Computer Assignment HW4 Report

B00901087 潘柏丞

1. Goal:

Following the paper: **A Query-by-Singing System based on Dynamic Programming**, I try to make a hierarchical approach for combining DTW-based comparison engines and CBMR system. Then creating a query-by-singing system first takes the user's acoustic input from a microphone and converts it into a pitch vector. Then two levels of comparison procedures, both based on the concept of dynamic programming, are invoked to compute the similarity between the user's pitch vector and that of each song in the database. CBMR then shows a ranked result according to the computed similarity scores.

2. Blockdiagram:

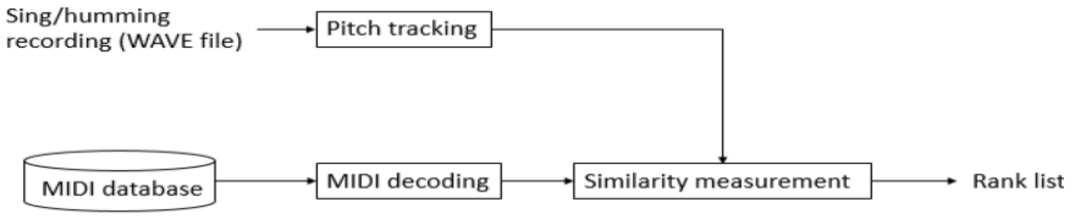


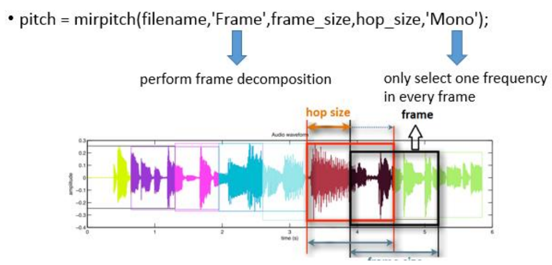
Fig. 1

1. Analyze the pitch of the user recording frame by frame. This operation is called **pitch tracking** in the literature.
2. Since MIDI files comprise only instructions for sound synthesizer, the system decodes the MIDI files to obtain the pitch for each frame.
3. It measures the similarity between the query input and songs in the data base according to the pitch lines obtained in the previous step
4. Select the songs that match the query from the database.

3. Algorithm:

a. Pitch tracking:

I use mirpitch.m to read the testing data (.wav files) into "pitch" data and then uses mirgetdata.m to get the "pitch_data", that is the value of pitch of each frame I sample the input file. And following the paper, I set the frame size to be 0.5 and the hop size to be 0.125 in order to get 0.5*0.125 = 1/16 second for each pitch I sample.



Because the pitch data is saved as semitone in MIDI file, I convert the pitch data I get form the input to semitone by the below equation.

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

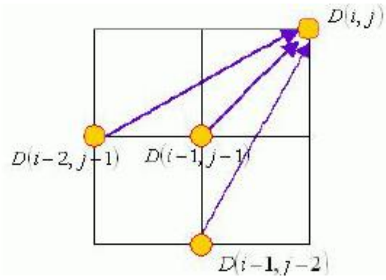
Besides, to eliminate the undesired pitches, such as unvoiced segments and random noise, if the pitch semitones are higher than 84 or NaN (which stands for unvoiced segments), they are all set to 0. Then I replace the 0 part in my input testing data as its previous semitone to fill the unvoiced part and increase the correct rate of the following DTW programming.

b. Dynamic Time Warping(DTW):

For elastic match, we can simply use dynamic time warping to compute the distance between the input pitch vector and that of each songs in the database
Optimal value function: (Recurrence relation)

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

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



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

Boundary Condition:

- $D(i,1) = \infty, i = 2, .., m$

- $D(1,j) = \left| t(1) - r(j) \right|, j = 1, \ldots, n$

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
The cost of the optimal DTW path is defined as

$$\min_j D(m, j)$$

c. 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

1. Set initial parameters and make t and r zero mean:

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

2. Compute the DTW distances:

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

3. Find the minimum DTW distance and update $center$:

If $s_{-1} = \min\{s_{-1}, s_0, s_1\}$, then

center = center – span

else if $s_1 = \min\{s_{-1}, s_0, s_1\}$, then

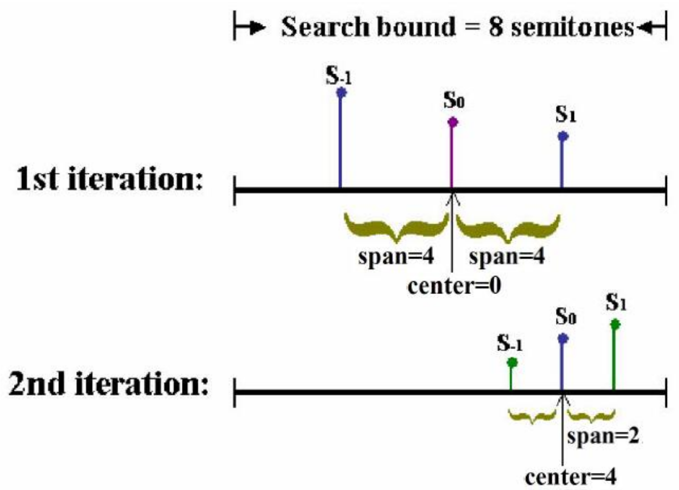
center = center + span

4. Update $span$ and check stopping condition:

If $span > 2$, $span = span / 2$, go to step 3.

Otherwise stop the iteration.

We can express the above steps in a figure below:



4. Experiment Result

Test of 1-DTW:

I ×

I <30x30 double>

	1	2	3	4	5	6	7	8	9	10
1	27	2	22	11	4	6	1	21	14	25
2	2	30	9	14	4	5	18	12	3	16
3	3	14	11	25	28	22	9	27	19	24
4	4	6	30	25	5	28	22	12	3	14
5	4	5	24	23	1	21	14	19	25	22
6	25	7	8	10	29	28	2	4	3	30
7	7	8	5	29	11	16	23	27	2	28
8	8	25	29	7	4	10	3	30	2	28
9	22	30	3	28	23	9	21	14	25	17
10	27	10	15	11	19	26	2	9	30	6
11	22	6	14	11	21	3	23	25	9	15
12	12	26	25	3	5	24	8	22	23	6
13	13	29	8	16	28	11	27	30	7	21
14	14	28	27	24	4	22	19	6	23	12
15	15	4	14	5	7	2	30	22	6	17
16	16	19	5	7	25	4	28	10	23	27
17	14	17	21	19	3	23	6	30	22	5
18	22	21	4	14	11	6	12	3	5	23
19	3	25	23	7	10	8	5	28	14	16
20	22	14	3	4	23	20	2	9	30	25

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the rank list (From lowest DTW cost to highest DTW cost)

M ×

M <30x30 double>

	1	2	3	4	5	6	7	8	9	10
1	51.4883	66.1854	70.1143	74.9961	83.6583	85.4351	85.4557	86.9999	87.6459	97.0983
2	32.3646	210.7821	219.1080	220.5100	233.6563	241.5198	241.6105	246.8829	247.2705	251.7328
3	56.0405	63.3400	66.1735	73.0120	74.1559	78.0470	78.4241	79.5659	81.2962	81.6696
4	55.5776	68.1334	69.9988	74.4883	75.5702	77.9137	77.9510	81.3684	85.2342	86.4304
5	52.8015	65.0276	72.2315	86.8825	90.9872	92.7823	94.0789	94.4748	96.3812	99.8445
6	83.8661	88.7270	91.3782	96.5705	97.3148	110.6204	117.4067	130.3528	133.8937	139.9289
7	42.4341	66.6445	77.1794	83.4016	89.4613	91.4436	92.9264	96.6946	96.9471	101.6099
8	44.2917	64.4628	84.7553	86.6777	91.4916	92.7544	94.1077	98.2103	99.0074	105.9464
9	89.0907	98.5628	103.9944	118.4960	118.6835	123.6732	125.5226	132.1047	135.5163	136.0221
10	597.6965	615.4603	617.8171	627.3976	633.0241	657.6478	664.2979	665.1049	678.8581	700.7647
11	49.5891	55.1723	59.8679	62.0710	65.6058	66.3358	72.5252	73.0623	74.5988	74.6213
12	113.6980	157.9346	178.8066	199.2737	202.6445	203.7920	205.7903	215.3773	222.0612	227.0711
13	119.7861	224.0808	244.3418	271.5247	285.0828	297.3432	310.0347	324.2513	338.1442	343.9114
14	24.4881	70.5225	75.7448	86.4945	99.5979	101.9538	102.2930	103.3515	106.2267	107.3224
15	146.3654	177.0979	200.4588	231.4719	235.9772	244.7190	253.6495	255.3890	255.4957	255.5458
16	52.7143	95.1644	100.1703	116.1041	116.7382	122.1151	122.6798	129.8383	132.1611	138.1640
17	101.1183	113.3946	116.9056	131.1956	133.0447	141.6113	141.6863	149.6208	172.6551	181.6280
18	25.7174	33.7522	56.7260	58.3917	63.4959	65.1207	65.1990	65.4630	68.6971	74.1053
19	103.4638	109.7759	110.6190	113.6160	121.0096	122.7837	135.4361	145.5025	151.3249	157.8904
20	185.5500	195.1575	197.0755	211.6812	214.3269	221.0965	224.7887	229.0371	240.1395	242.5967

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the DTW cost list (From lowest DTW cost to highest DTW cost)

I ×

I <30x30 double>

	1	2	3	4	5	6	7	8	9	10
20	22	14	3	4	23	20	2	9	30	25
21	30	5	2	21	3	14	24	28	1	29
22	22	21	27	4	3	14	12	5	2	23
23	23	14	3	24	11	12	27	18	15	22
24	1	8	23	26	3	24	12	25	27	14
25	25	30	4	5	28	9	6	24	3	14
26	26	16	23	5	25	1	2	30	17	24
27	27	14	11	28	3	22	5	19	25	4
28	28	3	30	19	8	17	23	16	25	1
29	28	23	25	3	24	29	14	7	30	1
30	30	8	3	11	23	2	7	14	6	4

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the rank list (From lowest DTW cost to highest DTW cost)

M ×

M <30x30 double>

	1	2	3	4	5	6	7	8	9	10
21	237.9047	240.4330	242.3442	243.3602	252.8987	262.6297	271.0585	274.7200	283.8614	288.7030
22	14.7537	49.2276	56.2966	61.8711	67.0739	69.3362	74.7063	81.0836	89.1785	89.2782
23	109.8382	136.7970	152.4878	161.6739	166.9939	173.2878	185.0139	186.7544	192.6601	195.7574
24	145.6459	182.1878	215.7039	222.1589	223.9146	230.5099	231.5348	238.2351	247.7012	253.9374
25	65.5242	83.5131	95.1899	95.4714	99.0425	100.7953	108.6792	108.7722	108.7940	111.0328
26	93.4900	106.3464	126.5285	135.2403	162.1135	162.4668	163.3494	172.4905	178.2097	182.9777
27	31.2321	56.9588	76.5660	83.9123	87.6237	88.1599	88.6485	90.1515	99.0489	99.4631
28	132.8959	159.6361	185.6967	195.8198	198.0865	206.2160	213.4884	216.5472	217.5028	217.9890
29	111.0861	150.3754	156.6832	160.1082	161.1982	163.6186	170.8743	173.9708	176.0880	176.7498
30	40.1573	86.4997	90.4446	93.0453	94.6827	95.0357	98.6600	98.9400	101.4709	102.4622

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the DTW cost list (From lowest DTW cost to highest DTW cost)

Comment:

From the above four pictures, we know that top-1 recognition rate is 17/30 = 56.7%, top-3 recognition rate is 20/30 = 66.7% and top-10 recognition rate = 27/30 = 90%. I think the result isn't very good. So, I try to implement key transposition to make it improve better.

Test of 5-DTW:

With the use of key transposition, the computation of each similarity score requires the use of 5 DTW

I <30x30 double>

	1	2	3	4	5	6	7	8	9	10	11
1		2	11	26	22	16	12	18	4	6	1
2	2	30	9	14	4	5	18	12	3	16	19
3	3	14	11	25	28	22	9	27	19	24	7
4	4	18	6	17	30	25	5	28	22	12	3
5	4	5	24	16	23	29	17	1	21	14	19
6	25	7	8	10	6	29	15	28	2	26	4
7	7	8	5	29	15	11	19	16	23	27	2
8	8	25	29	7	6	4	10	3	30	2	15
9	22	30	3	28	23	18	9	21	12	20	14
10	10	25	27	8	16	7	15	30	9	2	23
11	22	18	6	12	14	16	2	11	21	26	3
12	12	26	18	25	29	3	5	24	8	22	23
13	13	29	19	8	16	28	11	27	30	7	21
14	14	28	27	24	26	4	22	17	19	6	23
15	15	4	14	13	5	7	2	30	22	6	17
16	16	15	19	5	1	7	25	10	4	28	23
17	14	13	17	21	19	3	23	6	30	22	5
18	22	21	16	18	4	29	14	26	13	11	6
19	3	25	23	1	7	10	8	5	28	19	14
20	22	14	3	4	23	20	2	9	18	30	25

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the rank list (From lowest DTW cost to highest DTW cost)

I × M ×

M <30x30 double>

	1	2	3	4	5	6	7	8	9	10	11
1	51.4883	66.1854	67.9121	68.4198	70.1143	79.7935	81.7726	83.2200	83.6583	85.4351	85.4557
2	32.3646	210.7821	219.1080	220.5100	233.6563	241.5198	241.6105	246.8829	247.2705	251.7328	252.0707
3	56.0405	63.3400	66.1735	73.0120	74.1559	78.0470	78.4241	79.5659	81.2962	81.6696	82.7530
4	55.5776	62.1348	68.1334	68.5322	69.9988	74.4883	75.5702	77.9137	77.9510	81.3684	85.2342
5	52.8015	65.0276	72.2315	83.6010	86.8825	90.7251	90.8069	90.9872	92.7823	94.0789	94.4748
6	83.8661	88.7270	91.3782	96.5705	97.1253	97.3148	99.0077	110.6204	117.4067	127.8736	130.3528
7	42.4341	66.6445	77.1794	83.4016	86.4586	87.4444	90.5280	91.4436	92.9264	96.6946	96.9471
8	44.2917	64.4628	84.7553	86.6777	88.5333	91.4916	92.7544	94.1077	98.2103	99.0074	104.2493
9	89.0907	98.5628	103.9944	118.4960	118.6835	119.2627	123.6732	125.5226	128.6801	131.2215	132.1047
10	505.8634	586.9748	597.6965	610.1249	612.4442	615.5030	617.8171	617.8276	624.2051	625.3552	625.3841
11	49.5891	51.6890	55.1723	58.9402	59.8679	60.0884	61.5831	62.0710	65.6058	65.6168	66.3358
12	113.6980	157.9346	173.4778	178.8066	198.6003	199.2737	202.6445	203.7920	205.7903	215.3773	222.0612
13	119.7861	224.0808	238.2967	244.3418	271.5247	285.0828	297.3432	310.0347	324.2513	338.1442	343.9114
14	24.4881	70.5225	75.7448	86.4945	90.3268	99.5979	101.9538	102.2118	102.2930	103.3515	106.2267
15	146.3654	177.0979	200.4588	229.0760	231.4719	235.9772	244.7190	253.6495	255.3890	255.4957	255.5458
16	52.7143	82.5783	95.1644	100.1703	113.8896	116.1041	116.7382	119.4522	122.1151	122.6798	132.1611
17	101.1183	113.2588	113.3946	116.9056	131.1956	133.0447	141.6113	141.6863	149.6208	172.6551	181.6280
18	25.7174	33.7522	37.3013	39.2112	56.7260	57.2315	58.3917	59.5825	61.6959	63.4959	65.1207
19	103.4638	109.7759	110.6190	112.4205	113.6160	121.0096	122.7837	135.4361	145.5025	148.1199	151.3249
20	185.5500	195.1575	197.0755	211.6812	214.3269	221.0965	224.7887	229.0371	232.3536	240.1395	242.5967

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the DTW cost list (From lowest DTW cost to highest DTW cost)

M ×

M <30x30 double>

	1	2	3	4	5	6	7	8	9	10	
21	21	30	5	2	3	14	24	28	1	13	29
22	22	21	18	27	4	16	29	26	3	14	12
23	23	14	3	24	11	18	12	27	15	22	5
24	1	21	8	13	29	23	15	26	3	24	12
25	25	30	20	4	5	28	9	17	6	24	3
26	26	16	23	5	21	13	25	1	2	19	30
27	27	14	11	26	28	3	22	5	19	25	4
28	28	3	30	19	8	17	23	13	16	25	1
29	28	23	13	25	3	24	29	21	14	7	30
30	30	8	3	19	11	23	2	17	7	14	6

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the rank list (From lowest DTW cost to highest DTW cost)

I × M ×

M <30x30 double>

	1	2	3	4	5	6	7	8	9	10	11
21	205.2303	237.9047	240.4330	242.3442	252.8987	262.6297	271.0585	274.7200	283.8614	283.9103	288.7030
22	14.7537	49.2276	49.3650	56.2966	61.8711	62.0269	65.9193	66.7702	67.0739	69.3362	74.7063
23	109.8382	136.7970	152.4878	161.6739	166.9939	172.4329	173.2878	185.0139	192.6601	195.7574	196.4905
24	145.6459	180.5827	182.1878	194.0076	205.1006	215.7039	217.8305	222.1589	223.9146	230.5099	231.5348
25	65.5242	83.5131	93.9720	95.1899	95.4714	99.0425	100.7953	103.9284	108.6792	108.7722	108.7940
26	93.4900	106.3464	126.5285	135.2403	135.4088	159.5839	162.1135	162.4668	163.3494	169.5124	172.4905
27	31.2321	56.9588	76.5660	83.1096	83.9123	87.6237	88.1599	88.6485	90.1515	99.0489	99.4631
28	132.8959	159.6361	185.6967	195.8198	198.0865	206.2160	213.4884	216.1345	216.5472	217.5028	217.9890
29	111.0861	150.3754	156.5223	156.6832	160.1082	161.1982	163.6186	167.2892	170.8743	173.9708	176.0880
30	40.1573	86.4997	90.4446	91.5327	93.0453	94.6827	95.0357	98.5958	98.6600	98.9400	101.4709

The leftmost column 1~30 represent the ground truth 1~30 and the row 1~30 represent the DTW cost list (From lowest DTW cost to highest DTW cost)

Comment:

From the above four pictures, we know that top-1 recognition rate is 19/30 = 63.3%, top-3 recognition rate is 21/30 = 70% and top-10 recognition rate = 30/30 = 100%. This result has been improved better but still not as good as the paper.

5. Conclusion & Future works:

1. I make a QBHS system to query songs in a training data set. Although the final result I get isn't very great compared with the paper, my code also provides a not bad QBHS system for people to use.
2. In the future, I want to improve my recognition rate better. I have some thoughts. One, Note segmentation for quick performance evaluation based on note-level comparison. Second, better smoothing techniques to enable the use of difference operator. Third, speeding the code up.

6. Reference

1. http://mirlab.org/jang/books/audiosignalprocessing/qbshDtw1&2.asp?title=14-4%20DTW%20of%20Type-1%20and%202
2. http://mirlab.org/jang/books/audiosignalprocessing/
3. http://whale.cse.nsysu.edu.tw/~cyyang/A%20Query-by-Singing%20System%20based%20on%20Dynamic%20Programming.pdf
4. http://nthur.lib.nthu.edu.tw/handle/987654321/67135