MUSICAL FEATURES MODIFICATION FOR LESS INTRUSIVE DELIVERY OF POPULAR NOTIFICATION SOUNDS

Jing Yang

Department of Computer Science
ETH Zurich, Switzerland
jing.yang@inf.ethz.ch

Andreas Roth

Department of Computer Science
ETH Zurich, Switzerland
rothand@ethz.ch

ABSTRACT

Less intrusive information delivery has been a popular research topic for auditory displays. While most research has addressed this issue by creating new notification cues such as rendering ambient soundscapes or modifying background music, we present a novel method to gently deliver artificial *notification sounds* that have been commonly used in digital devices and for popular applications. We propose to play a notification sound by embedding it into the music that a user is listening to, after changing the musical timbre, amplitude, tempo, and octave of the notification to match these features of the music. To implement this concept, we extend a melody extraction algorithm for notification timbre transfer, and we present a pipeline that algorithmically selects a proper time spot and harmoniously embeds the notification into music. To validate our design concept, we present a user study comparing our method with the standard method of playing notification sounds on digital devices. Through an extensive analysis of 96 tasks performed by 32 participants, we demonstrate that our method can deliver notification sounds in a less intrusive but adequately noticeable manner and is preferred by most participants.

1. INTRODUCTION

This paper explores a novel method to deliver notification sounds in a less intrusive but adequately noticeable manner. Various forms of notifications (e.g., visual, auditory, haptic) effectively connect us with our activity context and devices or objects we interact with. While many of us live with notifications on a daily basis, we may also suffer from the interruption caused by them that negatively impacts our productivity and causes stress [1, 2, 3, 4]. However, switching off notifications is not a satisfying solution either, since this might make users unaware of their activity context [5, 6].

Such a double-edged sword character has motivated some research that explores notification delivery through the auditory channel in effective but less intrusive ways [7, 8, 9, 10, 11, 12]. A typical approach is to deliver notifications by rendering an ambient soundscape [8] or integrating audio cues (e.g., a specific motif or melody pattern) into existing ambient soundscapes [10, 11, 12]. In recent years, as digital music has been widespread and many people have the habit of conducting activities over music, some researchers deliver auditory notifications by adding acoustic effects

This work is licensed under Creative Commons Attribution Non Commercial 4.0 International License. The full terms of the License are available at http://creativecommons.org/licenses/by-nc/4.0/

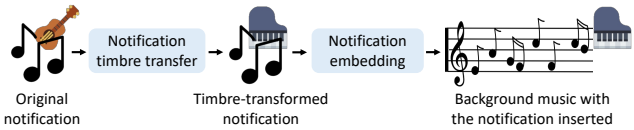


Figure 1: Our idea of delivering notification sounds in a less distracting but still effective manner. We first change the musical timbre of the original notification sound into the timbre of the music that a user is listening to, and then we deliver the timbre-transformed notification by harmoniously embedding it into the music. Implementation details are described in Section 3.

(e.g., reverb) to the music [9] or modifying properties (e.g., pitch) of the music [7] that the user is listening to.

A common feature of previous works is that they introduce new audio cues as the notification signal, while little work has focused on the already existing *notification sounds* that have been commonly used on digital smart devices (e.g., iOS/Android systems) or for popular applications (e.g., Skype, WeChat). Some of these notification sounds contain a monophonic note sequence or a simple melody, and some are polyphonic musical pieces. Since these notification sounds have been widely used, users may already be used to their interpretations that are associated with specific applications and/or services, which motivates an investigation into these notifications rather than introducing new audio cues.

We contribute a novel method that can deliver the commonly used notification sounds in a less intrusive but adequately noticeable manner. As illustrated in Figure 1, we propose to first transfer the musical timbre of the notification into the timbre of the background music that a user is listening to, while preserving the original notification melody envelope to still associate the notification with its interpretation. Next, we harmoniously embed the timbre-transformed notification into the background music for information delivery.

One potential approach to notification timbre transfer is to apply audio style transfer techniques [13, 14, 15]. However, the lack of sufficient training data and the issue of ambiguous timbre definition of the commonly used notification sounds largely limit the applicable conversion method. To tackle these challenges, we first extract the melody of a notification, which we implement by applying and extending the CREPE model [16]. Afterwards, we change timbre by flexibly assigning the target instrument to the extracted notification melody using the MIDI interface. To seamlessly integrate the timbre-transformed notification into music, we adjust the amplitude, tempo, and the overall octave level of the notification

https://doi.org/10.21785/icad2021.016

to match these features of the music. Finally, we embed the notification with fade-in and fade-out effects, and the embedding spot is selected by algorithmically checking and finding the segment of the music [17] that contains the most similar musical features to the notification sound.

To explore users' experience with our notification delivery method, we conducted a study in which the participants did cognitive tasks over a piece of background music, during which notification sounds were delivered using our method, and/or the standard method in digital devices nowadays that a notification is played unchanged in parallel to the music being played. Results show that compared with the standard method, the participants perceived notifications delivered using our method significantly less distracting. In addition, although our method was also perceived to be less noticeable, we only observed two missing notifications out of a total of 192 tests, and the participants generally believed that they captured all the notifications delivered using our method. Overall, the key contributions of this paper are the following:

- We present one of the first studies that explore a novel method to deliver the commonly used artificial notification sounds in a less intrusive but adequately noticeable manner.
- To implement our idea, we develop a melody extraction-based approach to timbre transfer for arbitrary notifications, and we present an algorithm for seamless notification integration into music.
- To demonstrate the effectiveness of our notification delivery method, we present a user study and an extensive evaluation. With the results, we also discuss implications for future work.

2. RELATED WORK

Balancing intrusiveness and noticeability and finding appropriate moments for notification delivery constitute popular research topics in the field of human-computer interaction (HCI) [18, 19, 20], in which the exploration on auditory notifications makes an important part and has also been covered in previous ICAD conferences [8, 11].

To provide auditory notifications effectively but in a less distracting manner, Kilander and Lönnqvist [8] proposed to render an artificial audio ambience using pre-designed sound signals (e.g., a thunderstorm sequence), and they added reverb effect to create an ambient atmosphere. Butz [10] and Jung [11, 12] first added an ambient soundscape in the users' environment and then delivered notifications by integrating audio cues (e.g., a specific motif) in the soundscape. The above works focus on public spaces like exhibition halls as the application environment. Hence, these approaches are more suitable for public notifications. However, it is difficult to scale these approaches to a large number of users who prefer different environmental soundscapes or notification cues.

On the other hand, some researchers developed methods for more personal auditory notification delivery. Barrington et al. [9] proposed to give notifications by adding acoustic effects (e.g., skipping by several beats, tempo modulation, frequency filtering) to the music that a user is listening to. Ananthabhotla and Paradiso [7] proposed to deliver notifications by modifying some properties (e.g., amplitude, tempo, pitch) or altering a short segment of the music being played, and they developed modifications at different levels of intrusiveness. These two approaches are more suitable for providing personal notifications as they leverage a user's own music and aim to deliver notifications when the user is conducting activities over music.

Two indications can be learnt from previous research. First, users might want to select their preferred notification signals [10, 11, 12]. This motivates us to explore the delivery of artificial notification sounds, since they have been commonly used, and users might already be familiar with their preferred notification sounds and the corresponding applications. Second, to deliver auditory notifications more gently along with background music, the notification signal can be designed to fit in the music with some noticeable change [7, 9], such as filtering some specific frequencies [9], or shifting the pitch of a short segment of the music [7], while the other musical properties are preserved. Hence, in our case, we propose to keep the original melody envelope of the notification sound to still associate the notification with its interpretation, while changing the timbre, amplitude, tempo, and octave level of the notification according to the background music.

While the other properties can be easily changed using basic signal processing techniques, timbre transfer is challenging for notification sounds. Musical timbre transfer techniques have been explored for normal music pieces [13, 14, 15], but these models are hardly feasible for the commonly used notification sounds due to the issues of ambiguous notification timbre and the lack of training data. In this work, we develop a novel melody extraction-based method for notification timbre transfer, and we extend the CREPE model [16] for melody extraction. While the original implementation of the CREPE model focuses on melody extraction from arbitrary monophonic music, our algorithm makes it also applicable for polyphonic music. Moreover, we propose to search for appropriate insertion spots in the background music algorithmically. Furthermore, as inspired by [21], we implement fade-in and fade-out effects for seamless integration of the notification sounds.

3. METHODS FOR LESS INTRUSIVE NOTIFICATION DELIVERY

In this section, we describe our approaches and implementation details (see Figure 2) of our notification delivery method. In this work, we focus on transfer into single-instrument timbre (e.g., piano, guitar, cello), and we use solo background music.

We collected 81 notification sounds from iOS/Android systems and popular applications such as Skype. These notifications contain a few notes or a simple melody with a duration of around $2-12\,s$. All collected notifications were processed into WAV format with a sampling rate of $16\,kHz$. Our notification dataset and audio samples of this work can be found in the supplement[1].

To evaluate our algorithms with respect to previous work, and to empirically determine the parameters for our implementation, we use the Slakh2100 dataset [22] that consists of paired WAV-MIDI data. Note that a general evaluation of our algorithms cannot be conducted over the notification dataset, since there is no ground truth melody, i.e., pitch notations, for each notification audio wave.

3.1. Notification Timbre Transfer

As illustrated in Figure 2, we first implement notification timbre transfer. To this end, we first extract the melody of a given notification audio wave into a piano roll, which is then transformed into MIDI data, and we change the notification timbre using the MIDI interface.

[1]Supplementary materials: `https://gladys0313.github.io/ICAD2021-notification-delivery/`

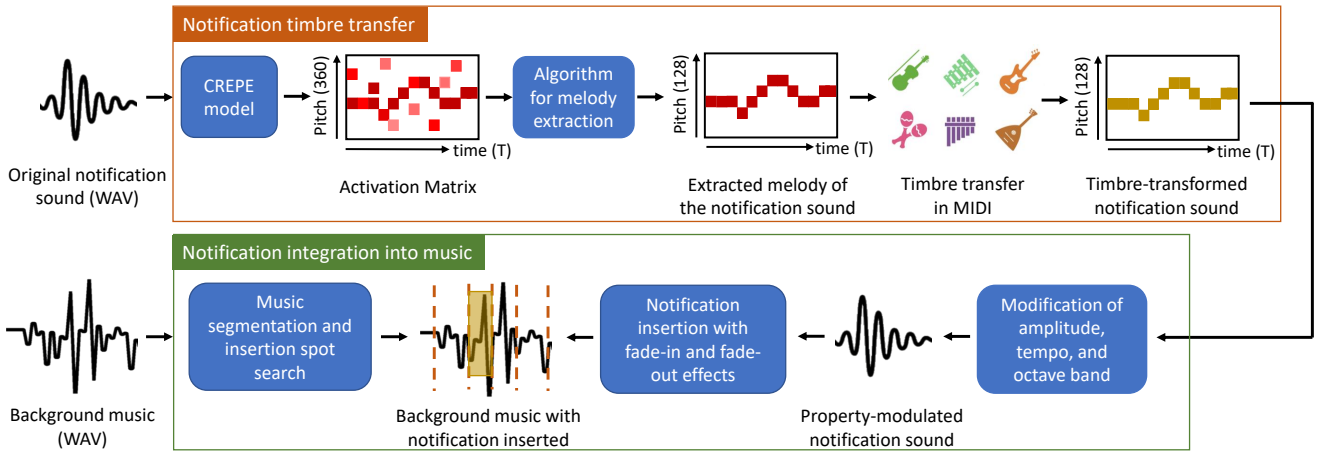


Figure 2: The implementation of our notification delivery method.

3.1.1. *Melody Extraction*

We build our melody extraction method based on CREPE [16], a deep convolutional neural network that has been trained to recognize frequencies in an audio signal. Given an audio wave, the CREPE model outputs an *activation matrix* of shape $(360, T)$. Each of the 360 entries corresponds to a specific pitch value between notes C1 and B7, and each of the T entries represents a time step of the audio signal with a step size of $4\,ms$. The activation matrix contains activation strength values (i.e., probabilities that a pitch is active) for each detected pitch at each time step. Afterwards, according to the original implementation by the authors of the CREPE model [16], noisy pitch detections are filtered by setting a threshold for the activation strength values. Depending on the threshold, this filtering step might already remove all detected pitches at some time steps. Then, at each time step that still has detected pitches, the pitch with the highest activation strength will be kept, from which the melody contour of the input audio is calculated.

One problem is that the CREPE model, together with the implementation mentioned above, was proposed and trained to extract melody from *monophonic* music. However, some notification sounds include polyphonic segments. Setting an activation strength threshold to detect active notes can remove noisy note predictions, but it may also remove some notes that should exist in the audio. This leads to ambiguous melody extraction when several notes are played together. Hence, to develop a melody extraction approach more suitable for polyphonic music pieces, we propose algorithms including note activity detection, polyphonic pitch extraction, and temporal smoothing based on the raw activation matrix from the CREPE model.

Note activity detection. Given the raw activation matrix, our goal of note activity detection is to filter noise while keeping as many valid pitches as possible for the sake of polyphonic pitch extraction. To this end, rather than using a threshold on the activation strength values [16, 23], we propose an algorithm to determine active notes based on the loudness contour of the input audio. Our method includes the following steps:

(1) We convert the amplitudes of input audio signal S to decibel(dB)-scaled values using the formula $S_{db} = 10 \cdot \log_{10}(S^2) - 10 \cdot \log_{10}(\text{ref})$ with ref $= 20.7$ [24].

(2) We then segment the dB-scaled time-series audio into bins. In each bin, we only keep the maximum amplitude while removing the others. Since the sampling rate of our audio data is $16\,kHz$, which is equivalent to a time step of $0.0625\,ms$, and the activation matrix from the CREPE model has a time step of $4\,ms$, we use a bin size of 64 to align the time step of our processing with the time step of the activation matrix.

(3) We normalize the remaining amplitude values from the previous step into the range $[0, 1]$, and we set a threshold for the normalized decibel value to select the active notes in the activation matrix. More specifically, at each time step, if the normalized amplitude is larger than the threshold, then all detected pitches at this time step in the activation matrix will be kept, otherwise they will all be filtered.

Experiments on the Slakh2100 dataset showed that our loudness-based method performed better than the original CREPE method. Our method could achieve an average note activity detection accuracy of 84.5% at a normalized threshold of 0.45. In contrast, the original CREPE method achieved the highest average accuracy of 79.1% at an activation strength threshold of 0.8. We also used the threshold of 0.45 for our notification sounds in the later user study.

Polyphonic pitch extraction. After note activity detection, the current activation matrix contains (most) pitches we expect to preserve, while it may also contain noise that should be further removed. At each time step, rather than only keeping the frequency with the highest activation strength value as implemented in the original method for the CREPE model [16], we implement the following two steps to preserve multiple notes at each time step:

(1) To remove noise that usually has little activation strength, we first remove all pitches in the activation matrix with an activation strength value smaller than an absolute threshold.

(2) Next, we normalize the remaining activation strength values into the range $[0, 1]$. Then at each time step, we preserve the detected pitches of which the normalized activation strength is larger than a normalized threshold.

After these two steps, we are left with the most prominent fundamental frequencies of the audio signal. We then convert the fundamental frequencies to pitches in the range of $[0, 127]$ using the formula $pitch = 12 \times (\log_2(f) - \log_2(440.0) + 69)$. Therefore,

at this stage, we obtain a matrix of shape $(128, T)$ where T still represents the length of the original audio input divided by the step size of $4\,ms$. This matrix is called a *piano roll* and can be directly loaded into MIDI representation for further processing.

Temporal smoothing. After previous processing, the piano roll of a melody contour has been computed for each time step. We observe two kinds of noise in the extracted melody at this stage. As shown in Figure 3 (a), we observe some small gaps in the predicted pitches (highlighted in the green frame) if the pitch is not recognized in all frames of its duration. This causes a fragmented perception of the melody. As highlighted in the blue frame, the sec-

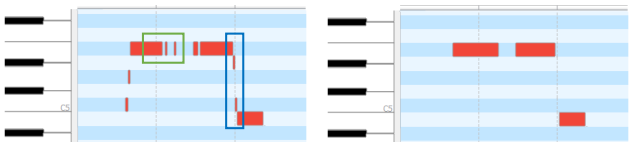


Figure 3: Illustrations of the extracted melody in a MIDI visualizer. The extracted melody after note activity detection and polyphonic pitch extraction still contains some noise that causes unnatural perception of the notification sound (a). We tackle this issue by implementing temporal smoothing (b).

We implement temporal smoothing to deal with these two kinds of noise. More specifically, we fill short gaps between predicted pitches that have the same frequency, and we remove noisy pitches that are only active for a short duration. Experiments show that a gap length shorter than $36\,ms$ and an active duration of shorter than $36\,ms$ for the noisy pitches worked well in our case. Figure 3 (b) shows the resulting piano roll of the melody after this temporal smoothing.

Summary. By processing the activation matrix from the CREPE model using our proposed algorithms for note activity detection, polyphonic pitch extraction, and temporal smoothing, we can obtain a better melody extraction for notification sounds than using the original CREPE method. We provide notification audio samples in the supplement[1] for readers to intuitively experience the difference. Note that the melody extraction method can be applied to audio input with arbitrary timbre, hence it is suitable for a variety of notification sounds and general music pieces.

3.1.2. Timbre Transfer

After extracting the notification melody into its piano roll representation, we convert it into MIDI data. Timbre transfer is then conducted by synthesizing the MIDI data into an audio wave with an arbitrary instrument that is distinguishable by the MIDI interface. For example, we use the sound font *Salamander Grand Piano* [25] for timbre transfer into piano, and we use the *General User GS Soundfont* [26] for other instruments. We implement this synthesis using the open-source synthesizer FluidSynth [27].

3.2. Notification Integration into Music

As illustrated in Figure 2, after notification timbre transfer, the next part is to integrate the notification into the background music. As we intend to provide a noticeable but not intrusive notification experience, we propose the following ideas:

(1) We insert a notification sound at the change of a melodic segment of the background music to not break a note that is being played in the music.

(2) As the notification replaces a part of the background music, we want the musical feature of the music segment being replaced to be similar to the feature of the notification sound.

The first idea requires the background music to be split into segments, each containing a relatively independent melodic structure. To this end, a commonly used methodology is to analyze the structure of time series data by computing its self-similarity matrix (SSM) [28]. In our work, we calculate the SSM of background music by implementing the algorithm described in [17]. This implementation can find boundaries corresponding to the starting and ending spots of repeating structures of the music, which fits our goal of finding melodic segments.

Next, as indicated in the idea (2) above, we aim to find the segment of which the musical feature is the most similar to that of the notification sound. Before calculating the feature similarity, we first apply the following processing to the notification sound:

(1) We align the octave band of the notification sound with the octave band of the most prominent fundamental frequencies in each music segment. Note that although the notification octave might be shifted, the overall melody envelope will not be changed.

(2) We scale the tempo of the notification sound to match the tempo of each music segment by multiplying the notification tempo with a factor of $tempo_{segment}/tempo_{notification}$.

(3) We scale the amplitude of the notification sound to match the amplitude of each music segment by multiplying the notification amplitude with a factor of $max(segment)/max(notification)$, where $max(x)$ corresponds to the maximum observed amplitude in audio x.

After the above three processing steps, a notification sound can be perceived with similar pitch height, rhythm, and loudness as the background music, thus can be blended into the background music more seamlessly. In addition, our initial experiments showed that the similarity calculation was rather sensitive to the above three features, so we aligned the octave band, tempo, and amplitude before the similarity calculation.

Next, we compute the similarity between the notification sound and each music segment based on their chroma features, and finally choose the insertion spot that starts a segment with the highest similarity score. Finally, we insert the notification into the background music at the desired spot with fade-in and fade-out effects that are defined as follows:

$$output[i] = \alpha[i] \times notification[i] + (1 - \alpha[i]) \times music[i]$$

with

$$\alpha[i] = \begin{cases} \frac{i}{12000}, & \text{for } i \in [0, 12000] \\ \frac{i-N}{6000}, & \text{for } i \in [N - 6000, N] \\ 1, & \text{otherwise} \end{cases}$$

where N is the length of the notification sound multiplied with its sampling rate ($16\,kHz$). The parameters in the above formula were set for our implementation in the user study. In a general

case, the fade-in and fade-out duration can be set based on the length and the general musical structure of the notification sound and the background music. Note that a notification sound might be delivered with a delay of up to several seconds since the notification insertion spot is searched in a given range. However, such a delay might be acceptable since most notifications do not require an urgent response. Audio samples of delivering notifications in our method can be found in the supplement[1].

4. USER STUDY

We conducted a user study in an experimental setting to explore users' experience with our notification delivery method. In this study, notification sounds were processed and integrated into background music offline using the methods described in the previous section. For comparison, our study also included the normal notification delivery method that is commonly used on digital devices nowadays, i.e., notification sounds are delivered unchanged at a consistent default volume on top of the music. In our study, we set this constant volume to be hearable, but the highest observed amplitude of the notification sound was lower than the highest observed amplitude of the background music. We refer to the normal delivery method and our method as "standard" and "modified". This user study aimed to answer the following questions:

1. How *noticeable* would users perceive a notification delivered in our method in comparison to the standard method?
2. How much would our delivery method *distract* users from an ongoing task in comparison to the standard method?
3. What aspects would users like and dislike about the standard and our modified notification delivery methods, respectively? And what would these insights indicate for future exploration and design?

"Noticeable" describes how clearly a notification can be recognized. "Distracting" (i.e., intrusiveness) describes how much a notification interrupts the user's thinking process in an ongoing task.

In the wild, it is common that users receive auditory notifications while conducting an unrelated mental activity. To mirror this situation, we designed the study following [7]: participants did the cognitive task of solving anagram puzzles while listening to music, and notification sounds were delivered at arbitrary timestamps. Table 1 shows examples of anagram puzzles. The letters in a puzzle should be rearranged to spell the answer that matches the hint.

In this study, we used the famous piano music composed by Debussy, *Rêverie*, as the background music. To reduce the participants' mental workload, we used one notification sound of approximately $2\,s$ throughout the whole study. Hence, for practical reasons, this study only consisted of one piece of background music, one notification sound, and one target timbre (piano). However, since the study aimed to assess the concept of our notification delivery method, and our proposed approaches to timbre transfer and embedding are not notification-specific, we argue that this experimental setting was adequate, and there is potential to generalize the results to other music and notifications.

4.1. Study Design and Procedure

We implemented a website to conduct an online study, and we deployed the study on Amazon Elastic Compute Cloud. The study involved a section of registration, a section of study introduction, a 1-minute trial task, and three 4-minute formal study tasks.

Table 1: Examples of anagram puzzles. In the study, the participants were asked to solve the given puzzles by rearranging the letters in the puzzle to spell the answer according to the given hint. In the parentheses of the hint it indicates the amount of letters in the answer word or phrase.

Puzzle	Hint	Answer
cheap	a kind of fruit (5)	peach
act	a kind of animal (3)	cat
near gym	name of a country in Europe (7)	Germany
a lac coco	a kind of drink (4-4)	coca cola

During registration, participants answered their gender, age, the average amount of hours they spend listening to music every day, and typical activity contexts (e.g., work, commuting) in which they usually listen to music. After registration, participants would be redirected to the page of the introduction.

The introduction page first explained the study procedure and the rule of anagram puzzles. For participants to get familiar with the background music and the notification sound used in the study, the introduction page also included the audio samples of the music and the notification. In addition, participants had the chance to listen to the examples of delivering a notification in the standard style and in our modified style, thus to intuitively understand the difference between these two notification delivery methods.

Afterward, participants would first conduct a 1-minute trial task that consisted of 15 anagram puzzles to get familiar with the study procedure. Then, they would conduct three 4-minute formal study tasks, each consisting of 40 anagram puzzles. Participants started each formal task by clicking the "START" button at the top of the page. Once clicked, the background music would be triggered, and participants would have four minutes to solve as many anagram puzzles as possible. The notification sound was delivered four times at arbitrary timestamps that were roughly evenly spaced throughout these four minutes. Participants were instructed to focus on the puzzles, and they were not informed whether or how many notifications should be expected during the task. However, they were asked to press the button "I Heard A Notification" immediately when they believed they heard one. When this button was pressed, a text window would pop up that either displayed a hint about the puzzles (e.g., "Tip: Only one letter needs to be moved to solve Q18") or a neutral message (e.g., "You are doing great, keep going"). After reading the text, participants went back to the puzzles by closing the text window. We implemented this pop-up window upon pressing the notification button as an incentive for the participants to remember to acknowledge their recognition of the notification sound [7]. After four minutes, the background music would stop, and an alert window would pop up reminding that the task was done. Afterward, participants would be redirected to the bottom of the page to answer a short questionnaire. After submitting the questionnaire, participants would be redirected to the next study task. They could rest for a few minutes before starting the next task by repeating the above steps.

Each of the first two tasks delivered notification sounds only in the standard method or in our modified method four times. We counterbalanced the order of these two methods among the participants. In the third task, notification sounds were delivered in both methods, each for two times.

4.2. Measurements and Questionnaires

To assess whether participants correctly captured notification sounds, we recorded the timestamps of participants pressing the "I Heard A Notification" button, and we acknowledged successful recognition with an acceptable latency within 5 s from the start of the notification sound.

Participants answered a questionnaire after each formal study task. For the first two tasks that each only involved one delivery method, the questionnaire included the following questions:

(Q1) I think I captured all the notification sounds.

(Q2) I think I recognized the notification sounds immediately when they were played.

(Q3) Overall in this task, rate how *noticeable* the delivery of the notification sound was to you.

(Q4) Overall in this task, to what extent did you feel that the delivery of the notification sound *distracted* you from the anagram puzzles?

Participants answered Q1 and Q2 on a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree). These two questions asked the participants' self-evaluation of their performance in recognizing the notification sounds. They answered Q3 and Q4 on an 11-point scale from 0 (I found it NOT noticeable/distracting AT ALL) to 10 (I found it to be EXTREMELY noticeable/distracting). Remember that Q3 and Q4 were related to the questions we intended to answer with this user study.

Regarding the last task, since it involved both notification delivery methods, we adapted the questionnaire as the following:

(Q1) I think I captured all the notification sounds.

(Q2) I think I recognized the notification sounds immediately when they were played.

(Q3) I clearly noticed the difference between the standard notifications and the modified notifications.

(Q4) If on the scale from 1-11, the *standard* style of delivering the notification sound was as noticeable as a 6, how would you rate the *noticeability* of the modified style in comparison?

(Q5) If on the scale from 1-11, the *standard* style of delivering the notification sound was as distracting as a 6, how would you rate the *intrusiveness* of the modified style in comparison?

(Q6) Do you prefer the standard or the modified style of delivering notification sounds?

(Q7) Regarding the *standard* style of delivering notification sounds, what do you like and dislike about it?

(Q8) Regarding the *modified* style of delivering notification sounds, what do you like and dislike about it?

(Q9) [optional] Do you have any further comments regarding your experience of the notification sounds in this study?

Q1 and Q2 were the same as before. Q3, answered on a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree), aimed to confirm that participants could experience the difference between the two delivery methods. Q4 and Q5 were answered on an 11-point scale with the *standard* delivery method as the neutral reference. We assumed that the answers to Q4 and Q5 could correspond with the participants' assessments in the first two study tasks. Q7-Q9 were open-ended questions where the participants were encouraged to give feedback. During the whole study, a study investigator was connected with the participants via a voice call. The investigator also encouraged the participants to think aloud, and the participants were free to extend their feedback beyond the scope of these open-ended questions.

5. STUDY RESULTS AND DISCUSSION

We recruited 32 participants (14 female, 18 male, age $\in [21, 42]$, $\overline{age} = 28.22$, $SD = 5.35$) for the user study. 50% of the participants listen to music for more than three hours per day; 25% for one to three hours per day; and the remaining 25% for less than one hour per day. The most common activities while listening to music were reported as *doing sports* (84.3%), *work* (78.1%), *commute* (78.1%), and *doing housework* (56.2%), while only 34.3% of the participants would also listen to music for pure *relaxing*.

5.1. Quantitative Analysis of the Results

As for the standard notification delivery method, the participants in general agreed that they captured all the notification sounds (Q1: 4.21 ± 0.61) and recognized the notifications immediately (Q2: 4.18 ± 0.64). Regarding our modified notification delivery method, although the participants were slightly less confident, their ratings were also high (Q1: 4.09 ± 0.58, Q2: 4.06 ± 0.56), and were close to their ratings for the standard style. By investigating the participants' accuracy of capturing the notification sounds, we found that all the standard notifications were successfully captured, while only two modified notifications were missed by two individuals out of a total of 192 test cases.

Figure 4 shows the participants' ratings on the noticeability (Q3) and the intrusiveness (Q4) of the two notification delivery methods in the first two tasks. Remember that each of the first two tasks involved only one notification delivery method, and the order was counterbalanced among the participants. On average, the participants perceived the standard delivery method more noticeable than our modified method (standard: 6.66 ± 1.53, modified: 6.00 ± 1.93). In addition, the standard delivery method also distracted the participants from the ongoing task more than our modified method (standard: 6.09 ± 1.61, modified: 5.25 ± 1.778). We further conducted Wilcoxon signed-rank tests. With a significance level $\alpha = 0.05$, we found that the standard method was rated significantly more noticeable ($Z = -2.599, p = 0.009$) and significantly more distracting ($Z = -2.583, p = 0.01$) than our modified delivery method.

To assess the participants' experience more extensively, we also analyzed their answers to Q3-Q5 for the last study task that involved both notification delivery methods. The participants in general agreed that they clearly noticed the difference between the two notification delivery styles (Q3: 4.09 ± 0.92). When the noticeability and the intrusiveness of the *standard* notification were

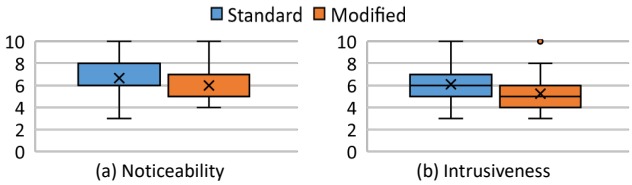


Figure 4: The participants' ratings on noticeability and intrusiveness of the two notification delivery methods in the first two formal tasks. Wilcoxon signed-rank tests show that the standard method was perceived statistically significantly more noticeable ($Z = -2.599, p = 0.009$) and more distracting ($Z = -2.583, p = 0.01$) than our modified method.

set at 6 as the reference, the participants rated these two aspects as 4.87 ± 1.75 and 5.09 ± 1.72 for the modified notification.

With these results, we can answer the first two study questions (see the beginning of Section 4). Our modified delivery method distracted the participants from an ongoing mental task less than the standard method. In addition, although modified notifications were also less noticeable, only two participants missed two modified notifications, and the participants generally believed that they had captured all the modified notifications. This indicates that our method can still deliver notification sounds in an adequately noticeable way.

5.2. Implications from the Open-ended Questions

The participants' feedback to Q6-Q9 in the last questionnaire further revealed their experience and preferences of the two notification delivery methods. From their comments, we can acquire concrete insights and summarize implications for future research.

Out of the 32 participants, 23 reported that they preferred our modified notification delivery method over the standard delivery method. The standard notification delivery method was reported to be *"easy"* and *"clear to recognize"*, while it was also *"distracting"*, and even *"interrupted me thinking and made me feel a bit nervous"* for a few participants. The above experience could partially be because that the standard notification felt like *"an extra layer over the current music"* that did not *"fit in"*. However, since the standard notification was delivered without any change, one participant also commented that *"when the background music is loud, it is also a bit difficult to notice the notification sound"*.

Regarding our modified notification delivery method, some participants clearly stated that it was *"less distracting"* but also *"a bit more difficult to notice"*, which corresponded with the quantitative results. One possible reason could be that the modified notification *"sounds like played by another part of the band, as a component of the whole music"* that *"fits in the music better"*. Moreover, two participants specifically commented that the modified notification sounded more smooth in the music in terms of the *"tonal color"*/*"instrument"* and the *"amplitude"*. A few participants also commented that the modified notification was not difficult for them to recognize because *"the melody is distinguishable from the background music"*. However, several participants also commented that the insertion of the modified notification *"disturbed the whole melody of the piano song* [background music]*"*.

From the participants' feedback, we summarize the following three major implications for future work in this direction.

User habits: Six participants specifically commented that they have been *"familiar with/used to"* the standard delivery that does not change the notification sound at all. Although one of them slightly preferred our modified delivery method after the study, the other five stated that they were *"more comfortable with"* the standard delivery due to this familiarity. However, we argue that there is potential to continue research in the direction of this work, since many participants acknowledged the advantages of our modified delivery method during the study.

Importance of the alert feature: The intrusive nature of the standard notification delivery method might be rather useful for reminding important messages and tasks. One participant commented it as the following: *"I think it's great to mix the notification sound as if it were already in the music like the modified version aims to do. It would be better if the mixing were more 'transparent'. However, if I hope to be notified clearly, I would use the standard style."* Two other participants also expressed a similar opinion. This feedback indicates that it might be important to differentiate between the notifications that alert for an important reminder which users do not want to miss and the notifications that remind something less urgent or important. For the former, users might prefer an intrusive notification alert, whereas our method could be more suitable for the latter situation, in which notifications are still useful to indicate new messages, thus to keep users aware of their surrounding activities, but users can check the information later.

Personalizing the modification parameters: To use our method in the real world, we may need to tweak the notification delivery setting for each user individually, as different people have different sensitivity to notification sounds. While some participants felt that *"the* [modified] *notification and the music were tuned well"* and our method already *"shows a good way to place a notification into music"*, some participants stated that *"the transition from the music to the modified notification is still detectable"* and they suggested making the insertion *"more smoothly"*. This feedback revealed the need for individual design adaptions or setting parameters that can be personalized to users' preferences.

Overall, we conclude that our modified notification delivery method, and the general concept of modifying notification sounds before delivery, could be accepted in many cases. However, the whole notification system can be designed with more flexible settings: (1) Users might choose different notification delivery methods for different events. (2) The parameters for the modified notification delivery can be adjustable according to users' preferences.

6. CONCLUSION AND FUTURE WORK

We proposed a novel method that can gently deliver artificial notification sounds by seamlessly embedding them into the music that a user is listening to, after adjusting the musical features (timbre, amplitude, tempo, octave) of the notification to match those of the music while preserving the original notification melody. To implement this design concept, we extended the CREPE model to extract notification melody and used the MIDI interface to change the original timbre. Moreover, we presented a pipeline that can algorithmically search proper spots in the background music to insert the timbre-transformed notification with fade-in and fade-out effects. We conducted a user study in which 32 participants did cognitive tasks with notification sounds delivered in our method and in the standard method that is commonly used on digital devices nowadays. Our results demonstrated that our notification delivery method provided users with a significantly less intrusive experience, while they could still adequately capture the notification.

As one of the first research efforts that explores a less intrusive delivery method for artificial notification sounds that are widely used on digital devices, this work indicates several directions for future research. First, we are interested in implementing our method on digital devices and conducting studies to explore user experience in the real world. Second, one limitation of our method is that the timbre transfer is currently restricted within single-instrument timbres. Correspondingly, we also used single-timbre background music in this work. Future work can explore notification timbre transfer into multiple instruments to accommodate a larger corpus of music. It is also interesting to investigate the applicability of our method/concept for background music of different styles (e.g., classical, jazz, popular).

7. ACKNOWLEDGMENT

We thank all study participants for their time, effort, and feedback.

8. REFERENCES

[1] P. D. Adamczyk and B. P. Bailey, "If Not Now, When? The Effects of Interruption at Different Moments within Task Execution," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2004)*.

[2] E. Horvitz, M. Cutrell, and C. Eric, "Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance," in *Human-Computer Interaction: INTERACT*, vol. 1, 2001, p. 263.

[3] G. Mark, D. Gudith, and U. Klocke, "The Cost of Interrupted Work: More Speed and Stress," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*.

[4] C. A. Monk, D. A. Boehm-Davis, and J. G. Trafton, "The Attentional Costs of Interrupting Task Performance at Various Stages," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, no. 22. SAGE Publications, Los Angeles, CA, 2002.

[5] S. T. Iqbal and E. Horvitz, "Notifications and Awareness: A Field Study of Alert Usage and Preferences," in *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW 2010)*.

[6] A. Oulasvirta, T. Rattenbury, L. Ma, and E. Raita, "Habits Make Smartphone Use More Pervasive," *Personal and Ubiquitous Computing*, vol. 16, no. 1, pp. 105–114, 2012.

[7] I. Ananthabhotla and J. A. Paradiso, "SoundSignaling: Realtime, Stylistic Modification of a Personal Music Corpus for Information Delivery," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–23, 2018.

[8] F. Kilander and P. Lönnqvist, "A Whisper in the Woods - An Ambient Soundscape for Peripheral Awareness of Remote Processes." Georgia Institute of Technology, 2002.

[9] L. Barrington, M. J. Lyons, D. Diegmann, and S. Abe, "Ambient Display Using Musical Effects," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2006)*.

[10] A. Butz and R. Jung, "Seamless User Notification in Ambient Soundscapes," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2005)*.

[11] R. Jung, "Ambience for Auditory Displays: Embedded Musical Instruments as Peripheral Audio Cues." International Community for Auditory Display (ICAD 2008).

[12] R. Jung, "Non-Intrusive Audio Notification in Emotion Classified Background Music," in *Proceedings of Meetings on Acoustics*, vol. 9, no. 1. Acoustical Society of America, 2010.

[13] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer," *arXiv:1811.09620*, 2018.

[14] J. W. Kim, R. Bittner, A. Kumar, and J. P. Bello, "Neural Music Synthesis for Flexible Timbre Control," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*.

[15] Y.-N. Hung, I. Chiang, Y.-A. Chen, and Y.-H. Yang, "Musical Composition Style Transfer via Disentangled Timbre Representations," *arXiv:1905.13567*, 2019.

[16] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A Convolutional Representation for Pitch Estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*.

[17] O. Nieto and J. P. Bello, "Systematic Exploration of Computational Music Structure Research," in *ISMIR 2016*.

[18] F. Wiehr, A. Voit, D. Weber, S. Gehring, C. Witte, D. Kärcher, N. Henze, and A. Krüger, "Challenges in Designing and Implementing Adaptive Ambient Notification Environments," in *Proceedings of the International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp 2016 Adjunct)*.

[19] J. Ho and S. S. Intille, "Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2005)*.

[20] J. E. Fischer, C. Greenhalgh, and S. Benford, "Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications," in *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2011)*.

[21] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, "A Universal Music Translation Network," *arXiv:1805.07848*, 2018.

[22] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2019)*.

[23] L. Hantrakul, J. H. Engel, A. Roberts, and C. Gu, "Fast and Flexible Neural Audio Synthesis," in *ISMIR 2019*.

[24] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *International Conference on Learning Representations (ICLR 2020)*.

[25] "Salamander Grand Piano," http://freepats.zenvoid.org/Piano/acoustic-grand-piano.html, accessed: 2021-06-17.

[26] "General User GS Soundfont," http://schristiancollins.com/soundfonts/GeneralUser_GS_1.442-MuseScore.zip, accessed: 2021-06-17.

[27] "FluidSynth," https://www.fluidsynth.org/, accessed: 2021-06-17.

[28] J. Foote, "Visualizing Music and Audio Using Self-Similarity," in *Proceedings of the International Conference on Multimedia (MM 1999)*.