Diaz-Marino, R., Carpendale, S. and Greenberg, S. (2005)
Lyric Text. Report iLab-2005-1, Grouplab, Dept. Computer Science, University of Calgary. Duration 3:42. Video plus 2-page paper. May.

LyricText: An Animated Display of Song Lyrics

Rob Diaz-Marino[1]
Department of Computer Science
University of Calgary

Sheelagh Carpendale[1]
Department of Computer Science
University of Calgary

Saul Greenberg[1]
Department of Computer Science
University of Calgary

ABSTRACT

LyricText is an experimental real-time visualization of song lyrics, intended for applications such as Karaoke. By visualizing vocal properties such as pitch and force, we can convey a sense of *how* lyrics are to be sung, rather than just *what* is to be sung. A video illustrating how this appears in practice is available at: http://grouplab.cpsc.ucalgary.ca/papers/index.html

CR Categories: 1.3.8[Computer Graphics]: Applications

Keywords: music, kinetic text, information visualization

1 INTRODUCTION

LyricText is a kinetic text system that visualizes intonation in lyrics using visual variables such as font size, colour, vertical positioning, and motion. The intention is to show viewers not only what is being sung, but also how it is sung. Visual cues convey incidental information about the mood of the singer, and can give insight into their speech and voice patterns. Lyric visualization has specific requirements. For instance, if lyric visualization is meant to encourage interaction from those viewing it, pre-emptive data is required in order to prepare the singer for what they will be singing in the immediate future.

2 BACKGROUND

Text is the means by which we express language when the use of voice is not possible, suitable, or feasible. Most written text, such as that in scientific papers, is usually *well-formed*. It is carefully thought out and written to promote a smooth flow and avoid ambiguity. Speech, on the other hand, is more spontaneous. When we set out to say something, we ordinarily do not have an exact sentence in our head – merely an idea of what we want to express. In fact, transcription of speech often results in text that is riddled with ambiguity, grammatical errors, and awkward wording that would be unacceptable as a formal piece of writing. Why is it so difficult to understand transcribed speech when it is trivial for us to understand spoken language? Birdwhistell [6] suggests that the information conveyed by words amounts to only 20-30% of the information conveyed in a conversation. By transcribing text, we lose a great deal of incidental information: Visual cues such as body language and posture [1], as well as auditory cues such as intonation, pace, volume, and other vocal qualities. All these factors help us to understand sentences that are less than perfectly well-formed.

Recently researchers have started investigating the expressive power of text animation [2,3,4,5]. The concept of Kinetic Typography [3] has been used to set mood and direct attention in Film and Television advertising. The Kinedit System [2] provides a framework for easily producing a wide range of text animations. These include a set of combinable, parameterized animation effects that can be applied to substrings within a larger text body. Kinetic Text has been applied to online messaging applications, presumably to make the expressiveness of these applications more closely resemble that of speech interactions [3, 4, 5]. The main drawback is that additional effort must be put into the manual selection of text animations, which can be an unwanted burden.

Applications such as Karaoke utilize a "bouncing ball" or changing text color to synchronize lyric presentation with music. This text is usually revealed in a chunk so that singers have the ability to read ahead and gain *pre-emptive* information. However, it is assumed that the singer already knows the tune – no information is provided for the musical properties of the song. Timing information is visualized at the exact moment the text is supposed to be sung or at most with a very short offset to allow the singer time to visually process the indication. Long pauses are denoted explicitly (ex. 8-bar instrumental interlude), or with a series of non-text symbols that serve to count down to the next section of lyrics.

3 VISUALIZATIONS

The song data is visualized using font size, colour, vertical positioning, and motion. To indicate one's place in the song, pre-emptive, current, and expired lyrics are shown in sequence in the bar along the bottom of the screen. Expired data is greyed out, current data is highlighted in yellow, and pre-emptive data is displayed prominently in white (Figure 1).

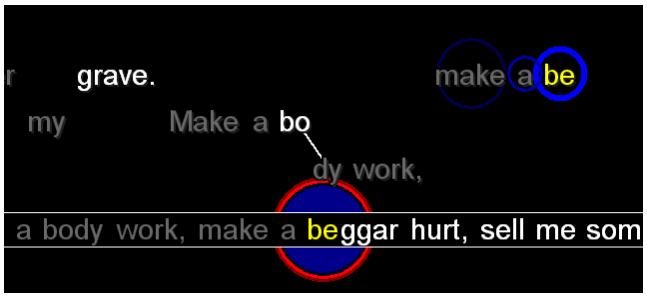

Figure 1: Screenshot of the LyricText application.

For metronome functionality, the beat and measure are visualized as a pulsating dot. The dot flashes blue at the beginning of each beat, while the beginning of each bar or measure is seen as a red ring surrounding the blue dot (Figure 1). As each syllable expires, the text in the main area of the screen is shifted to the left to make room for the next syllable being displayed. Initially the newest syllable appears yellow, but fades linearly. The duration of this fade corresponds to the temporal duration of the syllable.

Maximum and minimum pitch is used to set a vertical range for the positioning of all syllables. The pitch of each syllable is indicated by the relative height at which it is displayed. When different syllables in the same word are sung at different pitches, the word is split over the vertical range. As seen in the Figure, lines connect syllables and provide visual continuity. A syllable that is sung at more than one note jumps to the height of each pitch segment, eventually coming to a rest at the final note.

The beginning of each new segment is marked with a ripple that emerges in the background from the center of the syllable (Figure 2). Color, thickness, and size of the ripple all vary based

[1] e-mail: {robertod, sheelagh, saul}@cpsc.ucalgary.ca

Figure 2: Ripple colors indicate different vocal force levels.

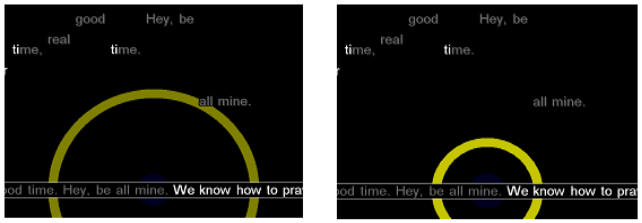

Figure 2: Lead-in indicator visualizes pauses between lyrics.

on the *blend function* and *vocal force*. The duration of the ripples equates to the duration of the current syllable and the next 2 to follow – this allows for 3 ripples to be present on the screen simultaneously, for a more visually pleasing effect than simply one ripple at a time.

Syllables fade to different colors depending on the *blend function* of the last segment. As seen in Figure 1 above, stabbed syllables fade to gray while sustained syllables fade to white. The connecting line takes on the color of the previous syllable's blend function. Less prominently, the segment ripples appear thicker and spread out further for sustained syllables than for stabbed ones.

Vocal force is visualized through the use of ripple color. Rainbow colors correspond to force levels, where *Whisper* is shown as dim purple and *Scream* is shown as vivid red. In the song data used for this project, only the *Spoken* (blue), *Sonorant* (green), and *Yell* (yellow) force levels occur.

When pauses of more than one second occur between syllables, a lead-in indicator is used to visualize the span of time that the singer must wait (Figure 2). This indicator is a thick yellow ring that starts at the edge of the display, gradually becoming more prominent in colour as it converges on the beat and measure indicators. When the ring reaches the centre, it is time to start singing the next section of lyrics.

While this explanation sounds rather abstract, the visual effect and its link to the music, as seen and heard in the accompanying video, is natural, aesthetic, and compelling.

4 Song Data Structure

The two main structures used to express song information in LyricText are *Syllables* and *Segments*. Syllables store *what* is being sung through logical chunks of text, while segments store *how* it is sung, containing properties such as pitch, blend function, and vocal force.

For the creation of LyricText, song data was stored in XML format. The data stored includes general song information such as title, artist, etc., and master tempo data such as *beats per minute* (BPM), *measure* (beats per bar), and *snap resolution* (possible syllables and segments per beat). Lyric data was stored as text divided into verses, lines, words and syllables. Timing data for syllables and segments was stored using bar number, beat number, fraction (of snap), and duration (in snap units). Figure 4 summarizes the main data structures and units used to store song data.

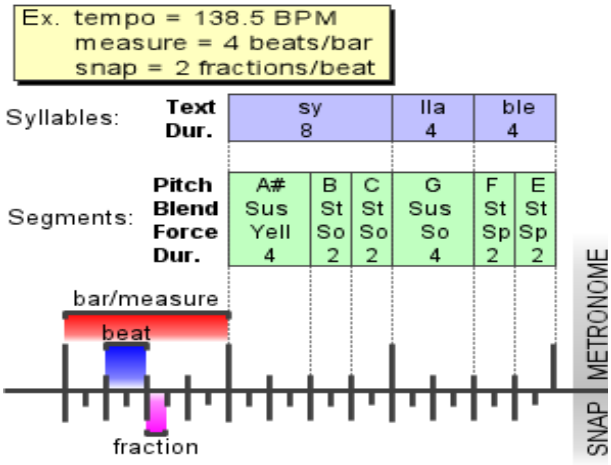

Figure 3: Syllable, Segment, and Timing Diagram

5 Future Work

Even more can be done to improve the cues provided to the singer. Currently, the bar along the bottom of the screen is the only source of pre-emptive data; however it is separate from the visualizations that appear in the main area of the screen. This discontinuity makes it unnecessarily difficult to get both a sense of what is to come and how the current text is being sung. One possibility is to eliminate the need for the bottom bar by placing the pre-emptive text in the main panel. It could appear vertically offset by pitch, and perhaps employ color and shadow to indicate force and blend in a passive, non-animated fashion.

Next, the horizontal space is allocated based on the width of syllable text, not as a scale of time. A user would get a better sense of timing if the X-axis corresponded to the start time and duration of each syllable. Similarly, the user would get a better sense of pitch if the Y-axis showed labelled horizontal lines to indicate notes at each pitch level.

Acknowledgements

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The source music in the video is 'Feel Good Time', by Pink, and is used purely as a demonstration of the visualization.

References

[1] Cassell, J., Vilhjálmsson, H. H., Bickmore, T. (2001) BEAT: the Behavior Expression Animation Toolkit. Proceedings of ACM SIGGRAPH 2001, 477-486.

[2] Forlizzi, J., Lee, J., Hudson, S. E. (2003) The Kinedit System: Affective Messages Using Dynamic Texts. CHI Letters, 5 (1), 377-384.

[3] Wang, H., Prendinger, H., Igarashi, T. (2004) Communicating Emotions in Online Chat Using Physiological Sensors and Animated Text. Proceedings of CHI 2004, 1171-1174.

[4] Bodine, K. Pignol, M. (2003) Kinetic Typography-Based Instant Messaging. Proceedings of CHI 2003, 914-915.

[5] Möhler, G., Osen, M., Harrikari, H. (2004) A User Interface Framework for Kinetic Typography-enabled Messaging Applications. Proceedings of CHI 2004, 1505-1508.

[6] Birdwhistell, R. (1970) Kinetics and Context: Essays on Body Motion Communication. University of Pennsylvania Press, 1970.