Query by Tapping

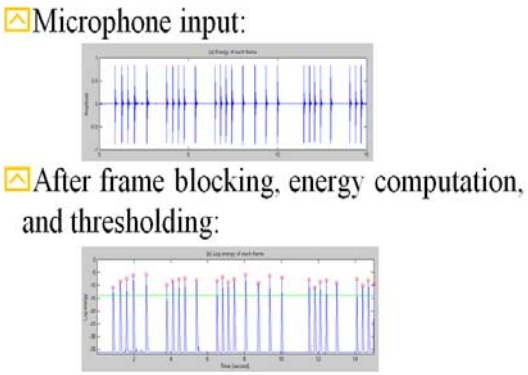
Chun-Ta Chen and Jyh-Shing Roger Jang

Multimedia Information Retrieval Lab, CS, NTHU

Feature Vector Extraction

The feature vector we used is simply a time vector in which each element represents the duration of a note. To extract such information, the user is required to gently tap on the microphone to indicate the beat information of the intended song.

A typical waveform of the user's tapping input is shown in plot (a) of the following figure; the corresponding log energy profile is shown in plot (b). To extract the duration of each note, we need to do frame blocking first and then find the energy of each frame. The circles in (b) indicate where local maxima of the log energy are located.



The way to extract the onset time involves the following two steps:

1. Keep the volumes which are higher than a volume threshold.
2. Apply a moving window to keep the volumes that are maximum within the moving windows.

Once the legal local maxima are found, the duration of each note is equal to the distance between two neighboring local maxima. The beat information is then represented as a timing vector in which each element is a note's duration.

Comparison Procedure

Once the timing vector from a user's input is obtained, we need to compare it with those of the songs in the database. We propose a comparison procedure based on dynamic programming. To match the input timing vector against those of the songs in the database, we need to be aware of two things:

1. The tempo of the user's input is usually different from those of the candidate songs in the database.
2. The user is likely to lose notes instead of gaining notes.

To solve the first problem, we need to normalize the input timing vector and those of the candidate songs. Suppose that the input timing vector is represented by vector t of length m, and the reference timing vector represented by r with length n. Usually n is greater than m. Suppose that the user did not lose/gain any notes, so we only need to compare t with the first m elements of r. However, since the user might lose or gain notes, we need to compare t with a selection of different versions of r with different lengths. Suppose that the first q elements of r are selected for comparison, and then the normalization step convert both vectors to a total duration of 1000:

$$\tilde{t} = round(1000 * t / sum(t))$$
$$\tilde{r}_q = round(1000 * r(1:q) / sum(r(1:q)))$$

In the above equations, r(1:q) indicates a vector formed from the first q elements of vector): sum(t) indicates the summation of all elements in vector t. The operation of round rounds all elements of the vectors into integers. The purpose of multiplication by 1000 is to guarantee high resolution in fixed-point computation. After the above normalization step, the subsequent comparison procedure is based on t ~ and r~. Actually, we need to vary the value of q to get different versions of r~; the distance between t~ and r~ is taken to be the minimum of distances between t~ and all variants of r~. In our method, the value of is varied from p-2 to p+2, where p is the length of t~.

The comparison procedure is based on the concept of dynamic time warping (DTW). For notation simplicity, we shall remove the "tilde" temporarily. Suppose that the (normalized) input timing vector is represented by t(i), i=1,…,m, and the (normalized) reference timing vector by r(j), j=1,…,n.
Optimal value function:

$$D(i,j) = \min\begin{Bmatrix} D(i-1,j-2) + |r(j-1) + r(j) - t(i)| + \eta_1 \\ D(i-1,j-1) + |t(i) - r(j)| \\ D(i-2,j-1) + |t(i-1) + t(i) - r(j)| + \eta_2 \end{Bmatrix}$$

D(I, j) is the minimum accumulated distance starting from (0, 0) of the DTW table to the current position.

Find the DTW distance between the input timing vector and that of each candidate song in the database. Note that the timing vector of a song has to be compressed or extended to have 5 versions of different lengths. Then the distance between the input timing vector and that of a song is taken to be the minimum among all 5 distances between the input timing vector and 5 variants of that of the song. The equation is

$$dist(t,r) = \min_{q=p-2\sim p+2} D(\tilde{t},\tilde{r}_q)$$

After comparing the input timing vector with each timing vector in the song database, we can rank the results according to DTW distances.