CrowdBand: An Automated Crowdsourcing Sound Composition System

Mary Pietrowicz, Danish Chopra, Amin Sadeghi, Puneet Chandra, Brian P. Bailey, and Karrie Karahalios

Department of Computer Science
University of Illinois
mpietro2@illinois.edu, danishchopra@gmail.com, m.a.ssadeghi@gmail.com, puneetchandra007@gmail.com, bpbailey@illinois.edu, and kkarahal@illinois.edu

Abstract

CrowdBand, a sound composition system, demonstrates how a crowd can create works that meet requested criteria and fulfill the aesthetic character given by keyword description and examples. CrowdBand allows flexibility in music composition in terms of duration of the music, completion time and cost of music composition by giving the requestor two modes - thrifty and normal. CrowdBand's workflow divides the composition task into three sections: requesting fundamental sounds, assembling sounds into compositions, and evaluating the results. Based on the crowd workers' responses, we conclude that crowdsourced workers who are non-musicians can design sound and create novel sound compositions through CrowdBand. We also conclude that CrowdBand gives the musically-untrained crowd workers the ability to use common compositional techniques, such as sound layering, vertical stacking of sounds to create harmonic effects, related melodic lines (contrapuntal techniques), and transitions between aesthetic notions, or sound themes. Finally, we show improved, faster results with successive simplification and examples.

Introduction

Music creation has long been the domain of trained composers. Many kinds of compositional styles and notation exist, and they range from traditional notated composition on the musical staff, to completely electronic compositions which have no score, to popular music notated in lead sheet format, to electro-acoustic combinations (partially notated), to improvisational, and more. Yet, all styles require training and practice in order to produce coherent pieces. It is also a time-intensive and costly process for the composer. And, each individual composer has a personal style that defines the music they create. This is both enabling and limiting. It is enabling in that the composer has bounded the problem (he is a certain type of composer, with certain skills, who uses certain techniques, and tends to write certain kinds of pieces, etc.), but limiting in that a single person will not explore the range of what is possible to create. In this paper, we explore how sound compositions can be created by non-expert crowds, because they can quickly explore a range of possibilities and are not limited by the boundaries of any one individual.

Our system, "CrowdBand," assists both musicians and non-musicians in the creation of electroacoustic sound compositions. The *requestors* (system users) specify the kinds of sounds to use, the structure, length, and character of the piece. Our *system* breaks the job down into a workflow of tiny composition tasks that fulfill the request, and the *workers* execute the tasks that the system has generated. Nonmusicians can use CrowdBand to create, for example, a "Happy Birthday" piece for a friend, a signature ringtone, or a sonic work to share on social media. These requestors and workers (who otherwise would not have the experience) get a taste of composing. Musicians, in contrast, can use CrowdBand to explore styles outside of their typical range, explore sounds, and introduce an aleatoric (chance) element to their work. CrowdBand does not *replace* the trained musician.

Figure 1 shows the requestor interface. First, the requestor fills in basic information about the piece that determines the cost, length, and character of the work. In Step 1, the crowd provides sounds in keeping with the duration and keyword descriptions. In Step 2, the crowd assembles the sounds into short pieces. The sounds can be arranged sequentially, simultaneously, and in multiple, related sequences. Finally, in Step 3, a crowd or the requestor select the best result.

CrowdBand's primary research contributions are 1) the system design, and 2) the iterative evaluation of the system. We discuss the system and its evaluation in the body of the paper below. We also address the following

two research questions to analyze the effectiveness of crowd-sourced music:

RQ1: Can crowdsourced workers who are non-musicians design sounds and create musical works?

RQ2: Can crowdsourced, untrained workers create pieces that use any common compositional techniques such as sound layering, counterpoint (multiple related melodies), and transitions between themes?

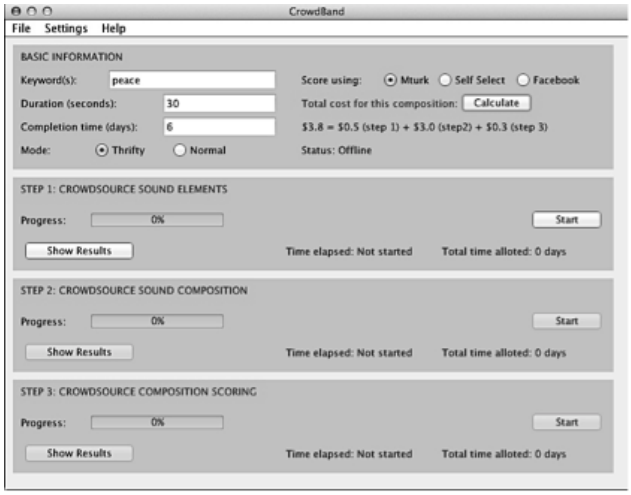

Figure 1: The CrowdBand Composition Interface. This interface shows the basic setup and the 3 steps for crowdsourcing a sound composition.

With CrowdBand, we found that the untrained crowd could provide sounds, which represented concepts given by the requestor, apply some techniques that trained musicians use, assemble the resultant sounds into musical pieces, and select the best productions from all of the workers. We also showed that CrowdBand could be used to create a piece that transitioned from one concept or theme to another, which is a common compositional construct.

Related Work

This project builds on work from multiple disciplines, including music composition, computer music, machine learning, crowdsourcing, and art. Most classical and popular Western music follows the conventions of "tonality," [16] which govern our expectations about how how music should function and sound. The 20th century brought alternatives to the tonal system, and composers created works based on other organizational methods, such as ordered sequences of pitch classes [28]. Varese famously redefined music as the "organization of sound" [31], so that all sounds, not just instruments, could be used; and Russolo pioneered the use of "noise," or "found sounds," in music [26]. Computer music techniques evolved along with technology [25]; and today, many software tools provide computer music support [2,6,10,18,23,24,29]. Composers and engineers experimented with machine learning algorithms too, especially genetic algorithms (GAs), to try to simulate human works. For example, the Spieldose [27] system used a GA with an interactive fitness function to make melody and harmony, and the Melodycomposition[7] system used a GA with a programmatic fitness function to create solo melodies. GAs, however, have limitations for music. A programmatic GA fitness function works well when things can be mathematically represented. But, musical aesthetics and emotion are human judgments, and difficult to capture programmatically. Interactive GAs can do the job, but the burden of human involvement in training the algorithm creates a slow, tedious, limiting "fitness bottleneck." And, most algorithmic GAs work from a small-scale, bottom-up evolutionary way, as in one kind of sound "morphing" into another. Many composers, like Stravinsky and Messiaen, don't work this way and instead juxtapose different ideas, and think at both high and low levels of organization. Getting this range out of an algorithm, and getting it to find interesting usable options is difficult [11]. Keup applied crowdsourcing to address the problem of the fitness bottleneck in interactive GAs for music composition, but did not show that the crowdsourced results were better than results obtained by small groups providing GA training. [14]. Our work, in contrast to the previous projects, uses "found sounds" and electronic composition techniques, and provides a fully-crowdsourced, fully-human, end-to-end system for defining the character of a piece, finding appropriate sounds, arranging the sounds into creative works, and evaluating the results.

Crowdsourcing has been applied to a range of sound projects [3,4,5,13,20,30]. The Gomes [12] survey cites 6 areas for music crowdsourcing, including music co-creation, decision support, music collection management (MCM), marketplace, music release, and crowdfunding. Our work is most concerned with co-creation and decision support, with elements of MCM. World Stage [32] used mobile phones to build large-scale crowdsourced musical experiences; Avicii x You [9] is an online music composing website that allows the crowd to provide the raw materials for sound compositions. The Bicycle Built for 2,000 piece [15] collected 2000 voice recording, and assembled them into a unique rendition of "Daisy Bell."

Previous crowdsourcing technologies paved the way for our work. TurKit [21] provided algorithms on Mechanical Turk (mTurk)[1], and Turkomatic[17] described a system for workflow design. Lasecki [19] described a way to combine multiple streams of real-time annotation output into a single stream. Our current implementation has simple algorithmic and workflow support, and future work will require expanded support for these basic capabilities.

Method: Evolving Task Design

After the user provided basic information shown in Fig. 1, (keywords, duration, cost, etc.), generating a crowdsourced composition involved three steps: 1) getting crowdsourced sounds, 2) getting crowdsourced compositions, and 3) selecting the best composition. We evolved the associated microtasks experimentally, and made small adjustments to improve our results at each iteration. This section describes our approach to improving the results of each task.

Step 1: Getting crowdsourced sounds

This step asked the workers to provide sounds that represented a requestor-defined concept, or theme, such as serenity, anger, darkness, etc. Workers could find available sounds on the web, record their own sounds, or create their own sounds electronically (see Figure 2). Then, these sounds became the set of fundamental components for constructing the piece in Step 2. Note that the selection of basic sounds at this stage was very important for the final compositions. The best assembly job would fall short if the basic sounds were unrepresentative of the concept, or if they were of poor quality. We provided 1-3 examples of each desired concept to guide the worker, and set a limit on the duration of fundamental components so that the length of the components would support the development of a section or piece. Additionally, we required that the sounds be free.

This process sounds simple, but we went through multiple iterations to improve the quality and quantity of usable results. Our initial attempts included a short demographic survey, no examples, and requests for two sounds of opposing quality. The oppositional quality (such as bright vs. dark) served two purposes. First, it gave us the ability to compare the two sets of sounds qualitatively. Because we were working with abstract concepts, we expected variation in interpretation of these concepts and the kinds of sounds that users provided in a given category. Two opposing sets of sounds, however, should be different enough that class membership should be clear, and it would be easier to spot sounds that were obviously out of character compared to the other sounds in the category, regardless of any variation within each category. Second, the collection of oppositional sounds provided the opportunity to create compositions which transitioned from one character to another (e.g., from 'serene' to 'angry'). Composers do this often in practice (even in classical sonatas [16]), and we wanted to see how crowdsourcing could emulate a common compositional process.

We found that with this approach, workers either did not accept the Task, or provided unusable results. Many workers did not read the instructions carefully, and had confusion about abstract concepts (how does "serene" sound, for example). We thought that we might also have some language issues. If a worker was not fluent in English, then interpreting abstract ideas could be difficult.

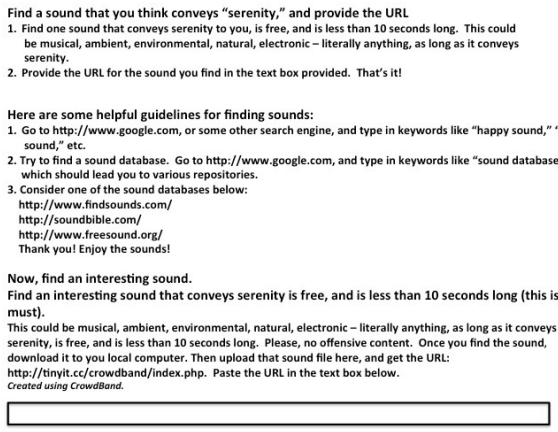

Figure 2. The Sound Collection Task. This simplified version generated the highest rate of usable results.

The next iteration included the survey, a request for two oppositional sounds, and 1-3 examples representing each kind of sound. This improved results, but we thought we could get faster, higher-quality results by simplifying the task. Given the primary goal (find the sounds), the worker tendency to avoid reading detail, and the potential language barriers in a platform like Mechanical Turk, we decided that the survey was not essential. We removed it, and the percentage of usable sounds increased.

For the final iteration in crowdsourced elemental sounds, we asked users to provide "one or more sounds" on a single concept. Workers did not have to process oppositional ideas, but instead focused on one concept, such as finding sounds that represented "brightness". We put the opposing concept (such as "darkness") in a separate Task. This way, we still collected sounds on both concepts, but made the process simpler for each worker and for us to programmatically separate the different type of sounds.

For our initial compositions, we requested sounds for two sets of oppositional concepts: 1) serenity and anger, and 2) brightness and darkness. The output of Task 1 was a set of 15 sounds representing each concept. We chose 15 sounds, because we planned to assemble the sounds into a 20-second composition, or composition section, in Task 2. This was the right amount of sounds to provide enough diversity in the fundamental sounds to be interesting, provide opportunities for assembling many different kinds of compositions, support a 20-second composition (or composition section) well, and not be overwhelming to novice workers who were not trained in music. All of these values (duration, number of sounds, etc.), can be easily adapted in future work. The next section describes

how workers assembled the fundamental sound components into interesting pieces.

Step 2: Getting crowdsourced compositions

This task asked workers to use the fundamental sounds provided in Step 1 to create a 20-second composition, and encouraged workers to use the open source Audacity software [2]. We provided the users with detailed instructions and a YouTube video to help them understand how to use Audacity to do the task correctly. Because we saw significant improvement in the results of Task 1 when we gave examples, we also provided example compositions for Task 2. The users could use Audacity to compose the music melodically (horizontally, left to right on the timeline), harmonically (vertical combinations of sounds at the same time), contrapuntally (multiple lines of related horizontal melody), and in sound layers (multiple horizontal lines of different sounds). See Figure 3.

Figure 3.The Sound Composition Task. This task provided an instructional video, sounds, and example compositions in the download. Simplifying this task generated more usable results in shorter times.

The evolutionary path of this task, however, began without either examples or the instructional video. Adding the video improved results. Later iterations experimented with the amount of payment given to the workers to get the job done. In our early experiments, each HIT took an average of 48 minutes to complete (more than many Turk tasks take), and the task required creativity, so we thought that we would need to pay accordingly (more than Task 1). We got results faster if we paid a little more. Some users seemed to like this kind of Task very much. For example: on worker wrote: *This was an interesting HIT. I was curious if you planned to do more like this?* This means a sound composition Task could be really motivating for crowd workers; it allowed them to express their creativity. In our final attempt towards this Task (figure 2), we added examples and were pleased to hear the music compositions we got from the Turkers, clearly acknowledging that crowdsourced workers who are non-musicians could create unique sound compositions.

Step 3: Selecting the best music composition

This task gave three options for selecting the best composition, including a MTurk task, self-select, and Facebook. If the requestor selected the MTurk option, CrowdBand created a zip of the various composition results obtained in Task 2 and posted HITs that included a pointer for downloading the zip file. This placed the responsibility of evaluation in the hands of the crowd, free from the biases of the requestor, and (often) free of the biases of musical training. The result was a voting function, where the composition with the most votes won. When ties occurred, CrowdBand reported all of the results with the most votes. The self-select allowed the requestor to choose the result himself. This is not objective, but it is often what the requestor wants to do. The Facebook option allows friends to select the best composition. This is an interesting option because it is partially objective. None of the requestor's friends have to participate; and the requestor doesn't control participation. The voters in this scenario, however, are the requestor's friends, an inherently biased population. Because the methods are so different, comparing them does not make sense. Each method has a place, and it is important to understand the built-in biases of each choice.

Exploring Themes

We first explored the emotional themes of "anger" and "serenity" and the abstract themes of "brightness" and "darkness" to show how thematic keywords could be used to inspire crowdsourced sound. We selected these themes because they represented two different classes of keywords, and we thought they would be familiar to everyone. These ideas proved the concept, but were limiting, so we expanded our keyword exploration to include four of the most common classes of themes in sound art, which include *emotional*, *conceptual*, *objective*, and *numeric* keywords. Emotional keywords relate to our internal reactions to human experience. Love, for example, is one of the most common themes in popular music, and it inspired Messiaen's "Turangalila Symphonie". Conceptual keywords are abstract, but nonemotional, like the idea of "large," which motivated Peter Gabriel's "Big Time." Objective keywords point to physical objects or beings in the natural world, like Prokoviev's "Peter and the Wolf," where instrumental solos represent a character in the story. Numeric keywords are important in modern serial techniques [28], such as works by Schoenberg and Boulez, where numeric sequences represent repeated sequences of

pitches, intervals, dynamic levels, and other musical structures. We selected keywords from each category to show that CrowdBand can produce a range of works (see Table 1).

Emotional	Conceptual	Objective	Numeric
Anger	Aliens	Birds	One
Serenity	Brightness	Cats	Multitude
	Darkness	Humans	
	Peace		
	War		

Table 1: Theme Categories and Crowdsourced Works

System Design

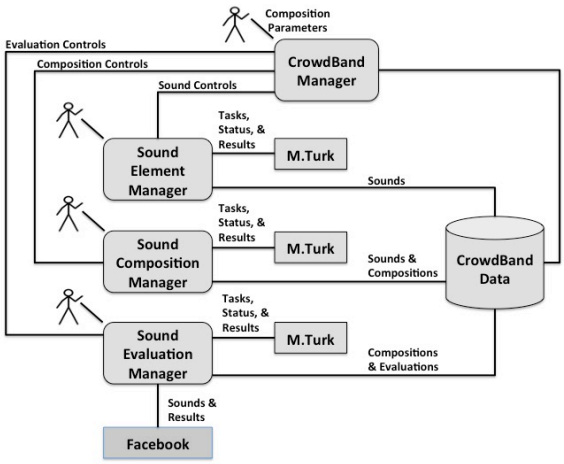

Figure 4: Structural System Overview

Figure 4 shows the CrowdBand system components and their interfaces with Mechanical Turk and Facebook. The CrowdBand Manager is the top-level controller. It accepts keyword descriptions that describe the desired piece, target composition length, desired completion date, cost controls, and evaluation methods to use. Then, it partitions the problem into a series of micro-tasks in finding fundamental sounds, composing small pieces or sections, and evaluating the resulting compositions. The *Sound Element Manager* generates tasks in Mechanical Turk which request sounds, stay within cost constraints, retrieve the results, and present the task status and final results to the requestor. Similarly, the *Sound Composition Manager* generates tasks in Mechanical Turk which ask the workers to use the crowdsourced collection of fundamental sounds to create a small piece. Finally, the *Sound Evaluation Manager* generates Mechanical Turk tasks and Facebook postings which allow workers to select the best resulting compositions. We kept the human in the loop at each stage to provide built-in usability and performance checkpoints, and used this information to make minor adjustments during iterative development that improved the quality of the crowdsourced output at each iteration (we discuss this further in the Results section).

When the requestor launches the job, the CrowdBand Manager partitions it into small tasks and organizes them into the workflow shown in Figure 5 and described step-by-step below.

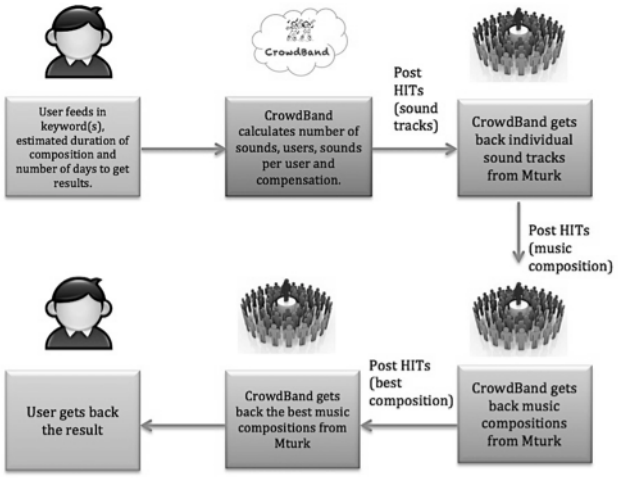

Figure 5: System Workflow

Step1: The requestor inputs one or more keywords, or "themes," of his choice that capture his imagination for the piece. For example, if the requestor wants a composition that portrays peace, he can feed in the keyword "Peace". If he wants his music composition to portray both peace and anger, he can feed in two keywords: "Peace," and "Anger".

Step 2: The requestor tells CrowdBand the desired duration (D) of the music composition. The default maximum duration is ten seconds, and there is no limit on the minimum duration.

Step 3: The requestor tells CrowdBand the number of days (T) in which he wants the results. Based on our experience, we set the minimum number of days required to be at least four. We allow one day to get sounds (T_G), two days to get compositions (T_C) based on the sounds, and one day to get the best composition (T_B) out of several compositions. The default is one week (two days to get sounds, four days to get compositions based on the sounds, and one day to get the best composition out of several compositions). CrowdBand does not, however, *guarantee* results in the desired number of days. We tailor the task to increase the probability of getting results within the desired time frame.

Step 4: For *each* keyword, CrowdBand estimates the number of individual sounds required, the number of users required, and total cost based on duration of the music composition. We give the requestor two options: *Thrifty mode* (which costs less, but has potentially lower-quality,

slower results), and *Regular mode* (which is the recommended cost, but has potentially higher-quality, faster results). We do the following calculations that for *each* keyword (i):

Duration in seconds = D
Number of days = T
Thrifty mode
Days for Task 1 HITs (T_G) = ceil (T * 0.33)
Number of sounds required ($N1_{Si}$) = ceil (D * 0.5) + T_G
Number of users ($N1_{Ui}$) = $N1_{Si}$
Individual compensation = $0.05
Total compensation ($C1_i$) = $N1_{Ui}$ x $0.05
Regular mode
Days for Task 1 HITs (T_G) = ceil (T * 0.33)
Number of sounds required ($N1_{Si}$) = ceil (D * 0.75) + T_G
Number of users ($N1_{Ui}$) = $N1_{Si}$
Individual compensation = $0.05
Total compensation ($C1_i$) = $ $N1_{Ui}$ x 0.05

Step 5: The requestor saves the project, and CrowdBand automatically launches the first HIT to request fundamental sounds, based on the parameters described in steps 1-4. CrowdBand uses MTurk's Java API to post HITs, approve them, and get results. Currently, CrowdBand automatically approves the workers' results, downloads the sounds, and saves them locally. A progress bar shows the status in terms of the number of sounds received from the Turkers. CrowdBand also shows the elapsed time and the total time targeted for requesting sounds. When the requested number of sounds has been received, CrowdBand automatically zips up all of the files received from the Turkers, uploads them to our online repository, and creates the URL to be used in the next step. The requestor may use the Show Results feature anytime to hear the received sounds. Also, the user may start Step 6 manually after Step 5 is 80% complete.

Step 6: CrowdBand requests sound compositions that use the sounds collected in Step 5 for each keyword (i), and waits for the results to arrive. We do the following calculations:

Duration in seconds = D
Number of days = T
Thrifty mode
Days for Task 2 HITs (T_C) = ceil (T * 0.5)
Number of compositions ($N2_C$) = ceil (D * 0.1) + T_C
Number of users ($N2_U$) = $N2_C$
Individual compensation = $ (1/20 * D)
Total compensation (C2) = $ ($N2_U$ x (1/20 * D))
Regular mode
Days for Task 2 HITs (T_C) = ceil (T * 0.5)
Number of compositions ($N2_C$) = ceil (D * 0.15) + T_C
Number of users ($N2_U$) = $N2_C$
Individual compensation = $ (1/20 * D)
Total compensation (C2) = $ ($N2_U$ x (1/20 * D))

Again, CrowdBand automatically approves the results, downloads the music compositions from our online repository, and saves them. CrowdBand displays a progress bar to show the number of music compositions received, and shows the time allotted and elapsed. The user can click the *Show Results* button any time to hear the music compositions. When the Turkers complete all of the compositions, CrowdBand automatically zips all of the Turkers' music compositions, uploads them to our online repository, and creates a link to be used in the next step. The requestor may also start Step 7 manually after 80% of the compositions have been completed.

Step 7: CrowdBand selects the best of all of the compositions, according to the option selected in Step 4. We describe each option below.

Self-select
The requestor selects the best composition.
Crowd select
Turkers select the best composition by voting, and ties are allowed.
We do the following calculations:
Number of days = T
Thrifty mode
Days for Task 3 HITs (T_S) = ceil(T * 0.16)
Number of compositions from Step 6 = $N2_C$
Number of users ($N3_U$) = ceil($N2_C$ /3) + T_S
Individual compensation = $0.05
Total compensation (C3) = $ $N3_U$ x 0.05
Regular mode
Days for Task 3 HITs (T_S) = ceil(T * 0.16)
Number of compositions from Step 6 = $N2_C$
Number of users ($N3_U$) = ceil($N2_C$ /2) + T_S
Individual compensation = $0.05
Total compensation (C3) = $ $N3_U$ x 0.05
Again, CrowdBand automatically approves the Turkers' work, and downloads and saves the results. This stage also displays the progress bar, time elapsed, and total time allotted; and the requestor can click the *Show Results* button at any time.
Friend select
CrowdBand posts a Facebook status update, and allows his/her friends to select the best musical composition.

Results

We describe the results at each step of the process, and then the overall compositional output. The process for requesting sounds grew organically. The first attempt at crowdsourcing primitive sounds (referenced in Table 2, Version I) was the most complex, and provided the lowest ratio of usable sounds to total sounds provided. Version II simplified the overall presentation of the Task, and provided examples so that workers could see at least one type of sound which represented an abstract concept. Increasing simplifications of the Task resulted a larger percentage of usable sounds. Version III removed the

demographic survey, and IV split the Task into two parts – one for each opposing concept.

We observed a steady increase in the number of usable results as we simplified the Tasks, from about 30% usable results to about 90% usable results at the extremes. The rate of completed Tasks also increased. We also saw a slight decrease in the rate of sound production by the crowd as we simplified the Tasks. The simpler Tasks produced about 16% fewer sounds than the more complex Tasks. If necessary, we might be able to compensate for this in the future by simply posting more batches of requests for sounds.

Version Number	Prototype Description
I	-Task description only -No examples provided -Short demographic survey questions included -Oppositional concepts (e.g. "bright vs. "dark") -Request 2 sounds, one for each concept -2 separate entry boxes, 1 for each sound
II	-Add examples for each type of sound requested -Simplify the description - Lighten the tone of the description
III	-Simplify by removing demographic survey -Continue to request 2 sounds, one for each oppositional concept (e.g. "bright" vs. "dark) -Simplify by using only 1 text entry box for all sounds
IV	-Simplify further by requesting one or more sounds about a *single concept (e.g., "bright")* -Put the other concept in a separate Task (e.g., "dark") -Only 1 text entry box for all sounds provided

Table 2. An increasingly simplified series of prototypes for requesting sounds

Version Number	No. of tasks Complete	Avg. no. of Sounds/ Task	Avg. no. of Sounds/ Day	% of usable Sounds
I	10	2	2	30%
II	15	2	6	53%
III	17	1.76	6	74%
IV	17	1.65	2.5*	90%

*Table3. Result of simplification on volume of sounds produced and number of usable sounds. *Recall that in Version IV we split the Task into two parts.*

In Table 3, note that in Stage IV, we split the Task into two parts. The stage III Task asking for sounds representing both "brightness" and "darkness" split into two separate Tasks in Stage IV: one Task requesting sounds about "brightness," and another requesting sounds about "darkness." Since we ran the two separate Tasks concurrently, the combined output of the two Tasks produced a total average 5 sounds per day. This combined average is smaller than the average 6 sounds per day we got in Stage III, but the overall quality of the results was greatly improved. In addition, we note that as the Tasks were simplified, the workers accepted and completed more Tasks within the same amount of time (a 5-day window).

When the workers failed to provide usable sounds, the failures fell into one of the following categories: 1) failed to provide a URL or a sound, 2) provided a URL, but not to a sound, 3) provided a URL to a library of sounds instead of specific sounds, 4) sound was not free, 5) sound was too long, or 6) provided sounds which were clearly outside the character of what we requested. When users failed to provide a URL, they submitted unrelated text, which showed confusion about what they were being asked to do, possibly a language barrier. When they provided a URL to something other than a sound, often it was a link to a music website or a movie which related to the abstract concept (for example, the movie or the band named "Serenity," when asked to provide sounds about "serenity"). The most common unusable results were workers providing a link to a library or index of sounds relating to the concept, and providing sounds which were too long. Users gave us links to entire YouTube performances of famous pieces, such as Beethoven's "Fur Elise."

We were lenient when judging whether a sound was "within character," since users had many ways of interpreting abstract concepts. Some workers selected sounds which made them feel emotions which were similar to the concepts, such as "shimmery" sounds for the concept of "brightness". Some workers were not able to translate an abstract concept such as "brightness" directly into sound, but instead picked sounds relating to visual light. We received sounds of Bic lighters and light sabers for "brightness", and we thought that this was fitting. It worked well and added to the diversity of sounds within the category. Similarly, for the concept of "darkness," we received abstract rumbling, electronic ambient sounds, and sounds of owls and crickets (the abstract vs. the concrete interpretation). What were sounds clearly out of character? The yowling cat and screaming man we received when we asked for sounds representing "serenity."

We first focused on two sets of opposing concepts, and followed these ideas through the entire workflow to produce compositions: 1) serenity vs. anger (in stages I-II), and 2) brightness vs. darkness (in stages III and IV). We experimented with other oppositional concepts, such as cats vs. birds, humans vs. aliens, and war vs. peace. Not surprisingly, when the concepts were more concrete, the workers gave us more results, faster.

When we explored the two other theme categories (objective and numeric), we found that workers most often found actual recordings of the object (birds, cats, and

humans). When a range of sounds was easy to find (like bird calls), we got a diverse set of results. The initial range of sounds we got for the keywords "cats" and "human" was more narrow. Very few workers abstracted the concept, for example, into "birdlike" sounds. Numeric keywords generated an interesting range of results, from recordings of "one" sound, to "lonely" sounds like a howling lone wolf.

We followed a similar, organic process in Task 2 and implemented three increasingly simple versions of the software. Version I was relatively complex, and workers did not accept any of the available tasks for several days. In Version 2, we improved the task description, and added a video to describe the task to the workers in a fun and entertaining way. Although we received one very good result, we learned that we had to relax the requirements for participation (simplify the task, and not require master workers). In the third, and final, version, we simplified the requirements, increased the payment from $1.00 to $1.50 (because we needed results fast), and added a few example compositions to guide the workers. This significantly improved participation, and workers completed all of the tasks within a few hours. Table 4 shows the results.

Version Number	Prototype Description	No. of Tasks Complete
I	-Task description only -No examples provided -Limiting participation conditions -No descriptive video -Difficult uploading procedure -Low Payment ($1 per HIT)	0 of 5
II	-Video description -Increased payment ($1.5 per HIT)	1 of 5
III	-Added examples -Fewer participation criteria	5 of 5

Table 4. Prototype versions of our 2nd Task.

Please see *https://code.google.com/p/crowd-band/* for the fundamental sounds we collected in Task 1, the example compositions provided to the workers, and samples of the resulting compositions we collected from the workers.

Conclusion

Our original research questions were (RQ1) whether non-musicians could design sound and compose pieces and (RQ2) whether crowdsourcing techniques could support any common compositional techniques. Our results show promising answers to both questions. We were able to give compositional micro-tasks, both in sound-finding and sound-assembly, to the crowd, and the crowd produced results which met the aesthetic style and time requirements given in the request. We also showed that it is possible to create compositions which transition from one concept, or theme, to another. We selected oppositional concepts, such as "brightness" and "darkness" to emphasize this result.

Our methods also suggested that giving examples produced more usable results than not, and got results faster. We still got a range of sounds for a single requested concept. The number of duplicated results from the crowd decreased after we gave examples. We also discovered that simplifying the Tasks resulted in a higher level of usable results (from 30-90% over the course of 3 simplifications in technique for crowdsourcing fundamental sounds). Finally, we discovered that money did not drive the motivation to complete the Task. The workers effective average hourly rate was less than $1.50, and one of the workers even wrote us and asked for more Tasks.

Possible work for the future includes 1) improving the compositional assembly interface to support crowdsourced methods and untrained workers better, 2) creating algorithms which both emulate compositional techniques, and leverage the crowd, 3) improved, flexible workflow made specifically for sound design and music composition, and 4) algorithmically checking quality of the results of the sounds and music compositions from crowd workers before automatically approving the results. Improved workflow design will be particularly important to scaling out the design to longer compositions. We expect that workflow techniques that break up a long composition into a series of smaller, simpler Tasks will be more successful than posting fewer, larger, more complex Tasks, particularly given the results of our simplification in crowdsourcing fundamental sounds. The same workflow techniques can also give the requestor the ability to define the structure of a music/sound composition piece in a few simple steps, without having to have training in music composition and sound design.

References

[1]Amazon Mechanical Turk. http://aws.amazon.com/mturk/

[2]Audacity Software. http://audacity.sourceforge.net.

[3]Barrington, L, et al. 2009. User-Centered Design of a Social Game to Tag Music. HCOMP ACM SIGKDD Workshop on Human Computation.

[4]Biles, J. 2007. Evolutionary Computer Music. ch 2, pp. 28-51. Editors Miranda, E. and Biles, J. Springer-Verlag.

[5]Biles, J. 1994. GenJam: A genetic algorithm for generating jazz solos. ICMC.

[6]Common Music. http://commonmusic.sourceforge.net.

[7]Craane, J. 2009. Melodycompostion: application for melody composition using genetic algorithms. http://code.google.com/p/melodycomposition.

[8]DarwinTunes. http://darwintunes.org.

[9]Ericsson, Inc. and Avicii, DJ. 2013. *AVICIIXYOU*. http://www.aviciixyou.com/ CES.

[10]Garage Band. http://www.apple.com/life/garageband.

[11]Gartland-Jones, A., and Copley, P. 2003. The suitability of genetic algorithms for musical composition. Contemporary Music Revie2.

[12]Gomes, C. Schneider, D., Moraes, K. et al. 2012. Crowdsourcing for Music: Survey and Taxonomy. SMC.

[13]Gomes, C. et al. 2013. Cassino Musical: A Game with a Purpose for Social Recruitment and Measurement of Musical Talent. CSCWD.

[14]Keup, Jessica F. 2011. *Computer Music Composition using Crowdsourcing and Genetic Algorithms.* ProQuest Dissertations and Theses.

[15]Koblin, A., and Massey, D. Bicycle Built for 2000. http://www.bicyclebuiltfortwothousand.com .

[16]Kostka, S., and Payne, D. 2012. Tonal Harmony. McGraw-Hill.

[17]Kulkarni, A., Can, M. and Hartmann, B. 2011. Turkomatic: Autmatic, Recursive Task and Workflow Design for Mechanical Turk. AAAI Workshop.

[18]Kyma Software. http://www.symbolicsound.com.

[19]Lasecki, W., Miller, C. Sadilek, A., et al. 2012. Real-Time Captioning by Groups of Non-Experts. UIST.

[20]Lee, Jin Ha. 2010. Crowdsourcing Music Similarity Judgments Using Mechanical Turk. ISMIR.

[21]Little, G. et al. 2010. TurKit: human computation algorithms on mechanical turk. UIST.

[22]Morton, B., Speck, J. et al. 2010. Improving Music Emotion Labeling Using Human Computation. HCOMP Proceedings of ACM SIGKDD.

[23]Max/MSP Software. http://cycling74.com.

[24]Native Instruments. http://www.native-instruments.com.

[25]Roads, C. 1996. The Computer Music Tutorial. The MIT Press.

[26]Russolo, L. 2005. The Art of Noises. Pendragon.

[27]Sanchez, A. et al. 2007. Spieldose: An Interactive Genetic Software for Assisting to Music Composition Tasks. In Bio-inspired Modeling of Cognitive Tasks Lecture Notes in Computer Science, Vol. 4527, pp 617-626.

[28]Straus, J. 2004. Post-Tonal Theory. Pearson.

[29]Supercollider Software. http://supercollider.sourceforge.net.

[30]Urbano, J. et al. 2002. Crowdsourcing preference judgments for melody composition using genetic algorithms. National Conference on Artificial Intelligence.

[31] Varese, E. and Wen-chung, C. 1996. The Liberation of Sound. Perspectives of New Music, Vol 1, No 1.

[32]Wang, Ge, Oh, Jieun, Salazar, Spencer, and Hamilton, Robert. 2011. *World Stage: A Crowdsourcing paradigm for social/mobile music*. ICMC.