Summarizing Musical Preferences as Audio Signatures

Jason Freeman*

*Music Department, Georgia Institute of Technology

Abstract

iTunes Signature Maker (iTSM) uses a feature-driven audio editing algorithm to rapidly generate a short sonic signature of an iTunes music library. iTSM stitches together small segments of songs, driving a concatenative algorithm with spectral features intrinsic to the audio files themselves and with environmental features which describe how those files have been used. This paper describes the software's implementation in relation to the project's objectives: accessibility to a broad audience, accurate representation of users' musical preferences, and interesting and enjoyable results.

1 Introduction

"What music do you listen to?" Nearly every day, someone asks me this question, and I always fumble to find an appropriate response in words. iTunes Signature Maker (iTSM) attempts instead to answer this question in sound, describing the music which users prefer by analyzing the music catalogued by the iTunes jukebox software (Apple Computer 2005), along with the statistics which iTunes tracks about how that library is used. These features together drive a concatenative algorithm that stitches together segments from a user's favorite songs to generate a short sonic signature.

iTSM uses two types of features to drive its algorithm. *Intrinsic features* describe the audio files' actual metadata or content, while *environmental features*, such as play count or rating, describe how those files are used and valued by a person or group of people.

iTSM draws inspiration from a variety of related artistic, scientific, and commercial projects. Feature-driven playlist generators, such as iTunes' smart playlist feature, the Personalized Automatic Track Selection (PATS) system (Pauws and Eggen 2002), and MusicMagic Mixer (Predixis 2005), algorithmically generate playlists based on environmental and/or intrinsic features of a user's music collection, but they operate at a higher level than iTSM; they concatenate entire audio tracks instead of small excerpts, creating playlists that last for hours rather than signatures that last for seconds. Music summarization projects, such as Luke Dubois' Billboard (2005), Brian Whitman's EigenRadio (2003), and my own Network Auralization for Gnutella (Freeman 2003), aim to encapsulate a set of songs through the concatenation, combination, and transformation of segments from them. Schwarz (2006) provides a thorough historical discussion of concatenation and mosaicing techniques in additional areas of artistic practice and scientific research.

2 Objectives

2.1 Accessibility to a Broad User Base

iTSM was commissioned by Rhizome, the online division of the New Museum of Contemporary Art in New York. Since the museum gears its programs towards a wide and diverse audience (New Museum 2005), the accessibility of iTSM was a vital concern. The software had to be easy to install from the Internet and easy to run on any personal computer with iTunes. Furthermore, iTSM needed to be compatible with the short attention spans of most Internet users: it could not take more than a few minutes to load the software, configure its parameters, and generate a signature.

2.2 Accuracy of Representation to Self and Others

iTSM seeks to create signatures which accurately represent the music users listen to, and by extension, which describe something about the users themselves. iTSM users should be satisfied that their signature accurately represents them and feel comfortable sharing it with others. Those who listen to their signature should be able to use it to assess the musical preferences of that person and the compatibility of their musical tastes, even when they are not familiar with any of the musical material included in the signature.

2.3 Interesting and Enjoyable Signatures

As a composer, I realize that no musical or sound object will ever be universally enjoyed, but a significant percentage of people who hear signatures generated by iTSM should find them interesting, enjoyable, and satisfying. The contents of the signature should sound like they belong together, and the signature should have a sense of unity: a coherent formal structure as well as relatively smooth transitions from one segment of audio to the next.

3 Implementation

3.1 Tools and Technologies

iTSM was developed in Java and deployed as a signed Java applet that runs inside of a web browser. It uses the Quicktime for Java API (Maremaa and Stewart 1999) for audio decoding and multitrack mixing. While the use of Java made iTSM much easier for users to download and launch, it also made the implementation of a fast algorithm more challenging: while Quicktime for Java handles audio decoding and multitrack mixing in native code, analysis routines had to be written in pure Java.

3.2 User Experience

To users, iTSM is simply a continuation of the web browsing experience, not a separate activity. Users quickly configure parameters of the algorithm through a wizard interface (Table 1), and then wait for the software to generate their signature. The signature is saved to the local disk as a WAV file, and users can listen to the signature from within iTSM, view a description of its contents, and upload it to an online signature gallery.

id	Parameter Name	Range of Allowed Values	Default Value
a	number of songs to include	10 – 100 songs	20
b	limits on song inclusion	no limits; one per album; one per artist; one per album or artist	one per album
c	ignore podcasts	enabled; disabled	enabled
d	ignore videos	enabled; disabled	enabled
e	primary selection criterion	play count; last play date; date added to iTunes; rating	play count
f	average segment size	0.5 – 6.0 seconds	4.0
g	maximum simultaneous layers	2 – 10	10
h	algorithm optimization	fastest operation; best results	fastest operation

Table 1. User-configurable algorithm parameters.

3.3 Algorithm

The signature-making algorithm proceeds in three stages: a high-level selection process, driven primarily by environmental features, identifies a small group of audio files on which to operate; a low-level selection process, driven primarily by intrinsic features, selects a single small segment from each of those files; and an assembly process optimizes the order of those segments and concatenates and crossfades them to create the signature. Depending on how a user has configured the parameters, signatures can last as little as one second or as long as five minutes.

High-level Selection. During high-level selection, iTSM creates a short list of songs from the user's iTunes music collection for use in low-level analysis and in the signature (Table 1a). iTSM focuses on this small subset of the collection so that it can quickly create a brief signature whose individual segments are long enough to be recognizable.

iTSM selects the songs for the list based on environmental features already monitored for each track by iTunes and stored in an XML file: rating, play count, time last played, and time imported. iTSM's algorithm considers each track in the library in succession, maintaining an ordered list L such that the length of L never exceeds the target size of the subset (Table 1a); L_i always compares favorably to its successor L_{i+1} using the current selection criteria (Table 1e); and any track L_i compares favorably to any track T that has been parsed but is not in L. Streaming radio stations and iTunes Music Store files protected by Digital Rights Management (DRM) are ineligible for inclusion in L, and users may also choose to exclude podcast and video files (Table 1c-d).

When comparing two tracks, iTSM chooses a winner based on the primary selection criterion (Table 1e): the highest rating, most recent play time, highest play count, or most recent import time. For tracks with equal play counts or ratings, the more recently played track is favored. iTunes tracks do not normally have equal play times or import times.

Performance was not a concern with this stage of the algorithm, since iTSM is able to handle thousands of tracks in a few seconds.

Low-level Selection. The low-level selection stage of the algorithm analyzes the audio content of each track in L and chooses a single, short segment from each track to include in the signature. An ideal algorithm would optimize the choice of segments such that, when placed in succession, those segments created the most unified, seamless, and aesthetically pleasing result possible. Not only is it challenging to algorithmically address this ultimately subjective metric, but practical performance concerns also necessitated a far cruder approach. Even in the absence of any audio analysis, simply decoding the MP3 or AAC data and transferring the sample data from the native Quicktime level up to Java takes too long to execute on most computers to meet performance objectives.

iTSM minimizes the amount of audio data it must decode, converting just three sections, each 30 seconds long, from the beginning, middle, and end of each track, into 8 kHz mono PCM buffers. For each audio file, the algorithm divides those sections into segments of equal length. That length is determined for each file as follows:

$$segmentlength = 0.5a + (1.0 - \frac{i}{n-1})a$$

where a is the user-configured average length (Table 1f), i is the file's ranking in high-level selection (beginning from 0), and n is the number of tracks in L (Table 1a).

For each segment, the algorithm computes an average spectrum by performing a 16-frequency-bin Fast Fourier Transform (FFT) on successive 4 ms frames in the segment and averaging them together. The bin size was kept small because this improved the performance of the FFT analysis and, more importantly, of the EMD analysis (below). I would have liked to increase FFT resolution, to transform these spectra using a more perceptually informed model such as Mel Frequency Cepstral Coefficients (MFCCs) and cluster analysis along the lines of recent research (Berenzweig et al 2004), and I would have liked to choose meaningful boundaries to delineate segments, but all of these techniques would have brought additional performance overhead and increased execution time.

To choose a segment from each track, then, iTSM selects the segment whose average spectrum is most similar to the segment it chose from the previous track. (For the first track in the list, iTSM chooses the segment with the highest average energy throughout all frequency bins.) Spectral similarity is computed using the Earth Mover's Distance (EMD) metric, which has been used by Logan and Salomon (2001) in music similarity applications and by Rubner, Tomasi, and Guibas (1998) in image analysis. Conceptually, EMD determines the amount of "work" it will take to transform one spectrum into the other, accounting for the distance between bins in its calculation. In comparing audio segments from arbitrary audio files, EMD provided a more useful metric than fundamental frequency or spectral centroid for facilitating perceptually smooth segment transitions. The computation of EMD is a linear programming problem, and a solution can be found using a simplex algorithm. Though the simplex solution has a worst-case exponential execution time, the low resolution of the spectra (16 bins) ensures that it will always execute with acceptable performance.

Assembly. The selected segment list S is forwarded to the third stage of the algorithm, which optimizes the order of segments within S and mixes them together into the final signature file. iTSM optimizes the order of segments in S using a traveling salesman algorithm in which the distance between segments is the EMD of their average FFT spectra. The traveling salesman algorithm has been used previously in music similarity-based playlist generators (Pohle, Pampalk, Widmer 2005), and though iTSM does not need to loop its signatures, this constraint pushes signature structures towards interesting circular forms. It also enables the optimal order to be rotated; iTSM rotates the segments to put the longest one last, giving the signature's structure a sense of closure.

Finally, the segments are mixed together according to their new order. iTSM crossfades between segments to facilitate smooth transitions; the maximum amount of overlap is determined by the user by specifying the maximum number of simultaneous segments to be played (Table 1g).

4 Evaluation

To evaluate how well iTSM met its objectives (section 2), data was collected from server-side logs, from an informal user study in which thirteen iTSM users answered a series of questions about their experience, and from user comments from hundreds of online forums, weblogs, and social bookmarking web sites.

4.1 Accessibility

Usage Patterns. Over the 17-day period following iTSM's public launch, 21,395 of the 37,801 hits on the entry page (57%) successfully passed the system compatibility check and launched the Java applet, and 13,833 of the 21,395 applet launches (65%) led to completed signatures. This data compares favorably to statistics from other online music projects (Freeman et al 2005) and from e-commerce sites (Horrigan 2004).

Execution Time. The server did not track execution time, but survey participants reported times ranging from 1 to 8 minutes, with an average time of 3.4 minutes. There is always room for further optimization, but iTSM's current implementation does meet performance objectives.

4.2 Accurate Representation

Users' Responses to Their Own Signatures. Survey participants were split nearly evenly as to how well their signatures represented them. Six of the respondents were embarrassed by their signatures — one commented that "embarrassment's part of the fun" — but four participants complained that the high-level selection process was inaccurate because "play count doesn't have a lot to do with my actual musical tastes." Online comments echoed frustrations with the limitations of environmental features used by iTSM. But since iTSM relies on iTunes for this data, the only remedy would be to install a custom monitoring application (which would impact the accessibility of the project).

Online comments revealed an additional problem: iTSM did a poor job for users who purchased most of their music from the iTunes Music Store. iTSM cannot include these tracks even if it would fall under fair use, since any circumvention of digital rights management would violate the United States Digital Millenium Copyright Act.

Signature Representativity to Others. Recognition of signature contents, even if more difficult than in one's own signature, often served as a barometer of compatibility. Social value was placed on recognition, and signatures also fueled discussions about shared tastes. Even when source material was unrecognizable to listeners, signatures still

provided a broad stylistic overview of musical tastes. Several online comments echoed one blogger's description of a signature as "flicking through the radio stations of someone's subconscious" (HalfPie 2005).

4.3 Signature Interest and Enjoyability

Users had mixed aesthetic reactions to their signatures. Descriptors on weblogs and web forums ranged from "beautiful" and "smooth" to "messy" and "a bunch of noise." Some survey respondents complained that they sounded "random" or lacked a "pattern."

All survey participants agreed that the transitions from one song to the next in the signature usually sounded aurally smooth; segment crossfading is probably as responsible for this as low-level selection. Only some participants, though, believed that groups of successive songs in their signatures combined to form coherent musical phrases, or that their signatures had coherent and interesting overall structures to them. At one extreme, a participant described a clear structure: "It began with soft ambient sounds, gradually grew in density, climaxed with a soprano high Bb around the Golden Section, and concluded with the actual end of one the source files, fade out and all." At the other extreme, another participant concluded that "the range of sounds in the signature is too disparate to sound coherent."

While there are certainly shortcomings in the similarity measures that cause some of these signatures to be aesthetically underwhelming, the biggest constraint lies with the 90-second limit for analyzed audio in each track. Especially as segment durations grow longer, the algorithm is left with few segments from which to choose.

5 Future Work

Future improvements to iTSM must enable users to further customize the software to meet their expectations regarding execution time, musical structure, and the recognizability of segments. A more formal user study could ask a larger pool of participants to compare the results of a variety of approaches and algorithms, driving the priorities for future development.

Extraction of analysis data directly from compressed audio formats (Merdjani and Daudet 2003) could eliminate the need to decompress and re-analyze many of the audio files, making it possible to analyze complete audio files (instead of 90-second excerpts) without increasing execution time.

Finally, expansion of the social aspect of the software — by adding commenting, voting, and user profile features to the online gallery or by integrating iTSM into an existing social networking or messaging service — could make it easier for users to incorporate signatures into their online social activities.

6 Acknowledgements

iTunes Signature Maker is a 2005-2006 commission of Rhizome.org. The Rhizome Commissions Program is made possible by support from the Jerome Foundation in celebration of the Jerome Hill Centennial, the Greenwall Foundation, the Andy Warhol Foundation for the Visual Arts, and the New York City Department of Cultural Affairs. Additional support has been provided by members of the Rhizome community. iTunes Signature Maker is available, along with its source code, at http://www.jasonfreeman.net/itsm.

References

Apple Computer. 2005. iTunes 5. http://www.apple.com/itunes/.

Berenzweig, A., Logan, B., Ellis, D., and Whitman, B. 2004. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures. *Computer Music Journal* 28:2, 63-76.

Dubois, L. 2005. Billboard. http://music.columbia.edu/~luke/billboard/.

Freeman, J. 2003. Network Auralization for Gnutella. http://turbulence.org/Works/freeman/.

Freeman, J., Varnik, K., Ramakrishnan, C., Neuhaus, M., Burk, P., and Birchfield, D. 2005. Auracle: a voice-controlled, networked sound instrument. *Organised Sound*, 10:3, 221-231.

Halfpie. 2005. Musical Signature. December 13, 2005. http://halfpie.net/article/539/musical-signature.

Horrigan, J. 2004. The holidays online: Emails and e-greetings outpace e-commerce. *Pew Internet & American Life Project*, Washington, D.C.

Logan, B., and Salomon, A. 2001. A Music Similarity Function Based on Signal Analysis. *Proceedings of the IEEE Conference on Multimedia and Expo*, Tokyo, Japan.

Maremaa, T., and Stewart, W. 1999. *Quicktime for Java: A Developer's Reference*. San Francisco: Morgan Kaufmann.

Merdjani, S., and Daudet, L. 2003. Direct Estimation of Frequency from MDCT-Encoded Files. *Proceedings of the 6th International Conference on Digital Audio Effects,* London, UK.

New Museum. 2005. About the New Museum of Contemporary Art. http://rhizome.org/info/.

Pauws, S., and Eggen, B. 2002. PATS: Realization and User Evaluation of an Automatic Playlist Generator. *Proceedings of the 3rd International Conference on Music Information Retrieval*, Paris, France.

Pohle, T., Pampalk, E., and Widmer, G. 2005. Generating Similarity-Based Playlists Using Traveling Salesman Algorithms. *Proceedings of the 8th International Conference on Digital Audio Effects*, Madrid, Spain.

Predixis Corporation. 2005. MusicMagic Mixer. http://www.predixis.biz/Predixis_Mixer.htm.

Rubner, Y., Tomasi, C., and Guibas, L. 1998. The Earth Mover's Distance as a metric for image retrieval. Technical Report, Stanford University.

Schwarz, D. 2006. Concatenative Audio Synthesis: The Early Years. *Journal of New Music Research* 35:1, 3-22.

Whitman, B. 2003. EigenRadio — The Top 20 Singular Values All Day Every Day. http://eigenradio.media.mit.edu/.