# JASON FREEMAN: ITUNES SIGNATURE MAKER TECHNICAL OVERVIEW

"What music do you listen to?" iTunes Signature Maker (iTSM) seeks to answer this question by generating a short sonic representation of a user's iTunes collection. It considers intrinsic and environmental features to summarize the music she owns as well as the way she listens to and values that music.

The signature-making algorithm proceeds in three stages: a high-level selection process, driven by environmental features, identifies a small group of audio files on which to operate; a low-level selection process, driven 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.
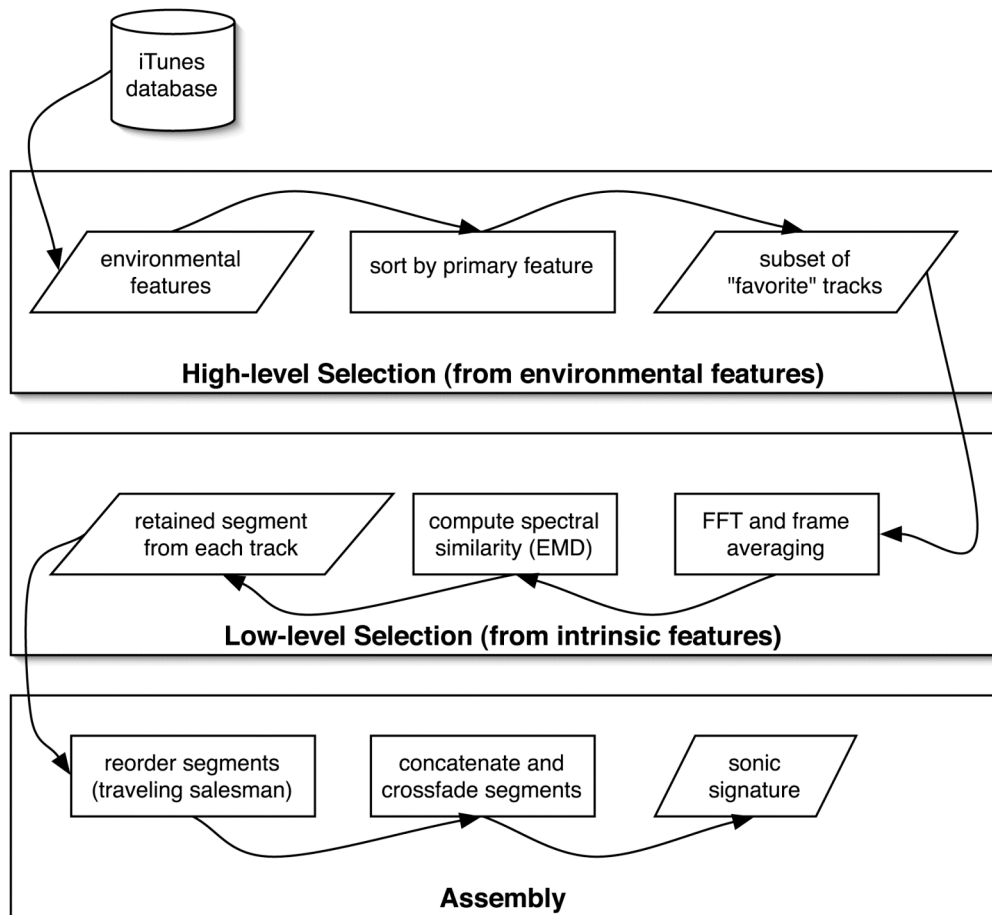
HIGH-LEVEL SELECTION

iTSM focuses on a small subset of the user's iTunes music collection so that it can quickly create a brief signature whose individual segments are still long enough to be recognizable. The user configures the exact size of the subset; it can vary from ten to one hundred tracks.

The iTunes software monitors several environmental features about each song — its rating, the number of times it has been played, the last date it was played, and the date on which it was imported into iTunes — and it stores this information in an XML file. iTSM selects its subset of songs based on these criteria. The user chooses the primary selection criterion, and the remaining features are used to break ties. Users can also choose to limit the subset to include only a single song from any particular artist or album if they wish.

LOW-LEVEL SELECTION

Once a subset of songs has been selected, the algorithm chooses a single small segment of each song to use in the signature. A 16-bin Fast Fourier Transform (FFT) is performed on tiny chunks of source audio; several successive FFT frames are then averaged together to describe a single longer segment. The user configures the duration of that longer segment, which can last between 100 milliseconds and 3 seconds. Higher-ranked songs in the subset use slightly longer segments, and lower-ranked songs use slightly shorter segments.

For the first song analyzed, the algorithm chooses the segment with the highest average energy throughout all of the frequency bins. For the second song, it chooses the segment that is maximally spectrally similar to the segment chosen from the first song.

For the third song, it chooses the segment that is maximally spectrally similar to the segment chosen from the second song, and so on until a single segment has been chosen from each song in the subset. Spectral similarity is computed using the Earth Mover's Distance (EMD) metric.

ASSEMBLY

Before the chosen segments are concatenated, they are reordered to minimize the total EMD spectral distance from one frame to the next. This optimization is treated as a traveling salesman problem; each segment is a city and the distance between two segments is the EMD of their spectra. iTSM uses a genetic algorithm to reach a solution.

Finally, the segments are mixed together. Since iTSM performs no beat detection or accent analysis, it crossfades between segments to create smooth transitions. The user determines the exact amount of crossfading: at one extreme, segments are played with minimal overlap; at the other extreme, all segments are played simultaneously.

USER INTERFACE

Users launch iTSM as a signed Java applet within their web browser and use a simple wizard interface to configure parameters of the algorithm. Once configuration is completed, the algorithm begins to make the signature, and a progress window monitors its status. When the signature is ready, it is saved to the user's desktop. Users can also listen to the signature from within iTSM, view a description of its contents, and share it in an online gallery.

SOURCE CODE

For further details about the implementation of iTSM, please refer directly to the source code, which is available under the GPL.