

AUDIO FINGERPRINT GENERATION

Version 1.2: Dec 22, 2006 9:45 am US/Central

Andre Mosley
Po T Wang
John Broadway
Yu-Heng Lee

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License *

Abstract

Often times, there is a need to represent a long audio signal in a way that it is smaller but yet retains enough information to distinguish it from another audio file. This is useful when implementing a system that compares audio files in order to determine their identity. This module explains how we used a particular audio fingerprinting method in our audio recognition system.

Audio Fingerprint Generation

In order to provide a more compact representation of our audio signals, we researched methods of audio fingerprinting. The method we used, which appeared to be very effective, was one that compared the amount of energy in different frequency bands. This method takes advantage of the fact that the human ear can only distinguish audio frequencies that are very different from one another. There are 25 audible frequency bands, also known as critical bands, that vary in width and range from 0 to 20kHz. The human ear can be modeled as a series of 25 band pass filters. As long as two frequencies fall within the same critical band, they are generally indistinguishable. Therefore rather than keep the entire spectrum for each one of our audio signals we can examine the patterns of energy variations from one critical band to the next, which only requires retaining one value, (the total energy), for each critical band.

To implement this system in Matlab, we first divided the signal into time frames using Hanning windows of about 37 ms. The purpose of using Hanning windows is to prevent extraneous oscillations from occurring in the spectrum. We also overlapped the Hanning windows in order to prevent the amplitude modulating effect of consecutive Hanning window.

Figure 1: Audio Fingerprint Generation scheme.

Each time window goes through this process.

The Fast Fourier Transform (FFT) is taken for each windowed timeframe and the generated spectrum is divided into the 25 critical bands. The energy is calculated for each critical band and energy difference between two consecutive bands is stored into an array of length 24. This process is done for each timeframe which results in a matrix of 24 rows

*<http://creativecommons.org/licenses/by/2.0/>



Figure 1

and one column for each timeframe. Within each row of the matrix, consecutive values are compared. If the value increases from one column index to the next, the value in the prior column is replaced with a 1. Otherwise it becomes a -1. This column to column, or timeframe to timeframe, comparison examines energy fluctuations similar to the way we compared energy differences between consecutive critical bands. This overall scheme results in a matrix of dimensions 24 x (timeframes -1). This set of 1s and -1s is the audio signal's fingerprint. The only thing left to do once is compare audio fingerprints to find a match and return song information.