A Query-by-Singing System based on Dynamic Programming

Author : Jyh-Shing Roger Jang, Ming-Yang Gao

Professor : 陳嘉平

Reporter : 楊治鏞

Introduction

- The system, known as CBMR (Content-Based Music Retrieval), facilitates the content-based song database retrieval via users' acoustic inputs.

- This paper presents a query-by-singing system that is based on two levels of dynamic programming as its comparison engine.

Input Collection

- The acoustic input is recorded from a PC microphone directly with a length of 8 seconds, sample rate of 11025, 8 bit resolution and single channel (mono).

Pitch Tracking

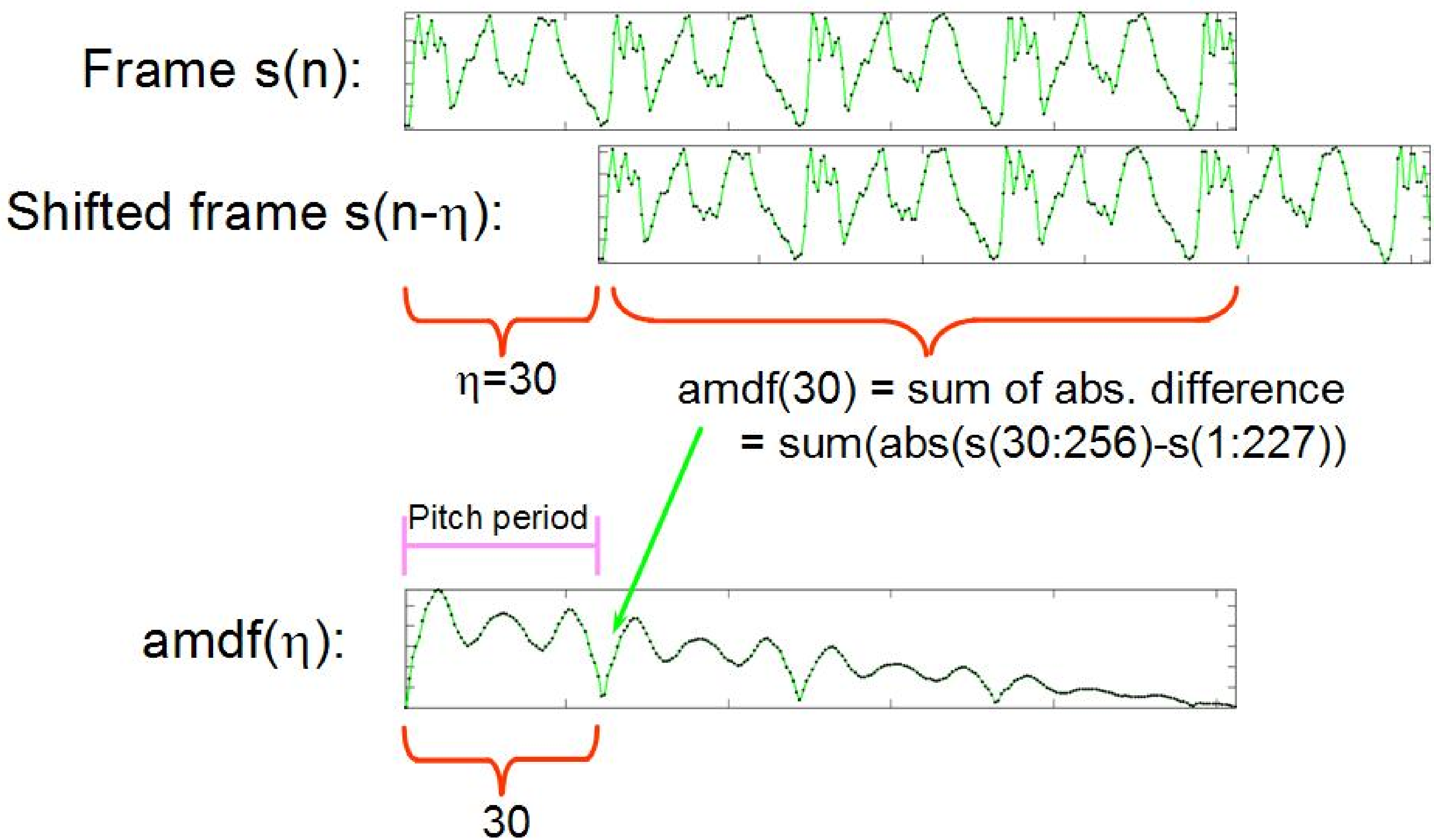
- The acoustic input is first put into frames of 512 points, with 340 points of overlap; this corresponds to 1/64 second for each pitch frequency.

- Then every 4 pitch frequencies are averaged to merge into a single frequency, thus the final pitch vector has a time scale of 1/16 second.

Average magnitude difference function (AMDF)

- $$AMDF(\eta) = \sum_{i=0}^{n-1} \left| S(i) - S(i-\eta) \right|$$

- where η is the time lag in terms of sample points.

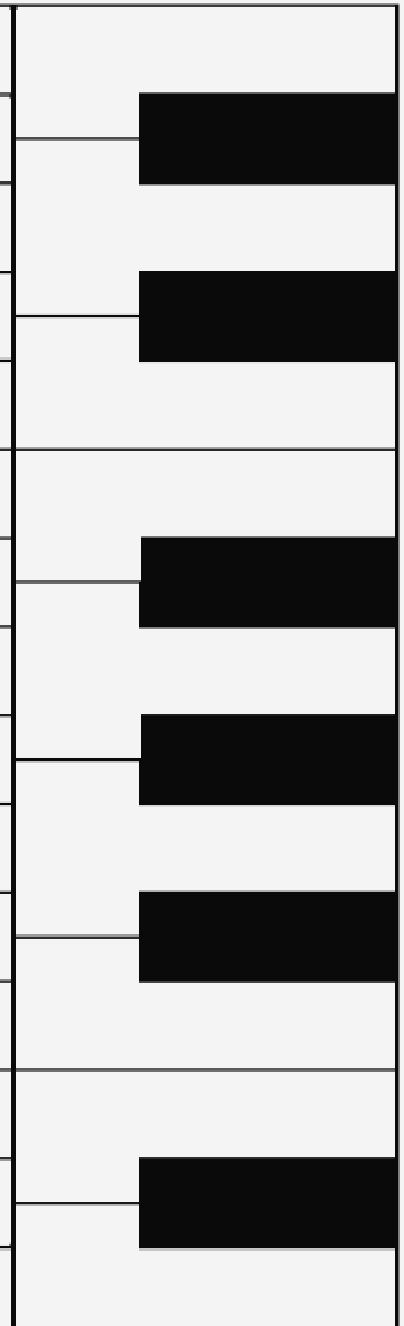

Frame s(n):
Shifted frame s(n-η):
η=30
amdf(30) = sum of abs. difference
= sum(abs(s(30:256)-s(1:227))
Pitch period
amdf(η):
30

Semitone

- After obtaining the pitch frequencies, we use the following formula to transform them into the representation of semitone:

$$semitone = 12 \times \log_2\left(\frac{freq}{440}\right) + 69$$

C1	Do	261.63
$C^{\#}$	$Do^{\#}$	277.19
D	Re	293.67
$E^{\flat}$	$Me^{\flat}$	311.13
E	Me	329.63
F	Fa	349.23
$F^{\#}$	$Fa^{\#}$	370.00
G	So	392.00
$A^{\flat}$	$La^{\flat}$	415.31
A	**La**	**440.01**
$B^{\flat}$	$Ci^{\flat}$	466.17
B	Ci	493.89
C2	Do	523.26

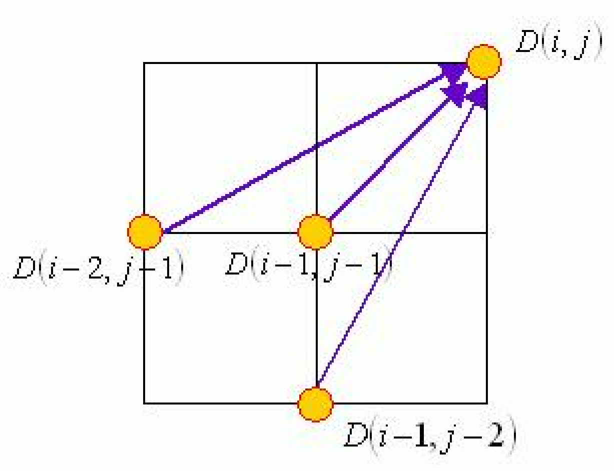


Pitch Smoothing

- Unvoiced segments and random noise can cause unreasonably high pitch.
- If the energy level is lower than a threshold, then the corresponding pitch semitones are set to zero.
- Also if the identified pitch semitones are higher than 84 (or 1047 Hz in frequency), they are also set to zero.

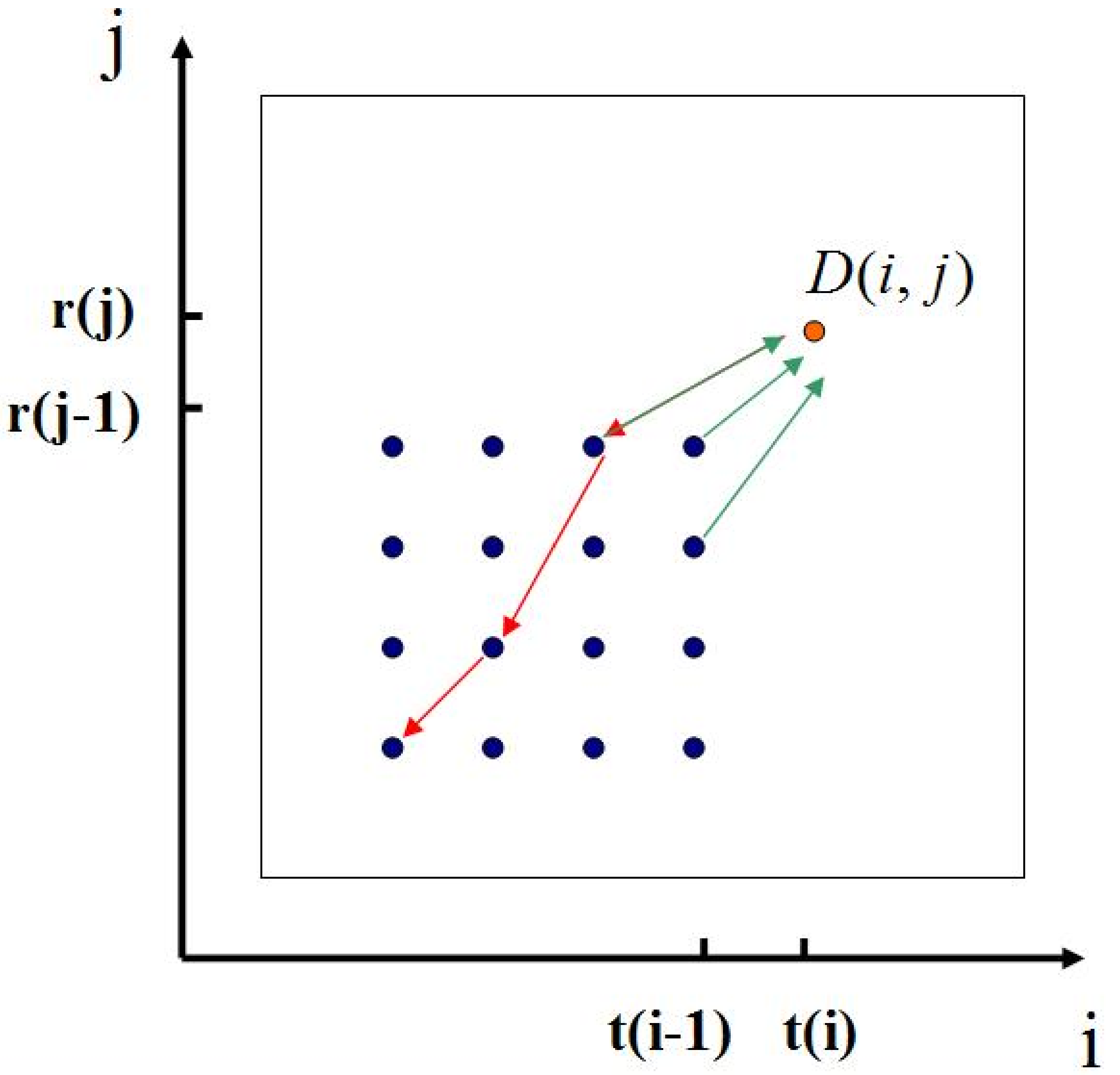
Dynamic Time Warping

- input pitch vector $t(i), i = 1, \ldots, m$
- reference pitch vector $r(j), j = 1, \ldots, n$



$$D(i, j) = d(i, j) + \min \begin{cases} D(i-2, j-1) \\ D(i-1, j-1) \\ D(i-1, j-2) \end{cases}$$

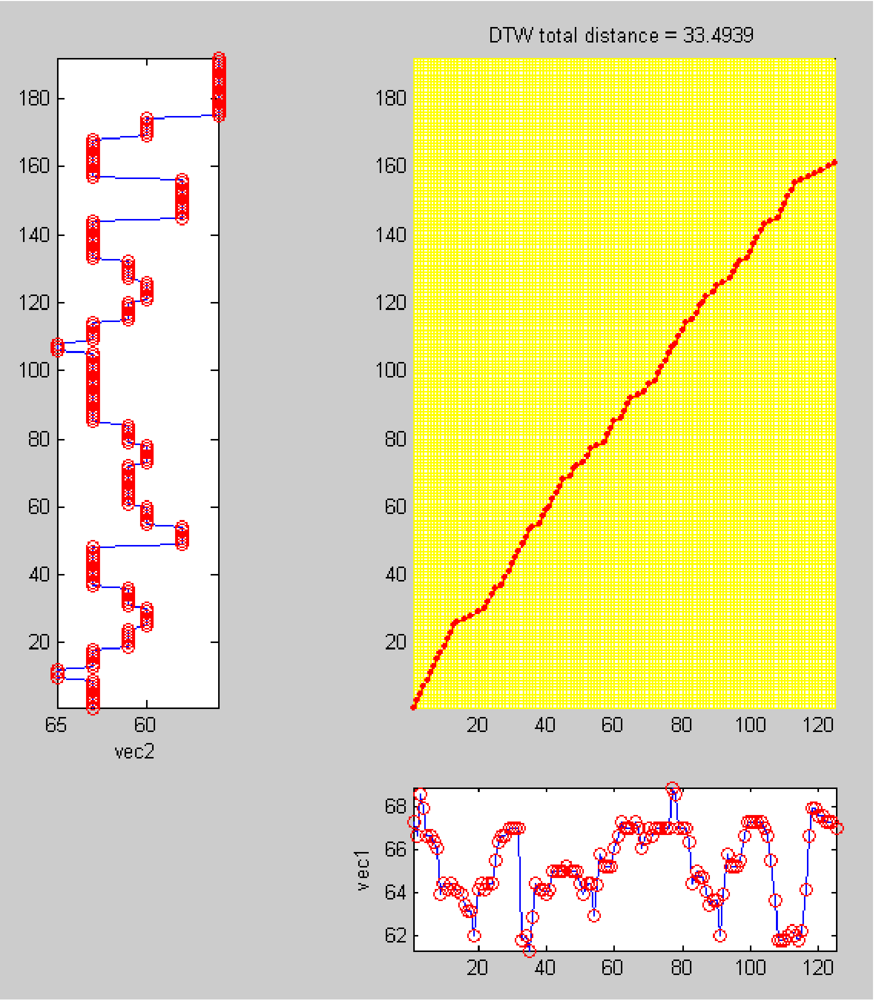
$$d(i, j) = \left| t(i) - r(j) \right|$$


j
r(j)
r(j-1)
$D(i, j)$
t(i-1)
t(i)
i

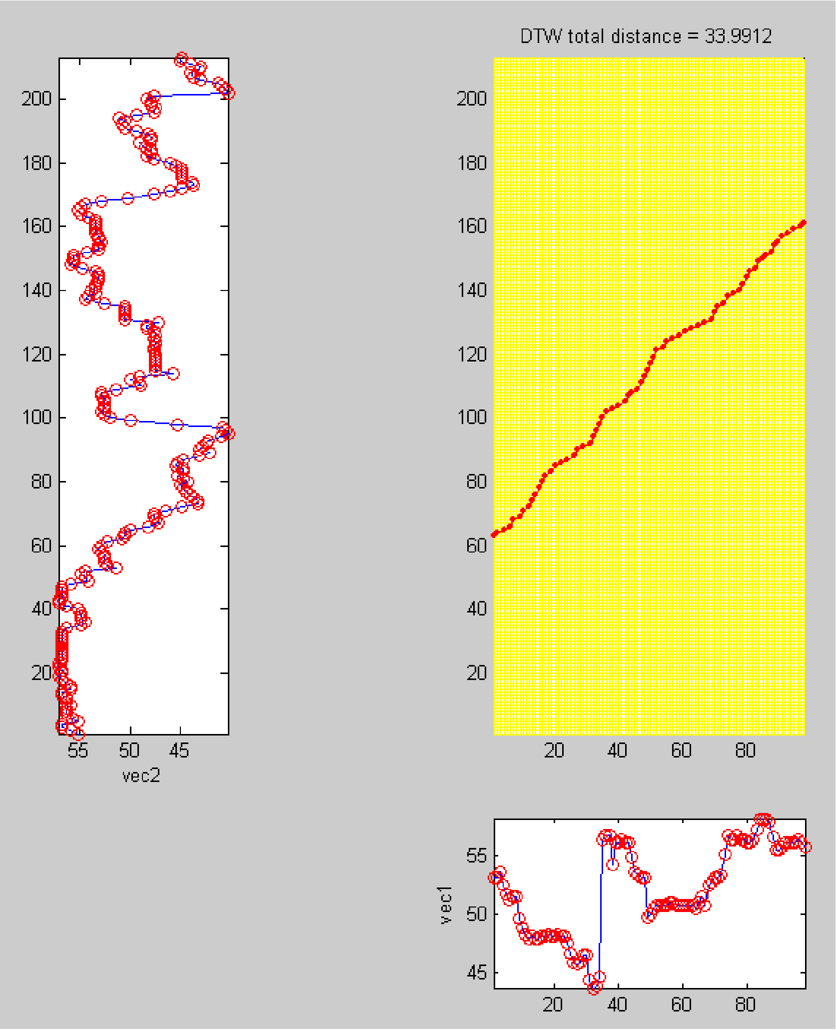
Boundary conditions

- $D(i,1)=\infty, i=2,..,m$
- $D(1,j)=\left|t(1)-r(j)\right|, j=1,\ldots,n$
- The first equation ensures that the optimal DTW path never starts from the middle of the test vector.
- The second equation indicates that the optimal DTW path can start from anywhere in the middle of the reference vector.

DTW Path of “Match Beginning”



DTW Path of “Match Anywhere”



key transposition

- Besides constructing the DTW table for computing each similarity scores, we still need to deal with the problem of different keys for different users.

$$\begin{cases} span = 4 \\ center = 0 \\ t = t - mean(t) \\ r = r - mean(r) \end{cases}$$

- (The last two equations make both t and r zero mean)

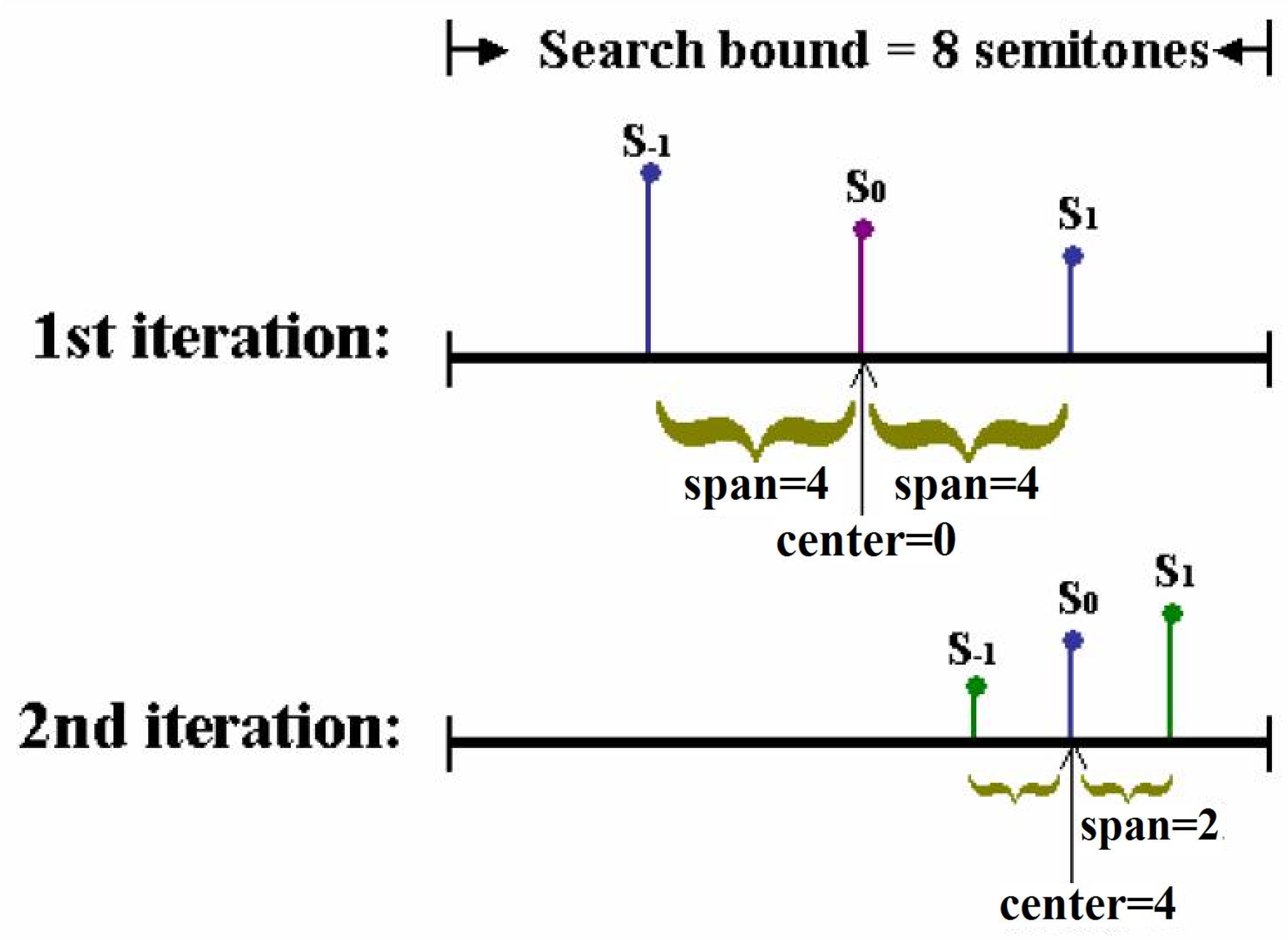
key transposition

- $$\begin{cases} s_{-1} = dtw\left(r, t - center - span\right) \\ s_0 = dtw\left(r, t - center\right) \\ s_1 = dtw\left(r, t - center + span\right) \end{cases}$$

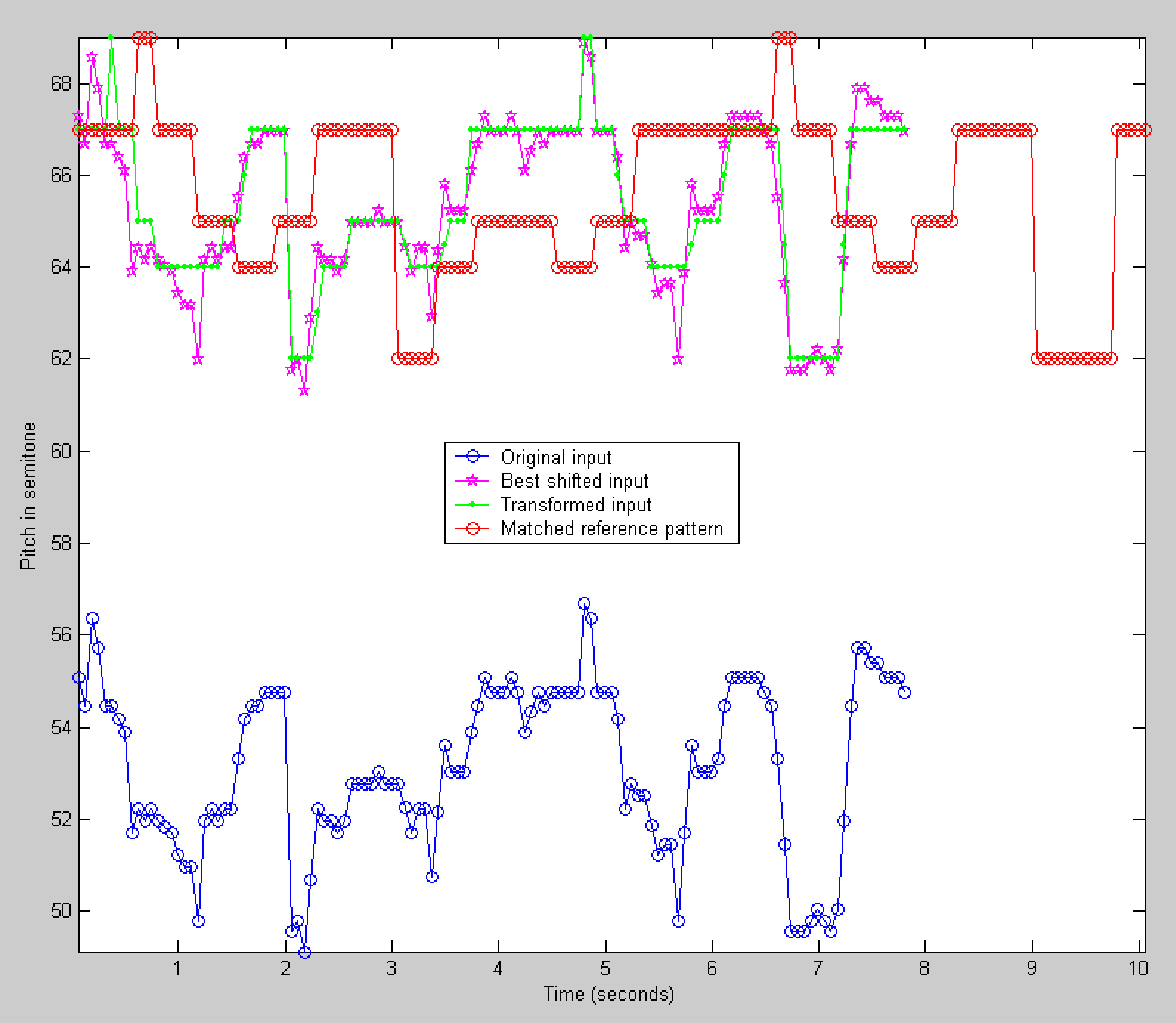
- If $s_{-1} = \min\left\{s_{-1}, s_0, s_1\right\}$,then $center = center - span$

else if $s_1 = \min\left\{s_{-1}, s_0, s_1\right\}$,then $center = center + span$

- If $span > 2,\ span = span / 2$


Search bound = 8 semitones
1st iteration:
S-1
S0
S1
span=4
span=4
center=0
2nd iteration:
S-1
S0
S1
span=2
center=4

Example of Key Transposition



Performance Evaluation

- We have around 200 recorded clips of songs sung or hummed by 14 persons (9 males, 5 females) ,each recording takes from 5 to 8 seconds.

- There are about 800 songs in the database.

- We divide the performance evaluation into two parts: one with the use of a single DTW for computing each similarity score, the other with the use of five DTWs to include key transposition.

Test of 1-DTW

- The top-20 recognition rate is 84%, the top-3 recognition rate is 75%, and the top-1 recognition rate is 66%.
- The average response time for each recording is about 1.557 seconds.

Test of 5-DTW

- The top-20 recognition rate is 86.5%, the top-3 recognition rate is 84%, and the top-1 recognition rate is 76%.

- The average response time for each recording is about 2.556 seconds.

Test of Combining 1-DTW and 5-DTW

- An intuitive idea to improve the system is to take a hierarchical approach that uses 1-DTW to filter out 700 of the 800 songs, and leave only 100 songs for 5-DTW to do a detailed comparison.
- The top-20 recognition rate remains 85%, the top-3 recognition rate is 83.5%, and the top-1 recognition rate is still 78%.
- In other words, the performance is almost the same as that of 5-DTW, but the response time have been effectively reduced from 2.556 to 1.765 seconds.