TweetDreams: Making music with the audience and the world using real-time Twitter data

Luke Dahl
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
lukedahl@ccrma.stanford.edu

Jorge Herrera
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
jorgeh@ccrma.stanford.edu

Carr Wilkerson
CCRMA - Stanford University
660 Lomita Ct.
Stanford, CA 94305
carrlane@ccrma.stanford.edu

ABSTRACT

TweetDreams is an instrument and musical composition which creates real-time sonification and visualization of tweets. Tweet data containing specified search terms is retrieved from Twitter and used to build networks of associated tweets. These networks govern the creation of melodies associated with each tweet and are displayed graphically. Audience members participate in the piece by tweeting, and their tweets are given special musical and visual prominence.

Keywords

Twitter, audience participation, sonification, data visualization, text processing, interaction, multi-user instrument.

1. INTRODUCTION

Increasing amounts of public social interaction takes place through computer networks. We share jokes, stories, and news, as well as music. Yet these online interactions take place at a distance, separated by screens and transmission delays, whereas music was originally a communal activity amongst people located together in time and space.

TweetDreams is a composition and software instrument which uses real-time data from the microblogging website Twitter[1] to bring co-located performers and audience members into a public and communal musical interaction. Tweets are pulled from Twitter's web server, displayed graphically, and sonified as short melodies. The audience, when enabled with portable computing devices and Twitter accounts, become participants in the piece. They are encouraged to tweet during the performance, and within moments of doing so their words become part of the piece for all present to see and hear.

The overall structure of the piece is controlled by the performers. They interact with the software and modify parameters to control which tweets are retrieved and how they are musically and graphically rendered.

The audience and performers knowingly participate in *TweetDreams*. Yet anyone in the world tweeting during a performance may become an unwitting musical collaborator as their tweets become part of the musical conversation.

[1]`http://twitter.com`

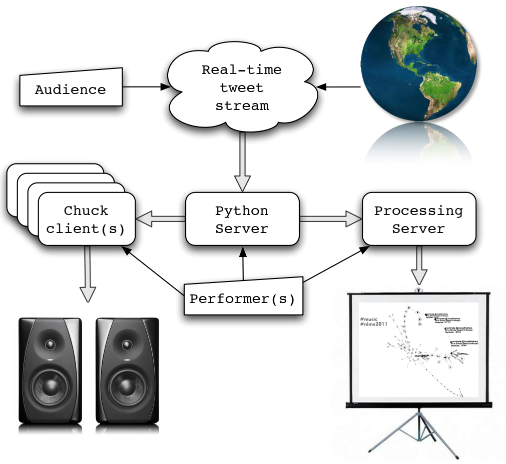

Figure 1: *TweetDreams* interaction overview

TweetDreams is implemented as three main software components: tweets are retrieved from Twitter's servers and processed by a Python application. Tweet melodies are computed and rendered using ChucK. And graphical display is rendered in Processing. Communication between all sub-systems occurs via OSC.

TweetDreams was first performed at the Milano Torino International Music (MiTo) Festival in September 2010, and has been been performed at CCRMA events a number of time since. In each case the performers have been some subset of the authors.

2. BACKGROUND AND PREVIOUS WORK

2.1 Audience Participation

The development of *TweetDreams* began with the desire to include the audience as participants in the music-making process. Audience participation in audio-visual performances has been addressed previously in a variety of ways. The audience's role may be passive yet essential, as in Levin's *DialTones*, where the music consists of the choreographed ringing of cell phones in the audience [11].

Tanaka et al. [12] discuss networked systems that present a *shared sonic environment* where participants are simultaneously performers and audience members. These systems provide simple yet powerful interfaces for creating or modifying sounds, and specific musical knowledge is not required. Barbosa presents a survey of networked digital systems for sonic creation [2].

In Freeman's *Glimmer* [5] there is a clear distinction between performers and audience. The performers are one part of a "continuous feedback loop" consisting of audience activities, video cameras and software algorithms. The au-

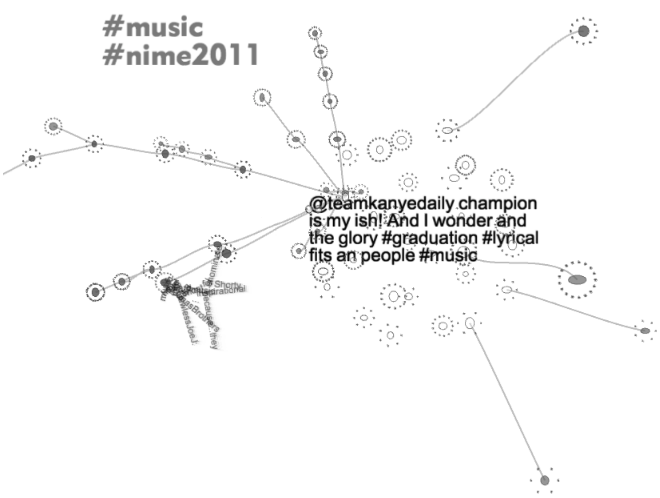

Figure 2: An example of the visualization. A new tweet, and the collapsing text of recently triggered *echos* are visible. *Search terms* are displayed in the upper left.

dience provides feedback via light-sticks that are analyzed by a digital vision system, which then provides instructions to the orchestral performers.

Typically the audience participates synchronously, that is during performance. However they may also participate before a performance takes place by submitting audio-visual materials to be used in the piece, as in *Converge* by Oh and Wang [9] or *Mad Pad* by Kruge [7].

2.2 Multi-user instruments

Jordà [6] uses the term *multi-user instruments* to describe an instrument that is performed simultaneously by multiple people, and lists examples from Cage to the *the reacTable**. He differentiates these from such concepts as *net-music* and *distributed musical systems*, and asks that we imagine instead "a hypothetical acoustic instrument that would invite many simultaneous performers".

Blain and Fels discuss *collaborative musical experiences* [3], and identify a highly restricted musical control as "the main factor common to the design of most collaborative interfaces for novices", allowing anyone, independent of their knowledge of music or the instrument, to participate. They note that with such instruments often the "overall experience takes precedence over the generation of music itself".

In many pieces for laptop or mobile phone orchestra the distinction between an ensemble of instruments and a single multi-user instrument becomes vague. The instrument may be fully distributed as in *SoundBounce* by Dahl, where network messages are used to pass sounds from one mobile phone to another [4], or centralized as in Herrera's *interV* in which a server sends performance instructions to each phone [8].

For *TweetDreams* we wanted to use mobile computing devices as a means to bring the audience into the piece. However, we found that the diversity of devices and operating systems made it prohibitively difficult to distribute a software instrument directly to all audience members. It is similarly difficult to make a web-based instrument which works on all mobile browsers. Due to these limitations we decided to use a pre-existing system to let the audience communicate with the instrument. Twitter, the popular service in which users broadcast short messages in real-time, seemed appropriate.

3. *TweetDreams* ARE MADE OF THESE

How does one make music from data that was originally created as textual statements in a natural human language? One approach would be to interpret the text of each tweet as a code or musical score, and map letters or words directly into musical notes, as in the approach taken by Alt [1]. However this would encourage the audience to compose messages that "play" this mapping, leading to tweets that are no longer idiomatic to natural language. Another approach would be to try to interpret the "meaning" of each tweet, and use that to change musical parameters (e.g. tweets could be given different sonifications based on their emotional valence). While we find this interesting, it is also quite challenging. We chose a different approach.

The music and graphics in *TweetDreams* is based on the idea of *association*. Tweets are grouped into graphs of related tweets, and associated tweets are given similar melodies and linked graphically. By this mechanism the meaning of a particular tweet does not lie in its text per se, but rather in its *network of relationships* to other tweets.

3.1 Associating tweets

The software works as follows: The system queries Twitter for any tweets containing a number of pre-defined *search terms*. One is designated as the *local search term*, and is used to recognize tweets from the audience and give them musical and graphic prominence. The others are *global search terms*, and are used to find tweets from anywhere in the world.

Each incoming tweet becomes a node in a tree-like data structure, where similar tweets are grouped together. When a new tweet arrives it is compared with all previous tweets. If it is sufficiently similar to a previous tweet it becomes the child of that tweet, and the melody for the new tweet is calculated as a mutation of its parents' melody. If a tweet is not similar to any previous tweets it becomes the root node of a new tree, to which subsequent tweets may be added.

The melody of the new tweet is then played at the same moment that its text appears along with a graphical representation of its place in the tree. After a short delay the new tweet's parent *echos*, displaying its text and playing its melody, though now acoustically and graphically attenuated. This cascade of gradually quieter and smaller echos continues up the tree, creating a rippling musical texture of related melodies.

3.2 Music

3.2.1 Calculating Melodies

Each tweet has a melody which is derived from the melody of its parent. A melody consists of six time-steps, each of which may contain a note. A note is specified as a scale degree. The new melody is constructed by a series of random mutations to the parent, where the possible mutations are *transposition* and *swapping*. In transposition a time-step is chosen randomly and the note at that time-step is transposed by a random number of scale degrees. For swapping two time-steps are chosen randomly and their note values are swapped. A total of five mutations are applied, each chosen randomly from transposition or swapping. A mutation may have no effect if the time-steps affected contain no notes, or if a swap occurs between a time-step and itself.

After mutation three checks are performed to insure the melody is well-formed. If the pitch range of a melody is too great it may not be heard as a single auditory stream, so these melodies have their range compressed. Melodies which after many mutations have become too high or too low in pitch are octave-shifted towards the center. Lastly, melodies are shifted in time so that the first time-step contains a note. This simple algorithm leads to a nice amount of variation and similarity between parent and child melodies, creating a distinct family of melodies for a given tree and achieving the desired affect that associated tweets sound similar.

3.2.2 Music Parameters

Any new tweet which is not similar to a previous tweet becomes the root node for a new tree, and its melody is chosen from a set of pre-composed melodies. Root nodes are assigned values for a number of parameters which control how the melody is performed, and subsequent tweets which join the tree inherit these values. The performers control the sonic direction of the piece by choosing which melodies and parameters will be used for new root nodes.

Melodies are synthesized by a simple wavetable synthesizer with a low-pass filter and envelope. The related parameters are `WavetableNumber`, `FilterCutoff`, `FilterQ`, `EnvelopeAttack` and `EnvelopeDecay`. The `Mode` parameter maps scale degrees to specific pitches.

Other parameters control temporal aspects of melody performance: `StepTime` sets the duration of each time-step; `FirstEchoTime` sets the delay between triggering a new tweet's melody and its parent's melody; and `EchoTime` sets the delay between subsequent echos.

Each tweet's auditory spatialization is determined by its `Pan` parameter which is a small deviation from its parent's, creating trees which gradually spread as they grow. The number of reproduction channels can be varied according to the performance venue.

3.3 Server

The Python server (Figure 1) is in charge of handling incoming tweets, adding them to the corresponding tree, and dispatching the necessary information to the visualization and sonification sub-systems.

Incoming tweets are first classified into one of two categories (*local* or *global*) and then appended to the corresponding queue. The queues act as buffers and allow the performers to control the rate at which tweets are displayed and sonified, thereby controlling the "density" of the piece.

Cosine similarity is used to compute the distance between tweets. A Porter stemmer [10] is used to preprocess tweets to account for similar words.

3.3.1 Server Controls

Performers are able to modify the following parameters in the server: i) **Dequeueing rate**: modifies the rates at which the tweets are dequeued and dispatched; ii) **Search terms**: adds or removes search terms; iii) **Distance threshold**: changes the minimum distance required to associate tweets and thus the rate at which new trees are created.

3.4 Graphics

Tweets are displayed both as text and as a 3D graphical representation of the relationships between tweets. The visualization was created in Processing and uses OpenGL rendering to take advantage of hardware acceleration. Each tweet is represented as a circular node surrounded by a number of small "satellites" according to the number of words in the tweet. Links between connected tweets are displayed as slowly moving, slightly animated splines to convey a feeling of liveliness. Alpha transparency is used to reduce occlusion between objects.

The nodes and their links create graphically the trees of associated tweets, and virtual physics is used to animate them. Each tweet node is assigned a mass and a charge, and each node is connected to its parent node by a virtual spring. The charge causes repulsion between nodes and the mass gives them inertia. The springs counteract the repulsion, leading to trees which radiating outwards in all directions. Root nodes are connected by springs to an invisible center point. It is a dynamic system which self-organizes each time a new tweet arrives. Special attention was paid to the physical parameter values in order to avoid instability. The *Traer.Physics 3.0* library[2] was used to implement this 3D force directed layout.

Along with the node representation, the actual text of a tweet is displayed whenever a new tweet arrives or is echoed after the arrival of a new tweet. A differentiating color is used for the first appearance of a tweet. Subsequent echos of the same tweet will display the text again, but with a color scheme that differentiates local tweets from global tweets.

Throughout the performance, the search terms are displayed at the upper left of the screen, reminding the audience of the local search term they must include for their tweets to make it into the piece.

3.4.1 Graphic Controls

Performers are able to modify certain graphics parameters in real-time, to help create visual effects. The parameters are: i) **3D navigation**: moves the camera through the scene; ii) **Link length**: changes the spring constant of connecting springs, which affects the distance between connected nodes, creating the effect of visual "explosions" or "implosions"; iii) **Text size**: makes it possible to adjust the text size on the fly, to ensure that text is readable in spite of the zoom level; iv) **Trace**: controls the transparency of previous visual frames, and allows for a tracer effect; v) **Global gravity**: adds a downwards gravity which counteracts the tendency of trees to radiate.

4. PERFORMANCE

4.1 Form

TweetDreams is not automated. The performers shape the piece by controlling which search terms are used to retrieve tweets, the rate at which new tweets appear, the tendency to create new trees or build on existing trees, the melodies, timbres, and temporal character of tweet sonification, and the physics and perspective of the graphical display.

Although details differ for each performance based on audience involvement and the random nature of the world's tweets, the same basic form has been used each time:

i) **Intro**: Performers begin by tweeting an invitation to the audience to join the piece. Only the local search term is active, keeping the event density low and allowing the audience to easily see their tweets. The timbres are simple, and the graphics are zoomed to a distance that allows a sense of space; ii) **Development**: The world is brought in by adding search terms, and the event density is increased. Musical timbres become more diverse. Once the density is too high to visually track individual tweets, the camera and physics are manipulated to "explore the space"; iii) **Finale**: Search terms are removed and the dequeueing rate decreased until only new local tweets are allowed. Tweet melodies are attenuated until only the reverberated sound is heard, and the camera zooms out to reveal the full constellation of tweets that made up the piece.

4.2 Critique

A short survey was conducted of people who attended a performance of *TweetDreams* in order to understand their

[2] `http://murderandcreate.com/physics/`

experience of participating. About half the respondents reported that they were unable to interact with the piece due to not having either an internet connected device or a Twitter account. Many expressed a desire to participate, and suggested we provide additional means of input such as SMS text messaging.

Some expressed concern about the appearance of offensive content in tweets that made it into the piece. This can be addressed by implementing filtering in the server. Another concern is that while participating in the piece one is also broadcasting tweets to any Twitter *followers* who might be annoyed by the barrage of messages that make no sense outside the context of the performance.

Those who did tweet were engaged in the process of looking for and tracking their own tweets. Some reported that this required so much attention that they could not appreciate what was happening on a larger scale (a variation of Blain and Fels' claim that the overall experience occludes the music itself.) Respondents reported that the instrument responded rapidly to their messages.

Another issue was the visibility of text. Effort was made to keep tweets readable, however as the density of the piece increases it becomes difficult to find one's tweets on screen.

Some people commented on the sonification process, and felt that tweet sounds were too similar. They suggested we map words or letters to pitches to create more variety. We discussed in section 3 why we did not choose this approach, but it raises a point about the nature of this instrument. Audience members do not *play* the instrument in the sense of directly controlling what sounds are made, however their actions trigger musical and graphical events whose details are determined by their actions. It is not necessary that they entirely comprehend the mapping, and we consider this part of the piece's aesthetics.

It seems there are two ways to experience *TweetDreams*. As a participant one engages directly in the communal musical event that is transpiring and provides the materials for it. In this way the piece is *audience-mediated.* However it is also possible to passively enjoy the piece: one can sit back and voyeuristically watch the conversations of the world become music. From this perspective it functions as a type of data sonification. As an anecdote of this capability, at times during rehearsal we became aware of news events or trending topics due to the large tweet trees and similar melodies they generated.

5. FUTURE WORK AND CONCLUSION

Given the audience feedback and critiques discussed above, as well our experience performing the piece, we are considering the following improvements to make *TweetDreams* even more engaging:

i) **Implement better algorithms for calculating association.** For example, being able to derive emotions or other forms of meaning from tweets will allow the system to build more natural associations between tweets. ii) **Use current discussions to add search terms.** Currently the performers decide beforehand which search terms are used in a performance. The instrument would be more flexible if terms could be added during the piece in response to audience tweets. It would be interesting to semi-automate this process, so that new terms are automatically derived from dominant topics in recent tweets. iii) **Increase readability of tweets.** As mentioned in section 4.2, under some conditions tweet visibility is not optimal. New techniques for sizing and distributing tweet text need to be explored. iv) **Add echos up and down the graph.** Currently echos travel up trees. More complex sound textures could be achieved if echo sequences travel in all directions through the graph. v) **Use geo-location.** Twitter provides geo-location data for tweets (if the user allows) which could be incorporated in the piece. vi) **Make an installation version.** The piece was conceived as a performance, but with modifications it could be made into an installation. It could also become an interactive web-based piece, but this would require significant implementation changes.

TweetDreams is a multi-user instrument, a performance piece that invites audience participation, and a sonification and visualization of Twitter data. More significantly, it is a way to bring people who are co-located and spread across the globe into a real-time, communal and public music-making experience. The *instrument* was conceptualized with this goal in mind, but it was also designed to be experienced as a *composition*, possessing an aesthetic unity achieved through the organizing principle of association between tweets.

6. REFERENCES

[1] F. Alt, A. Sahami Shirazi, S. Legien, and A. Schmidt. Creating Meaningful Melodies from Text Messages. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression*, NIME 2010, pages 63–68, Sydney, Australia, 2010.

[2] A. Barbosa. Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation. *Leonardo Music Journal*, 13:53–59, 2003.

[3] T. Blaine and S. Fels. Contexts of collaborative musical experiences. In *Proceedings of the 2003 conference on New Interfaces for Musical Expression*, NIME 2003, pages 129–134, Singapore, Singapore, 2003. National University of Singapore.

[4] L. Dahl and G. Wang. Sound Bounce: Physical Metaphors in Designing Mobile Music Performance. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME 2010, 2010.

[5] J. Freeman. Large audience participation, technology, and orchestral performance. In *Proceedings of the International Computer Music Conference*, ICMC, 2005, Barcelona, Spain, 2005.

[6] S. Jordà. Multi-user Instruments: Models, Examples and Promises. In *Proceedings of the 2005 conference on New Interfaces for Musical Expression*, pages 23–26, 2005.

[7] N. Kruge and G. Wang. MadPad: A Crowdsourcing System for Audiovisual Sampling. In *New Interfaces for Musical Expression*, NIME 2011, Oslo, Norway, 2011.

[8] J. Oh, J. Herrera, N. Bryan, L. Dahl, and G. Wang. Evolving the Mobile Phone Orchestra. In *Proceedings of the 2010 conference on New Interfaces for Musical Expression*, NIME 2010, 2010.

[9] J. Oh and G. Wang. Audience-Participation Techniques Based on Social Mobile Computing. In *International Computer Music Conference*, ICMC 2011, Huddersfield, Kirklees UK, 2011.

[10] M. F. Porter. *An algorithm for suffix stripping*, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[11] M. Sheridan. Composer Calling: Cell Phone Symphony Premiers. *NewMusicBox*, October 2001.

[12] A. Tanaka, N. Tokui, and A. Momeni. Facilitating Collective Musical Creativity. In *In ACM Multimedia 2005 Proceedings, November 6-11, 2005*, pages 191–198. ACM Press, 2005.