any past work. Finally, our interactions are not arbitrary: they are designed to be musically appealing and harmonious with background music.

3 MUSIC BACKGROUND

Since our interactions involve singing, this chapter provides background on singing abilities and relevant music theory.

3.1 ACCOMMODATING USER SINGING ABILITIES

An important consideration for our work is how people perceive pitch. People with *absolute pitch* (also referred to as perfect pitch) are able to identify and produce musical tones without needing an external reference pitch. However, this is a rare ability, estimated at less than 0.01% of the population [33]. Most people process pitches they hear and sing relative to other pitches. Providing a tonic note as a reference point can help people identify and produce relative pitches. Unlike past work in pitch-based non-verbal vocal interactions, we use background music to provide this reference.

Another consideration is how easily people match pitch. Amir et al. [1] measured the ability for musicians and non-musicians to reproduce pitches. While musicians could match pitches within 0.5 semitones, non-musicians were within 1.3 semitones. In practical terms, this means our techniques need to accommodate users that sing the wrong pitch before finding the correct one.

Finally, visual feedback has been shown to improve pitch accuracy. Using a visualization that highlights keys on a virtual piano, Wilson et al. [35] demonstrate that while visualizations increase cognitive load during use, participants improve their pitch accuracy afterwards. Similar types of feedback have been used in classroom settings. Welch et al. [34] use a simple frequency-over-time graph to improve pitch matching abilities for seven-year-olds. Similarly, Callaghan et al. [4] use a 2D graph of both frequency and volume over time to provide immediate feedback for pitch accuracy and vibrato. Because this feedback is visual, it should help people with good pitch, poor pitch, and potentially deafness sing in tune. Our needs for visual feedback are different. Specifically, users need to see the pitches being sung and interactions being recognized in a single interface. Furthermore, to use singing as an input modality, the visualization needs to be a simple overlay on top of existing interfaces.

3.2 MUSICAL TERMINOLOGY AND NOTATION

Since we define our interactions using standard concepts in music theory, this section provides an overview and a set of formalizations for common musical terminology. The *pitch* of a sound can be quantified using the fundamental frequency, $f_0 \in \mathbb{R}^+$, of its waveform. A *note* is a sound that has some associated pitch. Following the Musical Instrument Digital Interface specification [2], a note with a fundamental frequency f_0 is assigned an integer *note number*, $p = 69 + 12 \times \log_2(\frac{f_0}{440})$. For reference, A4 concert pitch has $f_0 = 440$ and note number 69. For a note with note number p, a *semitone* is the difference between p and $p+1$ and an *octave* is the difference between p and $p+12$.

For the purposes of this work, a *scale* is a vector of notes ordered by increasing pitch, independent of the octave. A scale is represented by an ordered pair $(t, \mathbf{s})$, where note number t $(0 \leq t < 12)$ is the *tonic*, or the first note in the scale, and $\mathbf{s}$ is a vector representing the notes in the scale relative to t. Each s_i $(0 \leq s_i < 12)$ represents the note s_i semitones above the tonic t. For a given scale $(t, \mathbf{s})$, the *degree d* of a note p is its index in the scale, where $1 \leq d \leq dim(\mathbf{s})$ and $s_{d-1} = p - t \pmod{12}$.

This work uses the major scale, $(t, \mathbf{s})$, for any tonic t and $\mathbf{s} = \langle 0, 2, 4, 5, 7, 9, 11 \rangle$. This is because a large portion of Western music is based on this scale, making the pitches pleasing and familiar for the average person. Because the major scale has 7 notes, there are effectively only $dim(\mathbf{s}) = 7$ distinct pitches that are used in our interactions.

When discussing interval and melody interactions, we sometimes use the pitch one octave above the first pitch sung, which is always the tonic in our case. We abuse the standard definition of degree to refer to the pitch one octave above the first pitch sung as degree $d = 8$.

Our work assumes users have a vocal range of one octave, which is reasonable for both singers and non-singers [29]. However, different people have different ranges: some can only sing high pitches while others can only sing low pitches. To accommodate different vocal ranges, the techniques require a calibration step to determine the appropriate tonic t for a given person.

For the remainder of this paper, numeric values for a note or pitch refer to its degree as opposed to its note number.

4 TECHNIQUE

This chapter presents three types of singing interactions: pitch, interval, and melody. In each, users can hum or sing on any vowel, such as "la," "tee," or "do."

4.1 SINGLE PITCH INTERACTIONS

The simplest type of singing interaction involves a single note. Our technique first involves detecting pitch by determining the f_0 of microphone input every 50ms using a fast Fourier transform with size 2048 and processing it using the McLeod Pitch Method [19]. As suggested by the method, if a frequency is below a minimum clarity of 95%, it is ignored to reduce noise. Each time a frequency is detected and not ignored, the current *time step* increments by 1. The frequency at the i-th time step is f_i. Note that time steps are at least 50ms apart, but gaps may be longer since unclear frequencies are ignored.

To ensure only intentional interactions are recognized, a user must sing and hold a note for at least 4 consecutive time steps ($\geq$ 200ms). If the user sings a different note and holds for 4 time steps, the new pitch is recognized and the old one is discarded. The recognized pitch is confirmed after a silence of 500ms. We informed this design with a series of pilot tests to find the best trade-off between facilitating quick interactions and ignoring accidental ones. An important benefit of this design is that users can correct their pitch during an interaction: they can start by singing one note, then slide into the desired note. This is crucial for usability since both musicians and non-musicians can have difficulty singing pitches precisely [20]. Our design also allows users to slide off pitch at the end of an interaction without penalty, improving on the pitch recognizer used by Sporka [29].

To compensate for variable audio quality, the technique smooths the detected frequencies using a recursive definition: $f_1^s = f_1$ and $f_i^s = \frac{f_i}{2} + \frac{f_{i-1}^s}{2}$. This can add an additional 50–100ms latency, but it improved performance in our pilot tests. Also, poor audio quality can sometimes result in large, sudden changes of pitch. To resolve this problem, if a frequency f_i is more than an octave away from f_{i-1}, it is ignored unless sustained for 3 time steps.

Since we assume users have a one-octave range, our technique is restricted to 12 possible semitones in Western music. However, since we are especially interested in interactions that sound harmonious with background music, we only evaluate interactions that use the 7 degrees in the major scale. Thus, there are $n = 7$ single pitch interactions.

An important part of our technique is a circular visualization for helping users find their desired pitch. We had four design objectives for our visualization. First, we wanted to show which notes are valid. Second, we needed to show which pitch was being sung, providing a feedback loop for users to improve their pitch. Third, we needed a consistent way to show which interaction was recognized, regardless of which technique was used. Finally, we wanted to illustrate how the pitches "wrap around," by our definition of pitch degree. Existing approaches in non-verbal vocal interfaces were insufficient for our purposes. In particular, Sporka's [29] visualization, with a 1D horizontal line of frequencies and a vertical line indicating the current pitch, does not satisfy the latter two objectives. Furthermore, existing pitch visualizations for music education [4, 34, 35] are not designed to show interactions being recognized.

For this reason, we designed a visualization inspired by circular pitch pipes (Fig. 4.1). The circular design illustrates how pitches "wrap around" by displaying all recognizable notes along the circumference of a circle. An arc outside the circle indicates the currently detected pitch. Once a pitch is recognized, the outer arc turns blue and a thick blue arc appears inside the circle indicating which pitch is recognized. This also indicates when the user can stop singing. The visualization is easily extensible to interval and melody interactions by making the inner arc span across all of the recognized pitches.

4.2 interval interactions

Interval interactions entail singing two notes in sequence. Users sing one note for at least 4 time steps ($\geq$ 200ms), change to another note for at least 3 time steps ($\geq$ 150ms), and fall silent for 500ms. To keep this interaction easy to perform during our study, we consider only intervals starting at the tonic t and ascending to an end pitch with degree $1 < d_e \leq 8$.

In total, our technique encompasses $n = 7$ interval interactions. In the future, this approach is easily extensible to have $7 \times 7 = 49$ possible interval interactions within a one-octave range, with 7 possible start degrees $1 \leq d_s \leq 7$, and 7 possible end degrees $1 \leq d_e \leq 8, d_s \neq d_e$. Note that $d_s \neq 8$ by our definition of degree 8.

The interval visualization is identical to pitch, except once the second note is recognized, the inside arc spans from the first note d_s to the second note d_e.

4.3 melody interactions

Melody interactions go one step further with three or more notes in sequence. Users first sing a note for at least 4 time steps ($\geq$ 200ms), then sing at least two more notes for any duration, subsequently falling silent for 500ms. Inspired by Lin et al.'s system for evaluating people's ability to accurately sing songs [17], we use dynamic time warping (DTW) to determine the cost of warping a sequence to match each of the possible melodies. The melody with the lowest cost at the end of the interaction is the recognized melody. Because DTW does not consider the timing of a melody, it is possible to have longer melodies without holding onto pitches for 150ms each, unlike our interval interaction

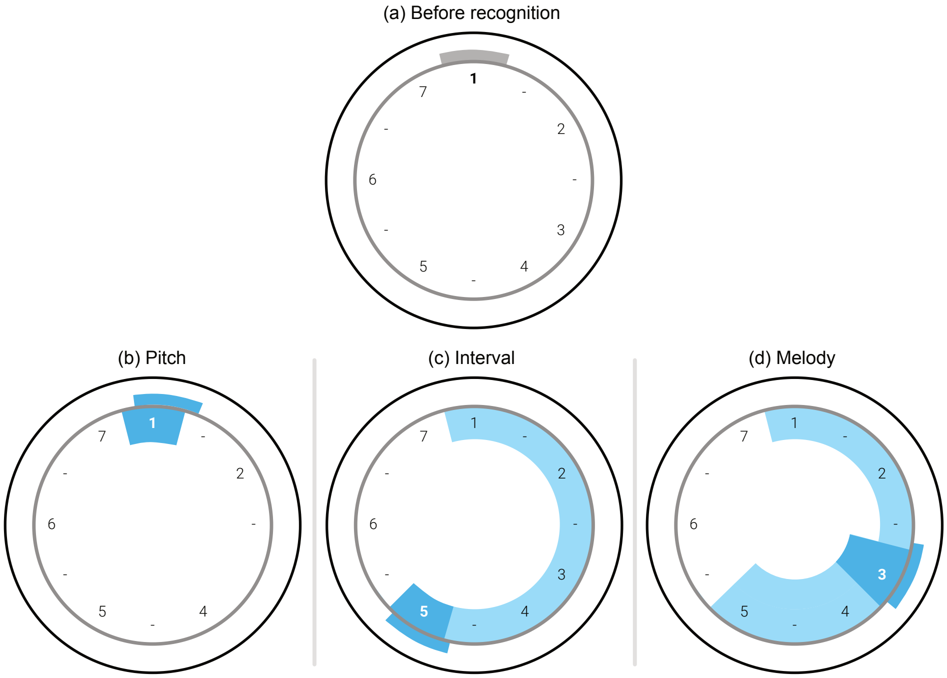


Figure 4.1: Our pitch visualization is a circle divided into 12 semitones, with the 7 degrees of the major scale labelled as 1 to 7: (a) before an interaction is recognized, the gray arc indicates the frequency detected and the bold "1" indicates degree 1 is detected but not yet recognized; (b) for pitch interactions, the outer arc turns blue and a thick inner arc appears when degree 1 is recognized; (c) for interval interactions, sliding vocal pitch to degree 5 moves the outer arc around the circle and creates a trail from the start degree "1" to the end degree "5"; (d) for melody interactions, sliding vocal pitch to degree 5 and then down to degree 3 moves the outside arc around the circle and shows the recognized melody using two inner light blue arcs from degree "1" to "5" and "5" to "3."

technique. It also means a user does not need to accurately sing the second and third notes. As long as they briefly get close to the appropriate pitch, DTW can typically determine the intended melody.

For simplicity and tractability, our study involves $n = 7$ three-note melodies (Table 4.1), all of which start at the tonic note, use only notes in the major scale, and stay within a one-octave vocal range. We designed the melodies to be relatively easy to sing by typically focusing on degrees 1, 3, and 5 (the degrees in a major chord) or moving by one degree at a time. The number of melodies is theoretically unbounded: they can have different start notes and unbounded length. In practice, this is limited by the number that one can accurately and quickly distinguish using DTW.

An alternative approach to our technique might allow users to sing any initial note followed by a sequence relative to that note. This has one fewer degree of freedom, but it is likely easier to perform. We chose to specify a specific initial note for two reasons. First, it maintains consistency with the

Table 4.1: Melodies evaluated by our experiment.

Melodies (sequences of notes)			
$1-3-2$	$1-5-1$	$1-5-3$	$1-5-8$
$1-2-3$	$1-4-1$	$1-3-5$	

other two techniques, making it possible to leverage all techniques in a single interface. Second, it ensures that melodies will use the same scale as the background music. Choosing a different initial note could cause the melody to clash with the notes in the background, thereby making the tasks more challenging. Thus, for this study, all melodies start with the tonic note.

Our visualization for melodies is similar to that of intervals. The difference is that there are two arcs inside the circle showing the melody detected: one from the first to the second note and one from the second to the third note in the melody.

All three types of singing interactions could be used in a single interface. For instance, one could sing the interval $1-5$, then melody $3-5-3$, followed by pitch 4. The requirement for these combinations is that each starting pitch d_s can only be mapped to one type of interaction. In this example, 1 is mapped to interval interactions, 3 is mapped to melody, and 4 is mapped to pitch. Our exploration focuses on each technique individually, but an application could incorporate all of them.

4.4 USING BACKGROUND MUSIC

A novel part of our interaction technique is in its use of background music. The music provides users with a reference pitch, which should make it easier to find the desired notes. Additionally, singing along with music has the potential to increase enjoyability. For the experiments, some conditions play a single 30 second loop of solo piano music, transposed into a specific user's vocal range using their tonic t. The background music uses a single tonic chord, emphasizes the tonic note, and maintains a constant 90 bpm. Using different music has the potential to impact performance, but comparing audio stimuli is outside of the scope of this study. We maintain internal validity by using the same simple background loop for all experiments.

5 EXPERIMENTAL DESIGN

For each interaction technique (pitch, interval, and melody), we performed an experiment to evaluate its feasibility and enjoyability. Because each technique builds on the previous one by adding additional notes, we always ran the pitch experiment first, followed by the interval and melody experiments. Given that singing accuracy decreases as the number of pitches increases for many tasks [23], this ordering should have increasing difficulty. Nevertheless, there is likely a learning effect between the three experiments. Our analysis takes this into consideration. This chapter outlines the shared experimental design. An accompanying video also demonstrates the different experiment tasks.

5.1 PARTICIPANTS

We recruited 25 participants, of which 4 were excluded due to technical issues with their audio setup. These 4 excluded participants self-reported technical problems, and we manually verified logs and audio recordings to confirm. The remaining 21 included participants averaged 25.8 years of age (min: 20, max: 57), of which 8 were female, 11 were male, and 2 were non-binary. Participants were recruited using email, social media, and an HCI course discussion board, receiving $20 for successful completion of the study. No participants had any experience with our system prior to the experiments.

When asked to rate their agreement with the statement, "I consider myself a good singer," 10 participants responded with "Agree" or "Strongly agree" and 11 participants responded with "Slightly agree" or lower. We used this as a guide to recruit participants with different musical ability.

5.2 APPARATUS

The study was conducted online through a React web application.[1] All participants were required to use Google Chrome on a laptop or desktop with a microphone and headphones. Participants were instructed to perform the tasks in a quiet room to ensure accuracy of the pitch detection. Ambient noise was recorded and analysed periodically throughout the experiment to ensure the levels were sufficiently low.

To ensure each participant's microphone input worked sufficiently well for the experiment tasks, participants completed an audio test. The laptop interface instructed participants to open a custom phone web app on a separate device. Then, participants tapped buttons indicated on screen to

[1] Experiment source code is available at https://github.com/exii-uw/evaluating-singing-input

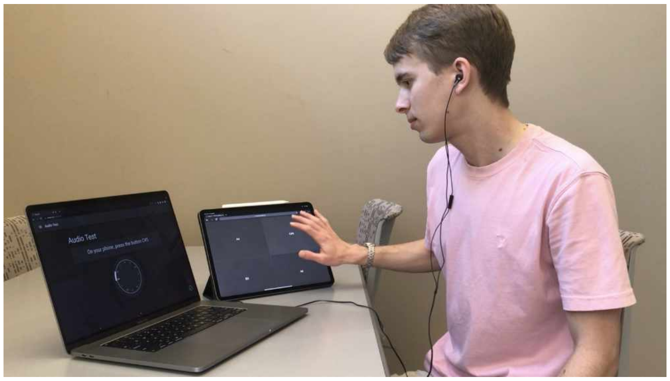


Figure 5.1: Setup for the audio test. The laptop interface instructs participants to tap buttons on a separate phone or tablet to test the laptop's microphone input.

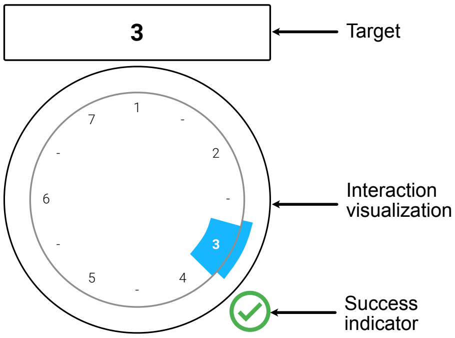


Figure 5.2: Illustration of the experiment task interface, showing an example for pitch tasks with target 3 while a participant is singing note 3.

produce tones and verify that their laptop could recognize the correct pitches (Fig. 5.1). This allowed us to verify that the audio system could detect pitches properly without excluding participants with poor pitch matching abilities from the study.

5.3 TASKS

Each of the three experiments prompted participants to perform tasks related to the interaction techniques explained in Chapter 4. Participants pressed a button to start the first trial in each block. Then, they were shown an interface with the interaction visualization at the bottom and the target above with the notes to sing (Fig. 5.2 and 5.3). Singing for at least 200ms and falling silent for 500ms ended the trial. During training, a green check mark or red x-mark appeared after each trial to indicate success or failure. During each task, the f_0 of microphone input was logged at every time step for post-hoc analysis.

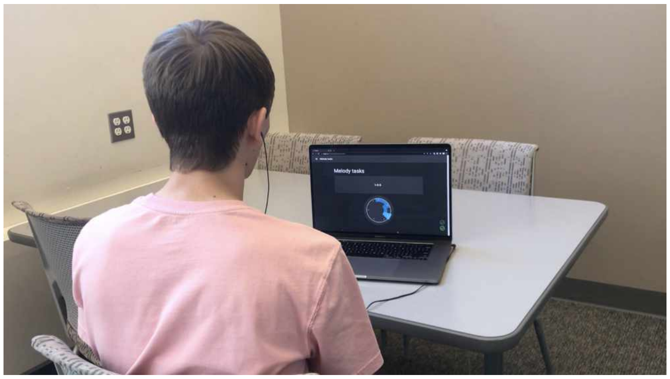


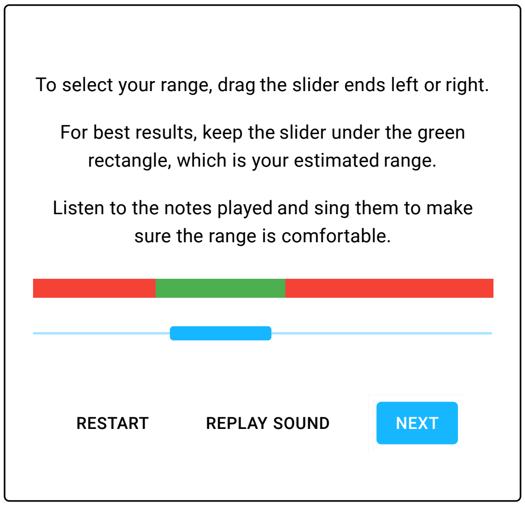
Figure 5.3: Experiment setup when a participant performs melody tasks.

5.4 PROCEDURE

Before starting the experiments, we asked participants questions regarding their musical and technical experience, including: "How many years of private musical lessons have you had," "How many years of group musical lessons have you had," "How many years of vocal training have you had," "How many years of ear training have you had," and "How many hours per day do you use a laptop or desktop?" We also asked participants, "How often do you sing?" on a scale from 1 ("rarely") to 7 ("multiple times a day").

After watching a short tutorial video, participants were asked to sing one low note and one high note to determine vocal range and the tonic *t* for the interactions. Then, they were shown a slider to fine-tune their detected vocal range (Fig. 5.4). Moving the slider played back an audio clip with the lowest and highest notes in the selected range. A green area above the slider indicated what the ideal range should be and deviating from the expected range caused a warning to appear. However, participants maintained full control over their selection, which was deemed important through pilot tests. Following this, the three experiments began.

For each technique, participants performed an experiment with three stages: pre-test (Blocks 1–2), training (Blocks 3–6), and post-test (Blocks 7–8) (Fig. 5.5). Each test stage measured performance with and without background music: the pre-test measured untrained performance and the post-test measured trained performance. The training stage differed from the test stages in three ways: it allowed two attempts per task; it used audio prompts, which were 5s recordings of a piano playing the notes to sing; and it always used background music to provide a reference pitch. To facilitate learning early in the training stage, Block 3 played audio prompts before every attempt, whereas Blocks 4–6 played audio prompts only after failing the first attempt. All blocks had a random ordering of task variations except Block 3, which presented each variation twice in a row.



To select your range, drag the slider ends left or right.

For best results, keep the slider under the green rectangle, which is your estimated range.

Listen to the notes played and sing them to make sure the range is comfortable.

Figure 5.4: The interface allowing participants to fine-tune their vocal range. After moving the slider, participants could restart the calibration or confirm their vocal range.

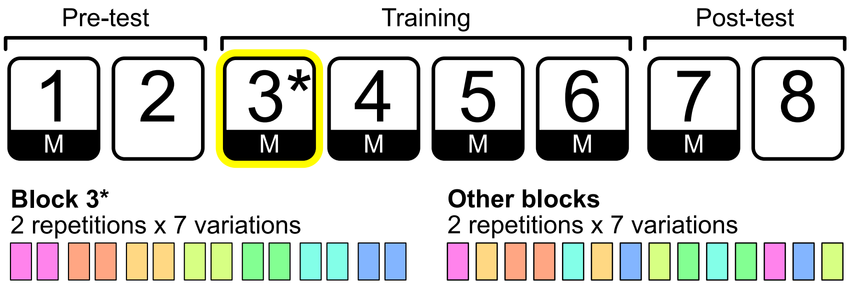


Figure 5.5: Example of the block structure for a participant. M indicates there is background music for the block. Colours indicate different task variations.

5.5 design

Each experiment (pitch, interval, and melody) has a within-subjects design with two primary independent variables: training with 2 levels (untrained for pre-test blocks, trained for post-test blocks) and background with 2 levels (music, no-music). For analysis, we classify participants based on their error rates across the three experiments. This classification is a between-subjects variable: skill with 3 levels (novice, intermediate, expert).

There were 7 task variations for each technique, representing the pitch, interval, or melody to sing. Each block had 2 repetitions × 7 variations. In each of the pre- and post-test stages, there was 1 block × 2 backgrounds, where background conditions were counterbalanced. In the training stage, all 4 blocks had Music. The training blocks are solely for training purposes, so our analysis relies solely on the data from blocks in the two test stages.

The primary measures computed from logs are *Total Time*, *Sing Time*, and *Error Rate*. *Total Time* is the time from the end of the previous trial until an interaction is completed. *Sing Time* is the time from the first detected pitch until the last detected pitch in a trial. *Error Rate* is the proportion of trials in a block that ended with an error.

In addition, a questionnaire after each experiment provides 6 NASA-TLX metrics for measuring task workload [21] and 2 subjective measures for measuring enjoyability and perception of background music. All measures are rated on a 7-point numeric scale for consistency. Participants could provide feedback in a free-form text box at the end of each experiment.

In summary: 4 blocks × 7 variations × 2 repetitions = 56 data points per participant, for each of the 3 experiments.

6 RESULTS

In all experiments, performance varied widely between participants. To meaningfully evaluate the data, we cluster the participants into groups of low- and high-performing individuals. We hypothesized performance would be strongly correlated with participant musical experience. However, using all measures of musical experience we collected, some musical participants performed poorly in the study, and similarly, some non-musical participants performed very well. Two factors likely contributed to this result. First, we relied on self-reported metrics as opposed to more objective musical evaluations. Second, using singing for computer input requires a moderate level of comfort with technology.

For analysis purposes, we instead use k-means to cluster participants into three SKILL levels based on their error rate across all three experiments. Since clustering was performed across all experiments, these SKILL levels represent three user groups with varying pitch control and comfort with technology. We use these clusters to explore the wide variation in performance and better understand the contexts in which our techniques are feasible.

The resulting k-means SKILL levels include EXPERT (8.8% error rate, $n = 8$), INTERMEDIATE (30.2% error rate, $n = 6$), and NOVICE (54.6% error rate, $n = 7$). Using the non-parametric Spearman correlation method, SKILL was correlated with years of private music lessons ($r_s = .48$, $p = .029$) and years of group music lessons ($r_s = .45$, $p = .043$), but interestingly, not years of music ear training ($p = .098$), years of vocal training ($p = .298$), participant perception of singing ability ($p = .195$), frequency of day-to-day informal singing ($p = .19$), or hours per day using laptops and desktops ($p = .563$). This indicates that while musical experience predicts performance to an extent, singing experience is not required or sufficient to perform well.

In the analysis to follow, a SKILL × TRAINING × BACKGROUND ANOVA with Tukey HSD post hoc tests was used, unless noted otherwise. When the assumption of sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). According to the Shapiro-Wilk Normality test, none of the residuals for collected data were normally distributed, so Box-Cox or ART-transformed [36] values were used for statistical analysis. For each measure, trials were aggregated by participant and factors being analysed.

6.1 PITCH INTERACTIONS

For every participant ($n = 21$), trial times more than 3 standard deviations from the mean time were excluded as outliers. Before removal, outliers skewed trial times longer, mostly for novices and intermediates before training. Outliers had minimal effect on post-training performance and on expert

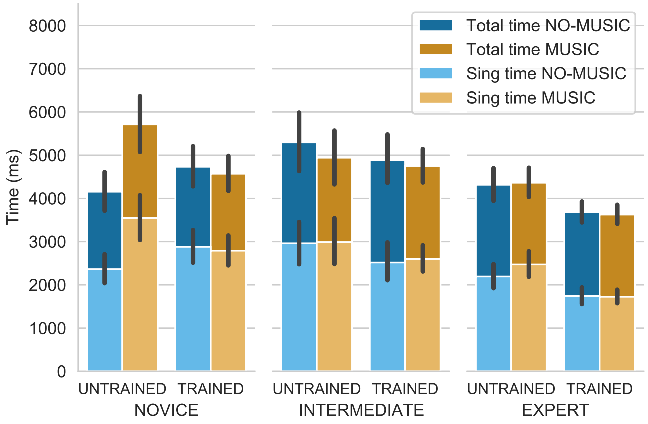


Figure 6.1: Pitch task *Total Time* and *Sing Time* by SKILL for each BACKGROUND (error bars in all graphs are 95% confidence).

participants. Upon further analysis, outliers were typically either: the first trial in a block, in which the participant took a long time to start singing; a trial where the participant had unusual difficulty finding the note to sing; or a trial where the participant sang too softly. In total, 15 outlier trials (1.3%) were removed.

Of the remaining trials, a minor error in client-side code resulted in 8 false positives and 2 false negatives that were corrected based on the system logs, representing 0.9% of the included trials.

6.1.1 *Total Time*

Time was consistent across most conditions. There were no main effects involving SKILL ($p = .32$), BACKGROUND ($p = .10$), or TRAINING ($p = .25$) on *Total Time* (Fig. 6.1).

Background music increased *Total Time* by 15% for novices, whereas it had an insignificant effect on other participants. There was an interaction between BACKGROUND and SKILL on boxcox-transformed *Total Time* ($F_{2,18} = 3.91$, $p = .039$, $\eta^2_G = .03$). Post hoc tests show, for the NOVICE participants, MUSIC (5129ms) was slower than NO-MUSIC (4447ms, $p < .0001$). While the standardized effect size of .03 is considered small [3], it is intriguing that music increases time.

6.1.2 *Sing Time*

Background music increased *Sing Time* by 241ms (Fig. 6.1). There was a main effect of BACKGROUND on on boxcox-transformed *Sing Time* ($F_{1,18} = 2.21$, $p < .007$, $\eta^2_G = .03$), where MUSIC (2648ms) was slower than NO-MUSIC (2407ms). This represents a 10% increase in time.

There was no interaction between BACKGROUND and SKILL for *Sing Time* ($p = .47$).

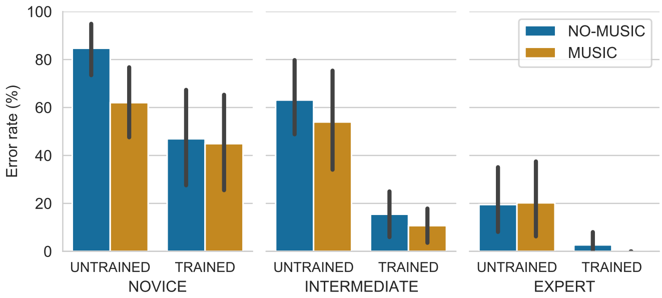


Figure 6.2: Pitch task *Error Rate* by SKILL and TRAINING for each BACKGROUND.

6.1.3 *Error Rate*

Training decreased *Error Rate* by 29.1% (Fig. 6.2). There was a main effect of TRAINING on *Error Rate* ($F_{1,18} = 43.52$, $p < .0001$, $\eta_G^2 = .71$), where TRAINED (19.6%) was lower than UNTRAINED (48.7%). The large .71 standardized effect size of TRAINING on *Error Rate* shows that people can improve, even if they perform poorly initially.

Unsurprisingly, skill affected *Error Rate* because we classified participant SKILL using this metric. Nevertheless, the results provide insight into the large variation in error rate for people with different pitch control and comfort with technology. There was a main effect of SKILL on aligned rank transformed (ART) *Error Rate* ($F_{2,18} = 43.73$, $p < .0001$, $\eta_G^2 = .83$). Post hoc tests show how EXPERT *Error Rate* (10.6%) was lower than INTERMEDIATE (35.8%), which was lower than NOVICE (59.6%) (all $p < .01$). This represents a decrease in error rate of 49.0% for the strongest participants, and the standardized effect size of .83 is considered large [3].

Training had a larger effect on *Error Rate* for intermediates than others. There was an interaction between TRAINING and SKILL on *Error Rate* ($F_{2,18} = 3.74$, $p = .044$, $\eta_G^2 = .29$). The decrease in *Error Rate* after training was larger for INTERMEDIATES (-45%) compared to EXPERTS (-19%, $p = .014$). After training, EXPERTS had a very low error rate of 1.3%. Considering the large standardized effect size of .50, intermediates have much more to gain from training than the others, while experts are able to nearly perfect the technique.

6.1.4 *NASA-TLX*

We measured task workload using the NASA Task Load Index (TLX) on a 7-point numeric scale [21]. Since measures were not normally distributed, all were analysed using Wilcoxon signed-rank tests with Holm-Bonferonni corrections. Experts had lower task load metrics than the other groups (Fig. 6.3). EXPERTS (4.0) perceived a lower *Mental Demand* than INTERMEDIATES (5.0, $p = .032$), but there were no significant differences between other groups. EXPERTS (1.9) also perceived better *Performance* than NOVICES (3.9, $p < .020$). Furthermore, EXPERTS (1.9) perceived lower *Frustration* than both INTERMEDIATES (3.8, $p = .039$) and NOVICES (5.0, $p < .003$). There were no significant

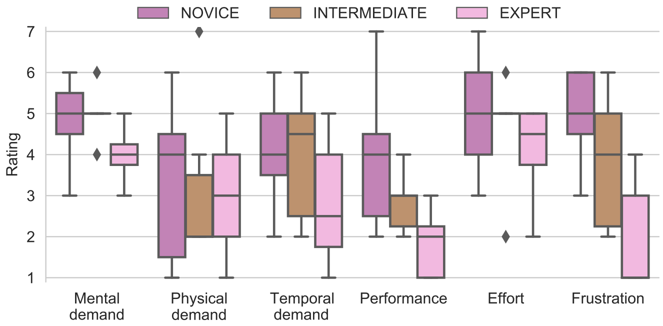


Figure 6.3: NASA-TLX ratings for pitch tasks. Lower values correspond to lower mental, physical, and temporal demand, as well as greater performance, lower effort, and lower frustration.

differences between ratings based on SKILL for *Physical Demand*, *Temporal Demand*, or *Effort*.

6.1.5 *Subjective Ratings*

Overall, 71% of participants perceived that background music made the tasks easier. When asked, "Did the presence of background music make the tasks easier or harder?" 15 rated it "somewhat easier" or better, 4 rated it "somewhat harder" or worse, and 2 said it had "no effect."

The majority of participants (62%) found pitch tasks enjoyable. When asked, "To what extent did you find the tasks enjoyable?" 13 rated the tasks "slightly enjoyable" or better, 3 rated them "slightly unenjoyable" or worse, and 5 rated them "neither enjoyable nor unenjoyable."

6.2 INTERVAL INTERACTIONS

As with the pitch experiment, for every participant ($n = 21$), outliers were removed (16 trials, 1.4%) and false positives (3) and false negatives (16) were corrected (1.6% of included trials).

6.2.1 *Total Time and Sing Time*

None of the studied factors impacted *Total Time* nor *Sing Time* for interval interactions (Fig. 6.4).

6.2.2 *Error Rate*

Training decreased *Error Rate* by 15.7% (Fig. 6.5). There was a main effect of TRAINING on *Error Rate* ($F_{1,18} = 9.28$, $p < .007$, $\eta^2_G = .34$), where TRAINED (20.9%) was lower than UNTRAINED (36.6%).

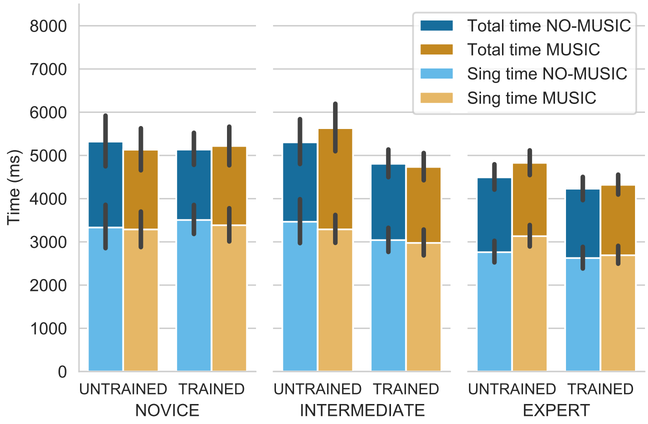


Figure 6.4: Interval task *Total Time* and *Sing Time* by TRAINING and SKILL for each BACKGROUND.

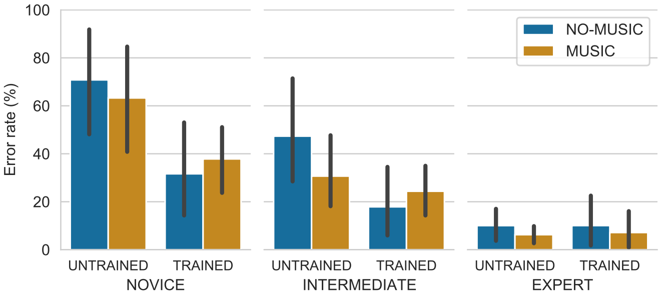


Figure 6.5: Interval task *Error Rate* by SKILL and TRAINING for each BACKGROUND.

As before, skill affected *Error Rate* due to our clustering. There was a main effect of SKILL on ART-transformed *Error Rate* ($F_{2,18} = 22.33$, $p < .0001$, $\eta^2_G = .71$). Post hoc tests show how EXPERT error rate (8.3%) was lower than both INTERMEDIATE (30.1%) and NOVICE (50.9%) rates (all $p < .002$). This represents a decrease in *Error Rate* of 42.6% for experts and the standardized effect size of .71 is considered large [3].

Interestingly, while training improved the *Error Rate* for most participants, it did not for experts. There was an interaction between TRAINING and SKILL on *Error Rate* ($F_{2,18} = 3.94$, $p = .038$, $\eta^2_G = .30$). Post hoc tests show the decrease in *Error Rate* after training for NOVICES (-32.3%) was much larger than for EXPERTS ($+0.5\%$, $p = .014$). The large .30 standardized effect size shows that NOVICES improve much more than others.

While background music did not directly affect *Error Rate*, training improved error rates more when there was no background music present. There was an interaction between SKILL and BACKGROUND on *Error Rate* ($F_{2,18} = 6.31$, $p = .022$, $\eta^2_G = .26$). Post hoc tests show TRAINING reduced *Error Rate* for NO-MUSIC (-21.4%) more than for MUSIC (-9.9%) ($p < .007$). Since the TRAINED *Error Rates* are comparable for both conditions, with 19.5% for NO-MUSIC

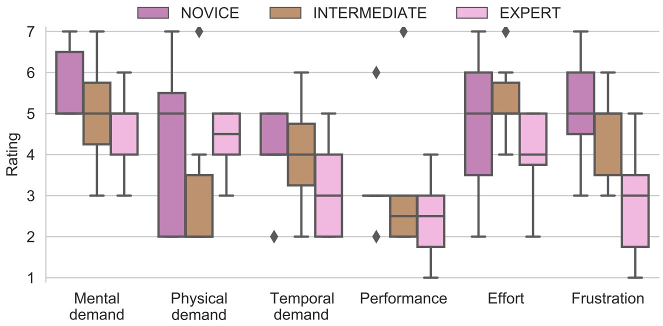


Figure 6.6: NASA-TLX ratings for interval tasks. Lower values correspond to lower mental, physical, and temporal demand, as well as greater performance, lower effort, and lower frustration.

and 22.3% for MUSIC, this suggests training is needed most when there is no music.

6.2.3 *NASA-TLX*

Again, experts had lower task load metrics than other groups (Fig. 6.6). EXPERTS (4.3) perceived lower *Effort* than INTERMEDIATES (5.7, $p = .049$). EXPERTS (2.9) also perceived lower *Frustration* than NOVICES (5.1, $p = .018$). There were no significant differences between ratings based on SKILL for *Mental Demand*, *Physical Demand*, *Temporal Demand*, or *Performance*.

6.2.4 *Subjective Ratings*

Overall, 67% of participants perceived that background music made the tasks easier. When asked how background music affected task difficulty, 14 rated the tasks “somewhat easier” or better, 4 rated them “somewhat harder” or worse, and 3 said background music had “no effect.” As with pitch tasks, the majority of participants (57%) found the interval tasks enjoyable. 12 rated the tasks “slightly enjoyable” or better, 8 rated them “slightly unenjoyable” or worse, and 1 rated them “neither enjoyable nor unenjoyable.”

6.3 MELODY INTERACTIONS

As with the previous two experiments, for every participant ($n = 21$), outliers were removed (15 trials, 1.3%) and false positives (8) and false negatives (10) were corrected (1.6% of included trials).

6.3.1 *Total Time*

Neither skill nor background music directly affected *Total Time* for melody interactions (Fig. 6.7). However, unlike the other tasks, training reduced *Total*

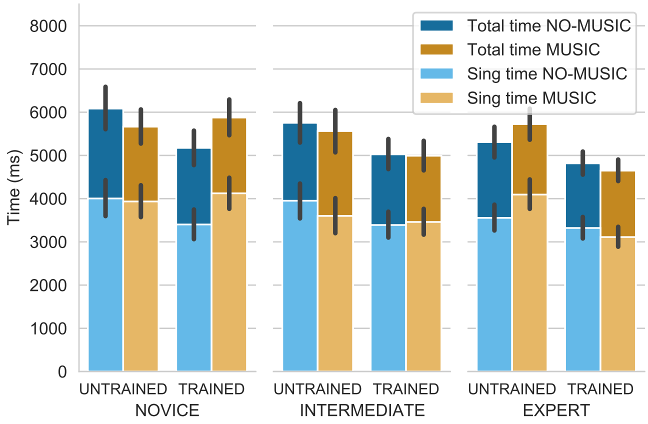


Figure 6.7: Melody task *Total Time* and *Sing Time* by TRAINING and SKILL for each BACKGROUND.

Time by 607ms. There was a main effect of TRAINING on boxcox-transformed *Total Time* ($F_{2,18} = 14.36$, $p < .001$, $\eta^2_G = .07$), where TRAINED (5073ms) was faster than UNTRAINED (5680ms). This represents a 11% decrease in *Total Time*.

Training impacted novices differently than experts when music was present. There was an interaction between SKILL, BACKGROUND, and TRAINING on *Total Time* ($F_{2,18} = 3.95$, $p = .038$, $\eta^2_G = .03$). Post hoc tests show that with MUSIC, training increased *Total Time* for NOVICES (+211ms) and decreased *Total Time* for EXPERTS (−1076ms) (all $p < .002$). This might indicate that novices found music distracting for these complex tasks.

6.3.2 *Sing Time*

Training improved *Sing Time* by 406ms (Fig. 6.7). There was a main effect of TRAINING on boxcox-transformed *Sing Time* ($F_{1,18} = 7.57$, $p = .013$, $\eta^2_G = .04$), where TRAINED (3456ms) was faster than UNTRAINED (3862ms). This represents a 11% decrease in *Sing Time*.

Again, there was an interaction between SKILL, BACKGROUND, and TRAINING on *Sing Time* ($F_{2,18} = 4.67$, $p = .023$, $\eta^2_G = .03$). Post hoc tests show that with MUSIC, training increased *Sing Time* for NOVICES (+189ms) and decreased *Sing Time* for EXPERTS (−982ms) (all $p < .002$).

6.3.3 *Error Rate*

Training decreased *Error Rate* by 10.4% (Fig. 6.8). There was a main effect of TRAINING on *Error Rate* ($F_{1,18} = 4.82$, $p = .041$, $\eta^2_G = .21$), with TRAINED (22.3%) lower than UNTRAINED (32.7%). Thus, training improved both time and error rate for melody tasks.

As before, skill affected *Error Rate* due to our clustering. There was a main effect of SKILL on ART-transformed *Error Rate* ($F_{2,18} = 54.77$, $p < .0001$, $\eta^2_G = .86$). Post hoc tests show how EXPERT *Error Rate* (7.2%) was lower than

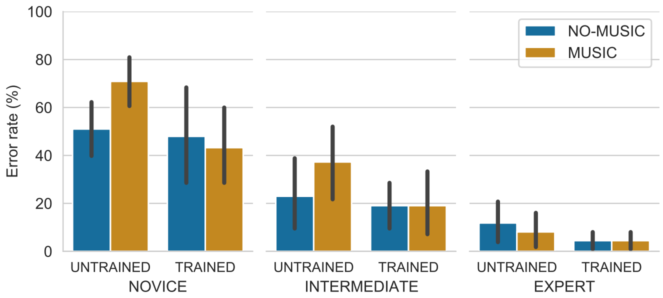


Figure 6.8: Melody task *Error Rate* by SKILL and TRAINING for each BACKGROUND.

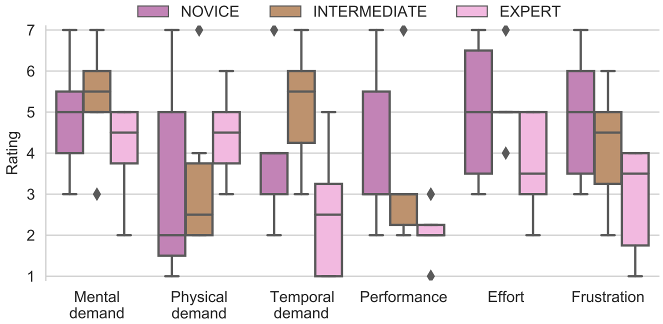


Figure 6.9: NASA-TLX ratings for melody tasks. Lower values correspond to lower mental, physical, and temporal demand, as well as greater performance, lower effort, and lower frustration.

INTERMEDIATE (24.6%), which was lower than NOVICE (53.3%) (all $p < .001$). This represents a decrease in error rate of 46.1% for the highest performing participants and the standardized effect size of .86 is considered large [3].

6.3.4 *NASA-TLX*

Again, experts had lower task load metrics than other groups (Fig. 6.9). EXPERTS (2.5) perceived a lower *Temporal Demand* than INTERMEDIATES (5.1, $p = .016$). EXPERTS (2.1) also perceived better *Performance* than NOVICES (4.1, $p = .015$). There were no significant differences between ratings based on SKILL for *Mental Demand*, *Physical Demand*, *Effort*, or *Frustration*.

6.3.5 *Subjective Ratings*

Overall, 57% of participants perceived that background music made the tasks easier. When asked how background music affected difficulty, 12 rated the tasks “somewhat easier” or better, 4 rated them “somewhat harder” or worse, and 5 said background music had “no effect.” Similar to the other tasks, 62% of participants found the tasks enjoyable. 13 rated the tasks “slightly enjoyable” or better, 5 rated them “slightly unenjoyable” or worse, and 3 rated them “neither enjoyable nor unenjoyable.”

7 DISCUSSION

The main objective for our experiments was to explore the feasibility and enjoyability of singing interactions. This chapter compares results for the three types of interactions and discusses the design considerations emerging from our research.

7.1 FEASIBILITY

In terms of feasibility, the main results are consistent across all types of interactions. For participants classified as experts, after training in all experiments, average time per trial was under 5s with error rates under 10%. For pitch tasks in particular, experts had a very low error rate of 1.3% after training. This is despite the pitch experiment's being first: the learning effect would have improved performance for interval and melody techniques, but pitch remained the easiest. On the other hand, those classified as novices performed poorly even after training, with average error rates between 35% (interval) and 46% (pitch and melody). This result shows that these techniques are only effective for a subset of potential users.

From our observations, the techniques work best for users with good pitch control and comfort with technology. Each of the 8 expert participants used laptops and desktops for at least 4h/day and had at least one year of private music lessons, excluding the one outlier who had no formal musical training. On the other hand, each of the 7 novices lacked either musical or technological experience: 5 had no private musical lessons and the remaining 2 spent only 1h/day using laptops and desktops. This time was much lower than the average of 7.7h/day. Lack of experience in either area likely made the interactions less intuitive and more difficult to perform. Thus, the results suggest both musical and technological skills contribute to user performance.

7.1.1 *Impact of Training*

While training improved performance for most groups and techniques, participants classified as intermediates showed the greatest improvement in error rates after training. This was particularly true for pitch, which is not surprising because it was the first technique evaluated. It also shows an expert does not have as much room for improvement as does an intermediate. Nevertheless, the results suggest that some people who perform poorly initially can improve with relatively little practice.

Of the three techniques, interval appears to be the hardest to master. Even experts had an error rate of 8.6% after training, whereas they achieved 1.3%

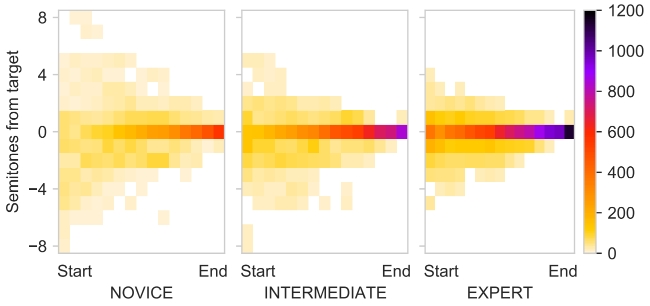


Figure 7.1: Heatmap for successful pitch trials showing frequency (number of semitones away from the target pitch) by time (from the start to the end of each trial) for each SKILL.

on pitch and 4.5% on melody tasks. Furthermore, with interval tasks, experts slightly worsened after training, perhaps indicating fatigue or impatience.

The pitch technique showed the best performance after training. Before training, novices (73.4%), intermediates (58.5%), and experts (19.9%) all had high error rates. While novices continued to have high error rates after training (45.9%), error rates were reasonably low for both intermediates (13.1%) and experts (1.3%) after training. Especially considering how the learning effect would have improved metrics for other techniques, our results demonstrate the pitch technique is the most effective.

7.1.2 *Patterns in Successful Trials*

Further analysis of the pitch experiment's data shows that experts were frequently able to find targets immediately, whereas novices typically had to slide into the appropriate pitch. The heatmap in Fig. 7.1 shows novices start on target for very few successful trials. On the other hand, for experts, the high density concentration on the correct target, along $y = 0$ in the figure, shows they frequently start on target. Furthermore, from observing the distribution of frequencies at the start of the trials, most EXPERT trials were within 3 semitones of the target pitch right from the beginning, whereas NOVICES were frequently much further. This indicates novices heavily rely on the visualization to find the correct pitch, whereas experts only need it for fine-tuning.

7.1.3 *Comparing Total Time*

Unsurprisingly, total time increases with interactions that have more notes. After training for all participants, pitch averaged 4319ms, interval averaged 4720ms, and melody averaged 5073ms. Interestingly, time does not increase proportionally to the number of notes in an interaction. This lack of increase is likely caused by two factors. First, the learning effect between experiments meant that participants were well practiced with pitch and interval interactions before performing melody interactions. Second, the recognition methods were different: after singing the first note, subsequent notes could be shorter (150ms for intervals, 50ms for melodies). Thus, adding additional notes to our interactions had less of an impact on time than expected.

In summary, all of our presented techniques are feasible for a subset of the population. However, some adjustments are necessary to make them easier for non-musical users.

7.2 ENJOYABILITY

Overall ratings of enjoyability indicate that all interactions were at least somewhat enjoyable for most participants. P6 (EXPERT), P7, and P13 (INTERMEDIATES) remarked that they really enjoyed the single pitch tasks. Furthermore, beyond their uses for general computing, the interactions could be used for ear training while performing day-to-day computing tasks. P6 remarked that they would "definitely use it to train my pitch in the future!"

P6 (EXPERT) and P16 (NOVICE) noted that by the melody experiment, they had mental and vocal fatigue. One can mitigate vocal fatigue by singing more softly or humming. With practice, the techniques would likely become much less mentally fatiguing. Nevertheless, these techniques are demanding and should be used in moderation in a real-world application.

Some participants found the tasks very challenging. P21 (NOVICE) said that the tasks were stressful and difficult to understand. Regarding melody tasks, P1 (INTERMEDIATE) said, "As a person who absolutely does not sing, I would have rather ran [sic] a 5km than do the task." Our techniques are probably best suited for motivated users who want to sing and improve their pitch.

It is possible that enjoyability could be improved by customizing the interactions based on the user's musical tastes and knowledge. For instance, users could define their own interactions in an application, such as intervals or melodies from a favourite song. This would make the interactions more familiar. Furthermore, using melodies from songs they enjoy might help them enjoy the interactions.

7.3 DESIGN CONSIDERATIONS

Our three experiments highlight key design considerations for using pitch, interval, and melody interactions in real computer interfaces.

7.3.1 *Use Background Music*

Interfaces using singing interactions should incorporate some form of background audio with an appropriate tonic and tuning. While background music did not affect error rate, and despite slightly increasing total time for pitch tasks, the majority of participants found tasks easier with background music in all experiments. P14 (EXPERT) stated that not having background music at the start of the study made it difficult to find notes until they figured out their tonic, and P4 (EXPERT) said the music helped them "count up" from the tonic to find notes. Background music also had positive effects for ensuring consistent performance over time: P12 (EXPERT) found it prevented them from getting out of tune, and P5 (INTERMEDIATE) said, "the music made it easier to hold the note steady." Thus, while background music did not always improve error rate, it did improve subjective metrics.

Background music should be simple and subtle. P4 (EXPERT) and P19 (NOVICE) found the music distracting and P12 (EXPERT) suggested simply playing the tonic every few seconds would be as good as having full background music. These comments might explain why background music increased total time for pitch tasks. P16 (NOVICE) also said that when the background music played the tonic, it was easier to sing notes correctly. Thus, simpler music that places even more emphasis on the tonic might improve error rates. Future work should explore how various types of auditory stimulation impacts performance.

7.3.2 *Facilitate Re-calibration*

For interfaces using singing interactions, it should be easy to recalibrate the system to a different vocal range. In our experiment, we disabled this functionality to ensure differences in performance were due to training. However, P19 (NOVICE), P5 (INTERMEDIATE), and P24 (EXPERT) felt tasks would have been easier with a different range than those they selected initially.

7.3.3 *Reduce Pitch Detection Granularity*

Pitch detection can use a lower granularity to reduce errors. Our system was capable of recognizing any of 12 possible notes in Western music by determining which of the 12 semitones were closest to the f_0 of the voice. This was a high level of granularity, resulting in small margins for error. This made it challenging to sing some notes, according to comments from P1 (NOVICE), P12, and P24 (EXPERTS). Reducing the granularity could allow for a larger margin of error.

Our experimental data logs included frequencies at every time step for every trial, allowing us to simulate how many trials would have been successful with a reduced granularity for pitch recognition. For instance, suppose only 3 of the 12 possible notes were recognizable. Then, singing a semitone off target would be labelled as correct, whereas the original pitch recognizer would have labelled it as incorrect. While user reactions might differ in a system with different granularity, our simulation still gives insight into how much granularity affected various skill levels.

This follow-up analysis adds an independent variable to our existing model: GRANULARITY with 3 levels (GRANULARITY-12, GRANULARITY-7, GRANULARITY-3). Each level represents a different pitch recognizer: GRANULARITY-12 recognizes all 12 notes, as in our original experiments; GRANULARITY-7 recognizes only the 7 notes in the major scale; and GRANULARITY-3 recognizes only de-

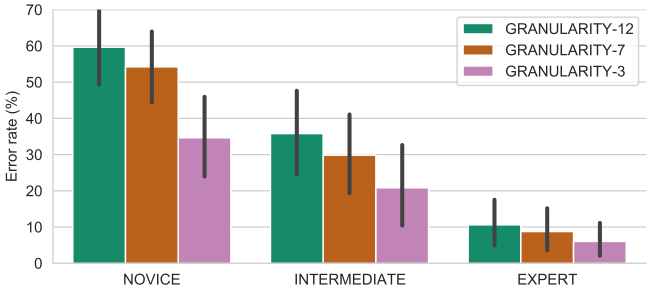


Figure 7.2: Pitch task *Error Rate* by skill for each granularity.

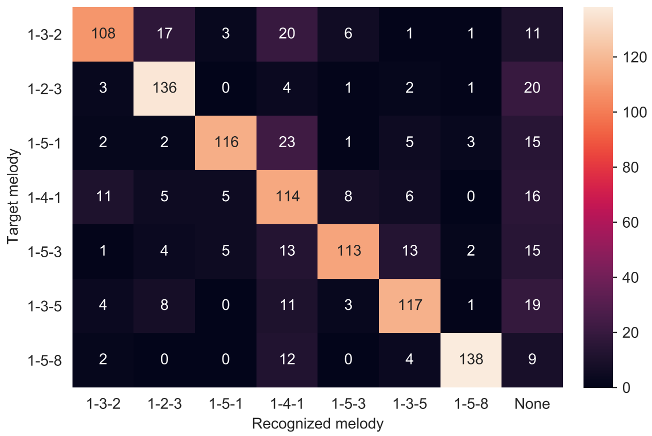
grees 1, 3, and 5. In the former two, all interactions were possible because our interactions used only notes in major scale. In the latter, only 3 interaction types could be used for each technique because of the much more limited granularity.

For brevity, we present results for only the pitch technique (Fig. 7.2). The interval technique has similar results, while the melody technique shows minimal improvement due to the use of DTW. In the analysis to follow, a granularity × skill ANOVA with Tukey HSD post hoc tests was used on ART-transformed *Error Rate*.

Decreasing granularity decreased *Error Rate* by as much as 14.3% for the pitch technique. There was a main effect of granularity on ART-transformed *Error Rate* ($F_{2,36} = 115.13$, $p < .0001$, $\eta^2_G = .86$). Post hoc tests show granularity-12 *Error Rate* (34.1%) was higher than granularity-7 (29.9%, $p < .003$), which was higher than granularity-3 (19.8%, $p < .0001$). This analysis shows a large improvement in *Error Rate*, as indicated by the standardized effect size of .86.

Decreasing granularity primarily improved *Error Rate* for novices and intermediates. There was an interaction between skill and granularity on *Total Time* ($F_{4,36} = 27.8$, $p < .0001$, $\eta^2_G = .76$). Post hoc tests show decreasing granularity from granularity-12 to granularity-3 decreased *Error Rate* much more for novices (-25.0%) and intermediates (-15.0%) than for experts (-4.6%) (all $p < .0001$). This suggests that while experts perform better with lower granularity, they are quite capable with higher granularity interactions.

In practice, one should certainly use granularity-7 instead of granularity-12. It has a positive effect on performance and virtually no drawbacks since we already recommend using only the 7 notes from the major scale. For novice users, it might make sense to use granularity-3 to further improve performance and make the techniques more accessible initially. Of course, there is a tradeoff in the number of commands or modes that can be invoked in a real application. Since the improvement is relatively small for experts, one could use granularity-3 as a beginner mode before a user is ready for granularity-7.



Target melody	1-3-2	1-2-3	1-5-1	1-4-1	1-5-3	1-3-5	1-5-8	None
1-3-2	108	17	3	20	6	1	1	11
1-2-3	3	136	0	4	1	2	1	20
1-5-1	2	2	116	23	1	5	3	15
1-4-1	11	5	5	114	8	6	0	16
1-5-3	1	4	5	13	113	13	2	15
1-3-5	4	8	0	11	3	117	1	19
1-5-8	2	0	0	12	0	4	138	9

Figure 7.3: Confusion matrix with the number of times each melody was recognized for each target melody.

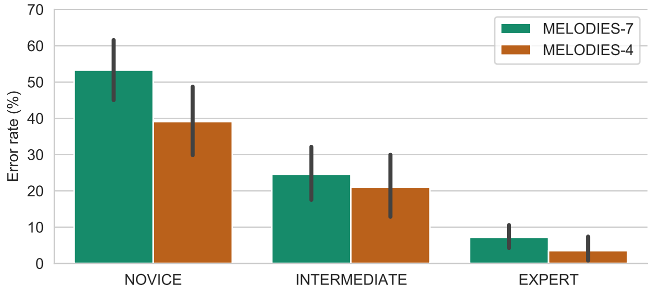


Figure 7.4: Melody task *Error Rate* by SKILL for each set of MELODIES.

7.3.4 *Reduce the Number of Melodies*

Choosing melodies requires care and using fewer melodies can improve performance. Our experiment evaluated 7 melodies, chosen to be easy to sing and distinguish. Despite this, some melodies were frequently confused with one another, as indicated by the confusion matrix in Fig. 7.3.

Three of our targets were frequently confused with other melodies: $1-3-2$, $1-5-1$, and $1-5-3$. This observation aligns with participant feedback: P5 (INTERMEDIATE) found it difficult to sing melodies that went up and then down again, especially when they did not return to note 1.

To see if recognizing fewer melodies could reduce error rates, we simulated the melody experiment using a melody recognizer that recognized only the other four melodies. This follow-up analysis adds another independent variable: MELODIES with 2 levels (MELODIES-7, MELODIES-4) representing the number of possible melodies. In the analysis to follow, a MELODIES × SKILL ANOVA with Tukey HSD post hoc tests was used on ART-transformed *Error Rate*.

Decreasing the number of melodies decreased *Error Rate* by 7.5% (Fig. 7.4). There was a main effect of MELODIES on *Error Rate* ($F_{1,18} = 42.21$, $p < .0001$,

$\eta_G^2 = .70$), where MELODIES-7 (27.5%) was higher than MELODIES-4 (20.4%). This is a large improvement, as indicated by the standardized effect size of .70.

Decreasing the number of melodies was especially helpful for novices. There was an interaction between MELODIES and SKILL on *Error Rate* ($F_{2,18} = 6.01$, $p < .01$, $\eta_G^2 = .40$). Post hoc tests show NOVICES (-14.1%) improved their rate more than EXPERTS (-3.7%, $p < .003$). This result mirrors how decreasing granularity especially improved *Error Rate* for novices.

The practical implication of these results is that using fewer melodies can improve performance for novices. To make interactions easier, one can consider using melodies that only ascend or that return to the tonic, making them easier to perform. Furthermore, one must ensure that melodies are very different from one another to ensure that they can be distinguished effectively.

7.3.5 *Interface Visualization Improvements*

The visualization should clearly communicate when a user can stop singing and what melody is recognized. P2 (INTERMEDIATE) noted it was hard to see how long they needed to hold a pitch. A clearer indication than just changing the colour of the arc (see Fig. 4.1), such as changing the background colour, might reduce *Total Time*. Furthermore, both P4 (EXPERT) and P7 (INTERMEDIATE) found it challenging to understand the visualization for melody tasks because the recognized melody changed frequently based on the most recent estimate from the DTW algorithm. The visualization could be improved by refreshing the recognized melody less frequently, perhaps once per second, as opposed to every 50ms.

7.4 LIMITATIONS

One potential limitation in our work is the variable system setup, which is a consequence of being a remote study. While all devices passed our system test, some participants likely used slower systems in suboptimal environments. This may have reduced performance compared to a tightly controlled experiment, but it allowed us to achieve some degree of external validity in varied environments.

Additionally, our study was limited to 21 participants. Since people vary widely in musical experience, the mean error rates could differ with different sample sets. We structured our analysis to illustrate the spread of performance for a variety of individuals.

The self-reported metrics for musical experience were not useful for our analysis. This was partially because self-reported measures of musical experience do not perfectly predict one's musical ability [16]. We instead relied on k-means to classify participants into skill levels. This allowed us to evaluate the spread of performance, but it prevented us from thoroughly analysing how musical ability affected performance. Future work should use a more objective measurement of musical ability [16].

We assumed participants had a one-octave vocal range. While this is generally a reasonable assumption [29], it is possible some participants had a more limited vocal range, thereby skewing the data for the novices in particular.

Another limitation is that our interaction feedback visualizations were not explicitly evaluated. While there was no strong evidence that the visualization caused significant problems, it is possible alternative visualizations could result in different performance. Because we needed a visualization that would support pitch, interval, and melody interactions, we deviated from the simple linear visualization suggested by Sporka [29]. Pilot studies guided our design for comprehension and visual appeal, but more research is needed to evaluate the effect of visualization on task performance.

8 APPLICATIONS

While the focus of our work is an initial evaluation of generalized singing interaction techniques, this chapter presents a series of envisioned use cases for these interactions. Specifically, it explores contexts where singing interactions can complement traditional input methods or provide standalone input.

8.1 COMPLEMENTING TRADITIONAL INPUT

Singing interactions can be used in tandem with traditional input, similar to how keyboard shortcuts supplement mouse input.

As one concrete example, consider a drawing application (Fig. 8.1). When drawing, one could switch to the eraser tool by singing pitch 4, then back to the paintbrush with 5. Changing brush size could be done with an interval starting at 1 and sliding up to the desired brush size. The melody $3 - 2 - 1$ could copy the user's selection and $3 - 4 - 5$ could paste. It would also be possible for users to create custom mappings for specific workflows. For instance, mapping the melody of the "Batman" theme song to inverting colour, or the starting interval in "All Star" by Smash Mouth to selecting a star shaped brush.

Singing interactions could be used to change how people experience computing, for both musicians and non-musicians alike. For example, in creative applications, this could integrate into an artist's flow to foster a new type of human-interface relationship, regardless of efficiency and productivity. Furthermore, users could choose to use singing interactions to improve their vocal skills while performing everyday computer tasks. Finally, melodies have the potential to be highly memorable: future research could compare the memorability of singing shortcuts to other techniques, such as keyboard shortcuts [7].

8.2 PROVIDING STANDALONE INPUT

Singing interactions can provide primary input for usage contexts like ubiquitous computing. For instance, low-power augmented reality (AR) glasses typically rely on speech recognition, which has latency and privacy concerns, or hand tracking and touch sensors, which can cause arm fatigue. Singing interactions could provide a viable alternative since they are hands-free and require minimal processing power.

A simple example is a 3D block construction application (Fig. 8.2). Users could use singing interactions to select colours, place new blocks, and delete existing blocks. For instance, singing intervals starting at degree 1 could select colours; singing degree 3 could create a new block with the current colour;

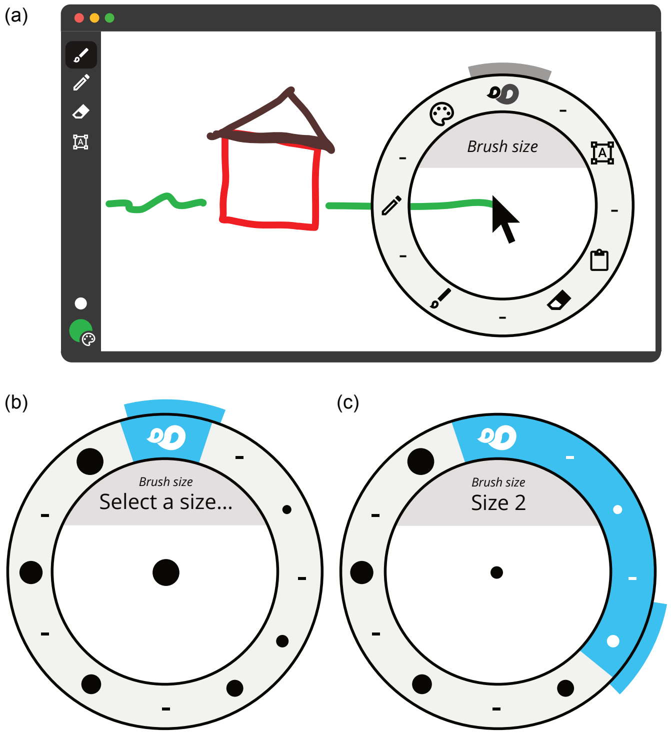


Figure 8.1: Illustration of how pitch can be used in a drawing application: (a) singing opens a visualization around the cursor with possible tools and settings; (b) when the first pitch of an interval interaction is recognized, more options appear for the second pitch; (c) when the second pitch is recognized, the interface displays the resulting effect.

and singing degree 5 could delete an indicated block. A translucent block at the centre of the view could indicate which block to delete or the location of a new block. As discussed in Chapter 7.3.3, using a limited set of pitches improves performance for novice users.

This design can be refined for expert users with low error rates. Instead of separate interactions for selecting colours and spawning blocks, experts could do both simultaneously. For instance, singing degree 5 could immediately spawn a blue block and 6 could spawn a purple block. Degree 7 could delete the indicated block. Using a wider range of interactions would decrease the time required for tasks with frequent mode switches. Furthermore, degree 1 could trigger standard and user-defined controls: singing a series of degrees $1-8-1$ could clear existing blocks, $1-3-5$ could save, and $1-8-5$ could load the last saved model. Adding these melody commands provides expert users with an array of shortcuts without needing physical inputs or speech recognition.

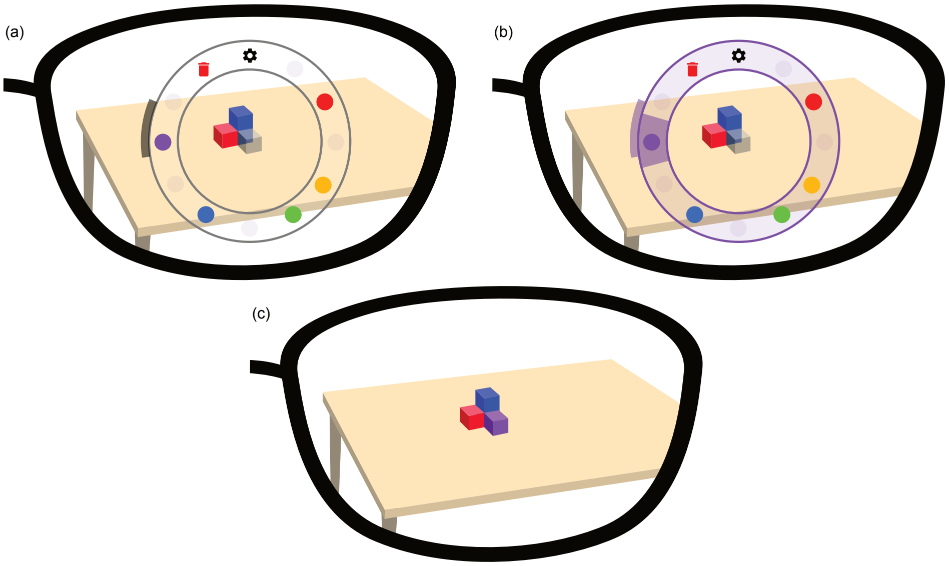


Figure 8.2: Illustration of expert singing interactions for an AR glasses 3D block construction application: (a) singing opens a visualization with a palette of colours for a translucent new block being placed in the middle of the view; (b) when a pitch interaction is recognized, the corresponding colour is selected for the new block; (c) when the interaction ends, the block becomes opaque.

8.3 EXTENDING TO EYES-FREE CONTEXTS

Singing interactions could be adapted to eyes-free contexts. Future work could explore how effectively non-visual feedback could help users perform the interactions. For instance, one could change the A/C fan speed in a car by singing an interval. This would allow a driver to make adjustments without letting go of the wheel, or a backseat passenger to take control without leaning forward. The physical feedback of air movement could be a sufficient replacement for a visualization, assuming a constrained range of possible interactions. Similarly, one could sing either a melody to play music or an interval to control volume on Bluetooth earbuds. The auditory feedback from changing the volume or changing playback could be sufficient to help users perform their desired interactions. Our work represents an early step towards making these types of eyes-free interactions possible.

8.4 SINGING FOR ACCESSIBILITY

Future work should examine how effectively singing interactions can help users with motor or speech impairments. Other pitch-based non-verbal vocal interactions, such as CHANTI [30], have been effective for quadriplegic users, as well as users with Friedreich ataxia and carpal tunnel syndrome. While our singing interactions require greater pitch control than CHANTI, the wider selection of possible pitches, intervals, and melodies could facilitate a broader set of interactions. Additionally, because our interactions rely on pitch and not vowel or consonant sounds, they may be feasible even for people with speech impediments.

9 CONCLUSION

Our work has explored the design of pitch, interval, and melody interactions for computer input and evaluated their feasibility and enjoyability. All interactions were feasible for the highest performers, with very low error rates for pitch interactions in particular. While a subset of participants had high error rates, the majority thought the interactions were enjoyable. Using our results, we made recommendations for using singing interactions, including using background music, recalibration, reduced pitch granularity, and fewer melodies. Our findings demonstrate that people with good pitch and comfort with technology can feasibly use singing for interactions. With possible applications in traditional, ubiquitous, and eyes-free computing, singing interactions have the potential to add a musical layer to interface tasks.

BIBLIOGRAPHY

[1] Ofer Amir, Noam Amir, and Liat Kishon-Rabin. "The effect of superior auditory skills on vocal accuracy." In: *The Journal of the Acoustical Society of America* 113.2 (Jan. 2003), pp. 1102–1108. ISSN: 0001-4966. DOI: 10.1121/1.1536632.

[2] MIDI Manufacturers Association. *Complete MIDI 1.0 Detailed Specification*. Mar. 1996.

[3] Roger Bakeman. "Recommended effect size statistics for repeated measures designs." en. In: *Behavior Research Methods* 37.3 (Aug. 2005), pp. 379–384. ISSN: 1554-351X, 1554-3528. DOI: 10.3758/BF03192707.

[4] Jean Callaghan, William Thorpe, and Jan van Doorn. "The science of singing and seeing." In: *Proceedings of the Conference on Interdisciplinary Musicology (CIM04)*. Graz, Austria: Society for Interdisciplinary Musicology, Apr. 2004, pp. 1–10.

[5] Supadaech Chanjaradwichai, Proadpran Punyabukkana, and Atiwong Suchato. "Design and evaluation of a non-verbal voice-controlled cursor for point-and-click tasks." In: *Proceedings of the 4th International Convention on Rehabilitation Engineering & Assistive Technology*. iCREATe '10. Midview City, SGP: Singapore Therapeutic, Assistive & Rehabilitative Technologies (START) Centre, July 2010, pp. 1–4. ISBN: 978-981-08-6199-5.

[6] Jackson Feijó Filho, Wilson Prata, and Thiago Valle. "Pufftext: A puff controlled software-based hands-free spin keyboard for mobile phones." In: *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '13. New York, NY, USA: ACM, Aug. 2013, pp. 468–471. ISBN: 978-1-4503-2273-7. DOI: 10.1145/2493190.2494661.

[7] Tovi Grossman, Pierre Dragicevic, and Ravin Balakrishnan. "Strategies for accelerating on-line learning of hotkeys." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. New York, NY, USA: Association for Computing Machinery, Apr. 2007, pp. 1591–1600. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240865.

[8] Susumu Harada, James A. Landay, Jonathan Malkin, Xiao Li, and Jeff A. Bilmes. "The Vocal Joystick: Evaluation of voice-based cursor control techniques." In: *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS '06. New York, NY, USA: ACM, Oct. 2006, pp. 197–204. ISBN: 978-1-59593-290-7. DOI: 10.1145/1168987.1169021.

[9] Susumu Harada, T. Scott Saponas, and James A. Landay. "VoicePen: Augmenting pen input with simultaneous non-linguisitic vocalization." In: *Proceedings of the 9th International Conference on Multimodal Interfaces*. ICMI '07. New York, NY, USA: ACM, Nov. 2007, pp. 178–185. ISBN: 978-1-59593-817-6. DOI: 10.1145/1322192.1322225.

[10] Susumu Harada, Jacob O. Wobbrock, and James A. Landay. "VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments." In: *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS '07. New York, NY, USA: ACM, Oct. 2007, pp. 27–34. ISBN: 978-1-59593-573-1. DOI: 10.1145/1296843.1296850.

[11] Susumu Harada, Jacob O. Wobbrock, and James A. Landay. "Voice games: Investigation into the use of non-speech voice input for making computer games more accessible." en. In: *Human-Computer Interaction – INTERACT 2011*. Ed. by Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 11–29. ISBN: 978-3-642-23774-4. DOI: 10.1007/978-3-642-23774-4_4.

[12] Brandi House, Jonathan Malkin, and Jeff Bilmes. "The VoiceBot: A voice controlled robot arm." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. New York, NY, USA: ACM, Apr. 2009, pp. 183–192. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518731.

[13] Perttu Hämäläinen, Teemu Mäki-Patola, Ville Pulkki, and Matti Airas. "Musical computer games played by singing." en. In: *Proceedings of the 7th International Conference on Digital Audio Effects*. Naples, Italy: DAFX, 2004, pp. 367–371.

[14] Takeo Igarashi and John F. Hughes. "Voice as sound: Using non-verbal voice input for interactive control." In: *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. UIST '01. New York, NY, USA: ACM, Nov. 2001, pp. 155–156. ISBN: 978-1-58113-438-4. DOI: 10.1145/502348.502372.

[15] Global Web Index. *Voice search insights report*. 2018. URL: https://www.gwi.com/hubfs/Downloads/Voice-Search-report.pdf.

[16] Lily N. C. Law and Marcel Zentner. "Assessing musical abilities objectively: Construction and validation of the profile of music perception skills." In: *PLoS ONE* 7.12 (Dec. 2012), e52508. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0052508. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3532219/ (visited on 07/29/2022).

[17] Chang-Hung Lin, Yuan-Shan Lee, Ming-Yen Chen, and Jia-Ching Wang. "Automatic singing evaluating system based on acoustic features and rhythm." In: *2014 International Conference on Orange Technologies*. ICOT '14. Xi'an, China: IEEE, Sept. 2014, pp. 165–168. DOI: 10.1109/ICOT.2014.6956625.

[18] David Marino, Paul Bucci, Oliver S. Schneider, and Karon E. MacLean. "Voodle: Vocal doodling to sketch affective robot motion." In: *Proceedings of the 2017 Conference on Designing Interactive Systems*. DIS '17. New York, NY, USA: ACM, June 2017, pp. 753–765. ISBN: 978-1-4503-4922-2. DOI: 10.1145/3064663.3064668.

[19] Philip McLeod and Geoff Wyvill. "A smarter way to find pitch." In: *Proceedings of the 2005 International Computer Music Conference*. Vol. 5. ICMC '05. Barcelona, Spain: ACM, 2005, pp. 138–141.

[20] Thomas Murry. "Pitch-matching accuracy in singers and nonsingers." en. In: *Journal of Voice* 4.4 (Jan. 1990), pp. 317–321. ISSN: 0892-1997. DOI: 10.1016/S0892-1997(05)80048-7.

[21] NASA. *NASA TLX: Task Load Index*. Dec. 2020. URL: https://humansystems.arc.nasa.gov/groups/tlx/ (visited on 07/29/2022).

[22] Nathalia Peixoto, Hossein Ghaffari Nik, and Hamid Charkhkar. "Voice controlled wheelchairs: Fine control by humming." en. In: *Computer Methods and Programs in Biomedicine* 112.1 (Oct. 2013), pp. 156–165. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2013.06.009.

[23] Peter Q. Pfordresher, Steven Brown, Kimberly M. Meier, Michel Belyk, and Mario Liotti. "Imprecise singing is widespread." In: *The Journal of the Acoustical Society of America* 128.4 (Oct. 2010), pp. 2182–2190. ISSN: 0001-4966. DOI: 10.1121/1.3478782.

[24] Ondřej Poláček and Zdeněk Míkovec. "Hands free mouse: Comparative study on mouse clicks controlled by humming." In: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '10. New York, NY, USA: ACM, Apr. 2010, pp. 3769–3774. ISBN: 978-1-60558-930-5. DOI: 10.1145/1753846.1754053.

[25] Ondřej Poláček, Zdeněk Míkovec, and Pavel Slavík. "Predictive scanning keyboard operated by hissing." In: *Proceedings of the 2nd IASTED International Conference on Assistive Technologies*. AT '12. Innsbruck, Austria: ACTA, Feb. 2012, pp. 862–869. DOI: 10.2316/P.2012.766-002.

[26] Ondřej Poláček, Zdeněk Míkovec, Adam J. Sporka, and Pavel Slavik. "Humsher: A predictive keyboard operated by humming." In: *Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*. ASSETS '11. New York, NY, USA: ACM, Oct. 2011, pp. 75–82. ISBN: 978-1-4503-0920-2. DOI: 10.1145/2049536.2049552.

[27] Daisuke Sakamoto, Takanori Komatsu, and Takeo Igarashi. "Voice augmented manipulation: Using paralinguistic information to manipulate mobile devices." In: *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '13. New York, NY, USA: ACM, Aug. 2013, pp. 69–78. ISBN: 978-1-4503-2273-7. DOI: 10.1145/2493190.2493244.

[28] A. J. Sporka, S. H. Kurniawan, and P. Slavík. "Non-speech operated emulation of keyboard." en. In: *Designing Accessible Technology*. Ed. by John Clarkson, Patrick Langdon, and Peter Robinson. London: Springer, 2006, pp. 145–154. DOI: 10.1007/1-84628-365-5_15.

[29] Adam J. Sporka. "Pitch in non-verbal vocal input." en. In: *ACM SIGACCESS Accessibility and Computing* 1.94 (June 2009), pp. 9–16. ISSN: 1558-2337, 1558-1187. DOI: 10.1145/1595061.1595063.

[30] Adam J. Sporka, Torsten Felzer, Sri H. Kurniawan, Ondřej Poláček, Paul Haiduk, and I. Scott MacKenzie. "CHANTI: Predictive text entry using non-verbal vocal input." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. New York, NY, USA: ACM, May 2011, pp. 2463–2472. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979302.

[31] Adam J. Sporka, Sri Hastuti Kurniawan, and Pavel Slavik. "Whistling User Interface (U3I)." en. In: *User-Centered Interaction Paradigms for Universal Access in the Information Society*. Ed. by Christian Stary and Constantine Stephanidis. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 472–478. ISBN: 978-3-540-30111-0. DOI: 10.1007/978-3-540-30111-0_41.

[32] Adam J Sporka, Ondřej Poláček, and Jan Havlik. "Segmentation of speech and humming in vocal input." en. In: *Radioengineering* 21.3 (2012), p. 7.

[33] Annie H. Takeuchi and Stewart H. Hulse. "Absolute pitch." In: *Psychological Bulletin* 113.2 (1993). Place: US Publisher: American Psychological Association, pp. 345–361. ISSN: 1939-1455. DOI: 10.1037/0033-2909.113.2.345.

[34] Graham F. Welch, D. M. Howard, and C. Rush. "Real-time Visual Feedback in the Development of Vocal Pitch Accuracy in Singing." en. In: *Psychology of Music* 17.2 (Oct. 1989), pp. 146–157. ISSN: 0305-7356. DOI: 10.1177/0305735689172005.

[35] Pat H Wilson, C William Thorpe, and Jean Callaghan. "Looking at singing: Does real-time visual feedback improve the way we learn to sing." In: *Proceedings of the 2nd APSCOM Conference*. Seoul, South Korea: Asia-Pacific Society for the Cognitive Sciences of Music, Aug. 2005.

[36] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. "The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures." en. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, May 2011, pp. 143–146. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1978963.

[37] Daniel Zielasko, Sebastian Freitag, Dominik Rausch, Yuen C. Law, Benjamin Weyers, and Torsten W. Kuhlen. “BlowClick: A non-verbal vocal input metaphor for clicking.” In: *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. SUI ’15. New York, NY, USA: ACM, Aug. 2015, pp. 20–23. ISBN: 978-1-4503-3703-8. DOI: 10.1145/2788940.2788953.

[38] Graeme Zinck and Daniel Vogel. “Evaluating Singing for Computer Input Using Pitch, Interval, and Melody.” In: *CHI Conference on Human Factors in Computing Systems*. CHI ’22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10.1145/3491102.3517691.