

# DP매칭에 의한 On-Line 악보인식

## (On-Line music score recognition by DPmatching)

구상훈\* 이병선\* 김수경\* 이은주\*

**요약** 컴퓨터의 기술적 발전은 사회 여러 분야에 막대한 영향을 끼쳤다. 그중 악보 인식분야에도 커다란 영향을 주었다. 그러나, On-line 상에서 그런 악보를 실시간으로 정형화된 악보형태로 변환하는 처리에 대한 연구가 미흡하여 이에 대한 연구가 필요하다. 본 논문에서는 실시간으로 악보를 인식하고, 사용자의 편의를 도모하기 위해 DP(Dynamic Programming) 매칭법을 이용한 On-Line 악보인식에 관한 방법을 제안하였다.

본 연구에서는 실시간으로 입력되는 악상기호를 인식하기 위해, 가장 유효한 정보인 악상 기호내의 방향, x, y 좌표를 이용하여 벡터형태로 추출한 후 음표와 비음표(섬표, 기타기호)의 두개의 그룹으로 나누어진 표준패턴과의 DP매칭을 통해 인식한다.

먼저 tablet을 통해 실시간으로 악상 기호를 입력할 때 생기는 x, y좌표를 이용하여, 펜의 움직임에 대한 16방향 부호화를 수행한다. 음표와 비음표를 구분하기 위한 시간을 줄이고자 16방향 부호화를 적용하지 않고 사사분면부호화를 적용한다. 음표를 약식으로 그릴 경우 음표 머리에 해당하는 부분의 좌표는 삼사분면에 분포하고, 폐곡선의 음표일 경우에는 좌표가 사사분면에 고르게 나타난다. 폐곡선을 제외한 음표의 머리는 폐곡선과 같은 조건이면서 입력 받은 y좌표값들 중에서 최소값과 최대값을 구한 다음 2로 나눈 값을 지나는 y좌표의 개수가 임의의 임계값 이상이면 음표로 판단한다. 위 조건을 만족하지 않을 경우 비음표로 취급한다. 음표와 비음표를 결정한 다음, 입력패턴과 표준패턴과의 DP매칭을 통하여 별점을 구한다. 그리고 경로탐색을 통해 별점에 대한 각각의 합계를 구해 최소값을 악상기호로 인식하였다. 실험결과, 표준패턴을 음표와 비음표의 두개의 그룹으로 나누어 인식함으로써 DP 매칭의 처리 속도를 개선시켰고, 국소적인 변형이 있는 패턴과 특징의 수가 다른 패턴의 경우에도 좋은 인식률을 얻었다.

## 1. 서론

최근 컴퓨터의 기능의 확장 및 통합에 의해 멀티미디어 마인드가 확산되고 있는 추세

이며 이를 뒷받침하는 특수 장치의 개발 및 소프트웨어 개발도 큰 진전을 보이고 있다. 정보화 사회의 발전과 함께 컴퓨터를 인간의 기능에 대신하도록 하기 위해서 이를 여러 각도로 발전시켜 왔으며, 이로인해 컴퓨터는 단순한 계산 기능 이외에 추론 및 결정 기능과 음성인식 및 출력기능, 그리고 시각기능

\* 한밭대 컴퓨터 공학과

즉 문서 정보의 인식기능을 갖게 되었다. 문서 입력의 자동화는 문자인식 분야에서 많은 연구가 이루어져 왔으며, 현재에도 여러 방향에서 연구 중에 있다. 음악 분야에서도 음악양식의 분석, 비교, 편곡, 그리고 작곡에 이르기까지 광범위한 분야에서 컴퓨터를 이용한 연구가 활발하게 수행되고 있다[1][2]. 인쇄체 문자 및 필기체 문자에 대한 연구는 여러나라에서 각국의 문자 인식에 대한 관심으로 연구가 활발하며 기타 여러 가지 도형 및 사물에 대한 연구도 많이 이루어졌으나 인쇄악보에 대한 연구는 아직 미흡한 상태이다.

인쇄악보의 인식은 악상 기호의 특징을 고려하여 주로 패턴 매칭과 히스토그램(histogram)을 이용하여 이루어진다.

최근에는 이미 인쇄된 악보에 대한 인식 및 시스템 개발이 활발하게 이루어지고 있다. 오프라인 인쇄악보의 인식은 컴퓨터로 인쇄된 악보영상을 입력받아 히스토그램을 이용하여 각 악상기호들을 분리하고, 각 방향에서의 특징 벡터를 구한 결과를 이미 존재하는 표준 패턴과 비교하여 패턴을 분류, 인식하는 방식을 취한다. 히스토그램 방식은 영상 또는 영상의 일부에 대한 화소 정도 값의 분포 그래프를 이용하는 방법으로 오선의 위치, 세로선, 심볼들을 인식할 때 히스토그램을 수직, 수평 또는 여러 방향에서 투영함으로써 특징을 추출하여 인식 한다.

이 방법은 기울어진 악보에 대한 처리가 불가능하다. 또한 기울기 정보를 구할 수 있다면 기울어진 방향으로 투영한다는 방법도 고려해 볼 수 있겠지만, 전 심볼에 대해서 기울어진 방향으로 투영해야 한다고 가정했을 때 검사해야 할 화소를 모든 계산해내야 한다는 것은 매우 비효율적이다[3].

기존의 악보 입력 방법으로는 알파벳이나 숫자 형태를 가진 코드로의 입력, 키보드 또는 마우스를 이용한 아이콘 선택에 의한 입력 등이 있다. 그러나 이와 같은 수동적인 악보 입력 방식은 입력 속도가 느릴 뿐 만

아니라 입력 시 오류 발생의 가능성이 크다 [4].

따라서 본 논문에서는 실시간으로 악보를 인식하고 사용자의 편의를 도모하기 위해 DP 매칭법을 이용한 On-Line 악보인식에 관한 방법을 제안하였다. 먼저 tablet을 통해 실시간으로 악상 기호를 입력할 때 생기는 x, y좌표를 이용하여, 펜의 움직임에 대한 16방향 부호화를 수행한다. 음표를 약식으로 그릴 경우 머리에 해당하는 부분의 좌표는 삼사분면에 분포하고, 폐곡선일 경우에는 좌표가 사사분면에 고르게 나타난다. 폐곡선을 제외한 음표의 머리는 폐곡선 음표와 같은 조건이면서 입력 받은 y좌표값들 중에서 최소값과 최대값을 구한 다음 2로 나눈 값을 지나는 y좌표의 개수가 임의의 임계값 이상이면 음표로 판단한다. 그리고 위 조건에 만족 하지 않는 것은 쉼표와 기타 기호로 취급한다. 음표와 비음표를 결정한 다음, 입력 패턴과 표준패턴과의 2차원 테이블을 구성하고 DP매칭을 통하여 별점을 구한다. 삼입, 삭제, 정합 이동을 통해 입력패턴과 표준패턴사이의 별점에 대한 각각의 합계를 구하여 최소가 되는 것을 악상기호로 인식하도록 하는 방법을 제안 하였다. 본 논문의 구성은 2장에서는 음표 머리 검사에 대해, 3장에서는 입력정합 벡터와 표준 정합 벡터 구성에 대해 기술 하였고, 4장은 정합 행렬에 대해, 5장에서는 DP매칭에 의한 경로 찾기, 6장 실험 및 고찰, 7장은 결론에 대해서 기술 하였다.

## 2. 음표 머리 검사

### 2.1 폐곡선

폐곡선음표 경우에는 [그림-1]과 같이 사사분면 전체에 걸쳐 고르게 나타난다. 따라서 입력된 좌표들은 [조건1]을 통해 각 분면을 만족하는 좌표에 대해 카운트를 한 후,

[조건1]

- ① 1사분면 :  $B_X < A_X$  And  $B_Y > A_Y$  Increase(one)
- ② 2사분면 :  $B_X > A_X$  And  $B_Y > A_Y$  Increase(two)
- ③ 3사분면 :  $B_X > A_X$  And  $B_Y < A_Y$  Increase(three)
- ④ 4사분면 :  $B_X < A_X$  And  $B_Y < A_Y$  Increase(four)
- ④ 2분음표 머리 인식 조건

$$\left(\frac{one}{tot} * 100\right) > 10\%, \left(\frac{two}{tot} * 100\right) > 5\%, \left(\frac{three}{tot} * 100\right) > 10, \left(\frac{four}{tot} * 100\right) > 5\%$$

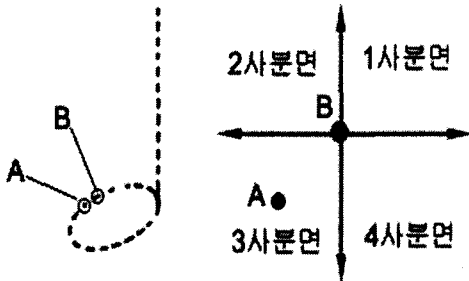
면 음표로 판단한다.

[조건2]

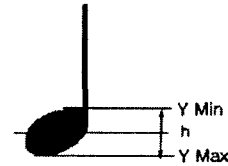
if  $A'_X < A_X$  and  $A'_Y > A_Y$  then Count++ else tot++

$$E = \frac{Count}{tot} * 100 > 60\%, \quad E: \text{전체 좌표의 개수 대한 3사분면상의 백분율}$$

각 분면을 전체 좌표개수로 나누면 각 분면의 좌표 비율이 고르게 난다.



[그림-1] 3사분면의 예



[그림-2] 4분음표의 예

$$h = \frac{YMin + YMax}{2}, \quad h-1 < h' < h+1$$

(식1)

$h$ : 기준선,  $h'$ : 기준선  $h$ 를 지나는  $y$ 좌표

## 2.2 폐곡선을 제외한 음표

[그림-2]과 같이 폐곡선을 제외한 음표의 머리는 (식1)에 의해, 폐곡선 음표와 같은 조건이면서 입력 받은  $y$ 좌표값들 중에서 최소값과 최대값을 구한 다음, 2로 나눈 값을 지나는  $y$ 좌표의 개수가 임의의 임계값 이상이

그러나 [그림-2]과 같이 폐곡선음표를 제외한 음표의 머리를 약식으로 그릴 경우, 음표 또는 비음표에 대한 여부를 [조건2]를 통해 구하게 되는데, 이것은 각각의 입력된 좌표에 대해 3사분면을 향하는 좌표가 전체 좌표개수에 대한 백분율 즉, 60%보다 크면 음표의 머리로 간주 한다.



[그림-3] 3사분면 형태의 머리

### 3. 입력정합 벡터와 표준 정합 벡터 구성

#### 3.1 입력정합 벡터 구성

##### 3.1.1 입력 및 표현

tablet 상에서 pen을 이용하여 악상 기호를 입력하면 tablet에 pen이 그린 악상기호에 대한 x, y의 좌표 즉, tablet상의 위치정보를 얻을 수 있다. 또한 입력의 끝은 현재 악상기호를 입력한 후 일정거리에 다음 악상기호를 입력할 때까지 이고, 이때, pen의 down에서 up까지의 모든 좌표를 포함한다.

여기서 입력된 악상 기호를 M, 악상기호의 i번째 획을S, P<sub>ij</sub>를 획 S<sub>i</sub>에 대한 x, y좌표로 아래와 같이 표현 하였다.

$$M = \{ S_1, S_2, S_3, S_4, \dots, S_i, \dots, S_N \}$$

$$S_i = \{ P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{ini} \}$$

$$P_{ij} = \{ X_{ij}, Y_{ij} \}$$

$$i = 1, 2, 3, \dots, N$$

$$j = 1, 2, 3, \dots, ni$$

N: 입력자료에 대한 획의 총수

ni: S<sub>i</sub> 포함된 좌표의 총수

X<sub>ij</sub>: P<sub>ij</sub>의 X좌표, Y<sub>ij</sub>: P<sub>ij</sub>의 Y좌표

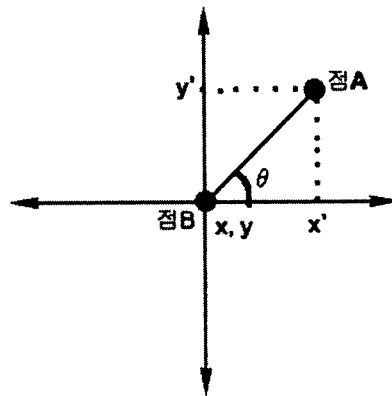
##### 3.1.2 부호화

tablet으로 입력된 인식의 주요한 정보를 제공하게 되는 악상 기호에 대한 방향성을 추출하고자 다음과 같은 단계를 거쳐게 된다.

#### (1) 각도 및 방향 구하기

tablet으로 입력된 모든 좌표 점에 대한 방향을 구하기 위해 [그림-4]의 좌표 점A와 B를 (식2)를 통해 두 점 사이의 x와 y좌표의 차를 이용하여 각도를 구한다. 그리고 [그림-4]와 같이 16방향에 대응 시키기 위해 구한 각도를 [그림-5]의 16방향 선분에 대응시킨다.

즉, 각각의 방향은 22.5° 의 간격으로 이루어져 있으며 16방향에 대응하는 각도와 (식2)에서 구한 각도와의 차를 구해 최소가 되는 각도를 대응하는 방향으로 결정한다.



[그림-4] 각도 구하기

$$Xlength = |X - X'|, \quad Ylength = |Y - Y'|$$

$$\theta = \frac{Ylength}{Xlength} * (180 / 3.14)$$

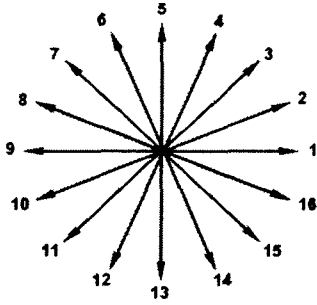
$$g(1) = 0, g(2) = 22.5, g(3) = 45, \dots, g(16) = 337.5$$

$$g = g(1), \dots, g(i) \quad i = 1, \dots, 16,$$

$$D = (Min g - \theta, i)$$

(식2)

D: 추출된 방향, θ: 각도, g: 각도를 16방향에 대응



[그림-5] 16방향

(2) 특징점 추출

[그림-6]은 악상 기호를 입력한 예로, 입력된 모든 좌표들이 (식2)를 통하여 [표-1]과 같이 16 방향 중 한 방향으로 구해지게 된다. 여기서 각각의 모든 점에 대한 방향성을 추출하게 되면 동일한 방향이 중복되어 계산되고, 또한 불필요한 좌표에 대한 고려도 해야 된다[6].



[그림-6] 악상 기호입력 예

따라서 [표-1]의 방향중에서 동일한 방향을 합친 다음 동일한 방향에 대한 카운트를 한다. 카운트가 1이 되는 것은 입력벡터를 구성하는데 있어서 큰 영향을 끼치지 않기 때문에 카운트가 1이되는 것을 제외한 나머지 방향에 대해서 입력된 악상 기호 벡터를 생성한다. [표-2]는 [표-1]의 방향을 축소한 결과, [그림-6]에 대한 [그림-7]은 최종 입력 벡터를 나타낸다.

입력 벡터 : {11, 12, 11, 11, 11}

[그림-7] 최종 입력 벡터

[표-1] 방향 축소 전

X 좌표	Y 좌표	방향
368	55	9
366	55	11
365	56	11
364	57	11
363	58	11
362	59	12
361	61	12
360	63	10
358	64	11
357	65	11
356	66	11
355	67	12
353	70	11
352	71	11
351	72	12
350	74	11
349	75	11
348	76	12
347	78	11
346	79	13
346	80	8

3.2 표준정합 벡터 구성


입력정합 벡터의 구성은 입력 악상 기호에 대해 3.1.1절을 통해 좌표점을 추출하고, 3.1.2절의 과정을 거쳐 16방향의 부호화를 수행하면 된다. 그리고 표준정합 벡터는 인식을 위해 정자로 악상 기호를 입력하여, 입력


[표-2] 방향 축소 후

X 좌표	Y 좌표	방향	카운트
368	55	9	1
366	55	11	4
362	59	12	2
360	63	10	1
358	64	11	3
355	67	12	1
353	70	11	2
351	72	12	1
350	74	11	2
348	76	12	1
347	78	11	1
346	79	13	1
346	80	8	1

음표	음표	폐곡선		2
	머리	그 외		5
비음표	쉽표			3
	그 외			3

[그림-8] 표준 패턴


  
 {10, 11, 10, 12, 14, 15, 1, 5, 6, 7, 9}  
 {10, 11, 13, 10, 1, 3, 4, 5, 9, 3, 4, 3, 4, 5, 9, 13}  
 {10, 11, 3, 4, 6, 13}  
 {11, 10, 11, 4, 13, 5, 4, 5, 1, 16, 14, 13, 12, 5, 6, 7, 8, 15, 14, 13, 12, 11}  
 {10, 11, 13}  
 {11, 13, 1, 2, 3, 5, 7}  
 {10, 11, 3, 4, 5, 4, 5, 13, 5, 1, 15, 14, 13, 11}


  
 {13, 5, 1, 13, 9, 10, 9, 10, 1, 3, 1}  
 {10, 12, 11, 10, 10, 11, 13, 10, 1, 3, 1}  
 {11, 13, 9, 14, 1, 3, 1, 3, 6, 11, 14, 1, 5, 7, 11, 1, 5, 11, 13, 10, 1, 3, 12}  
 {12, 8, 4, 1, 15, 11, 8, 5, 2, 5, 9, 12, 9, 5, 15, 7, 4}  
 {13, 5, 1, 10, 13, 10, 2}  
 {1, 10, 1, 5, 8, 5, 13, 13, 4, 13, 13, 4}

[그림-9] 표준정합 벡터

$$P_{ij} = [ \text{Min} ( \text{InputM}(D) - \text{StandM}(D), (16 - \text{InputM}(D) - \text{StandM}(D)) ) ]^3 + 1 \quad (\text{식 } 3)$$

*InputM(D)* : 입력정합 벡터

*StandM(D)* : 표준정합 벡터

*D* : 각 좌표점 사이에서 구해진 16방향 코드

S	...	...	Stand	Stand	Stand	Stand	...
I	...	...	M(D)	M(D)	M(D)	M(D)	...
...			P <sub>ij</sub>	P <sub>ij</sub>	P <sub>ij</sub>	...	...
InputM	...	...	P <sub>ij</sub>	P <sub>ij</sub>	P <sub>ij</sub>	...	...
(D)	...	...	P <sub>ij</sub>	P <sub>ij</sub>	P <sub>ij</sub>	...	...
InputM	...	...	P <sub>ij</sub>	P <sub>ij</sub>	P <sub>ij</sub>	...	...
(D)	...	...	P <sub>ij</sub>	P <sub>ij</sub>	P <sub>ij</sub>	...	...
...	...	...	P <sub>ij</sub>	P <sub>ij</sub>	...	...	...

[그림-10] 정합행렬

정합 벡터와 같은 단계를 거친 후 생성을 한다. [그림-8]은 표준패턴을 나타낸 것으로, 음표와 비음표로 크게 두 가지로 나누고, 세부적으로는 음표의 머리에 대해 폐곡선인 경우와 아닌 경우, 비음표는 쉽표와 그 외 기호로 구분을 하였다.

[그림-8]에서처럼 표준패턴들을 구분함으로

써, 입력 패턴에 대응 하는 표준패턴 전체에 대한 계산을 할 필요가 없으므로 계산 시간이 단축되며, 인식의 범위를 줄이는 효과로 인해 인식률을 높일 수 있다. 정합행렬에 쓰일 표준패턴에 대한 표준정합 벡터는 [그림-9]에 나타나 있다.

## 4. 정합 행렬

정합행렬은 입력정합 벡터를 행의 기준정보로, 표준정합 벡터를 열의 기준 정보로 하여 2차원 배열로 구성되고 각각의 교차점에서 입력 악상기호와 표준 악상기호사이의 별점을 (식3)을 통해 구하게 된다. [그림-10]은 정합행렬이고, 행렬의 요소인 "Pij"는 입력정합 벡터 InputM(D)과 표준정합 벡터 StandM(D)사이의 별점이다[5].

## 5. DP매칭에 의한 경로 찾기

식별대상 패턴이 국소적인 변형을 포함하는 경우나 특징의 수가 패턴에 따라서 다른 경우 등에 유효하게 이용되는 DP(Dynamic Programming) 매칭법을 이용하여 정합 경로를 찾는다[7]. 정합경로는 정합행렬의 첫 번째 요소부터 이동하여 별점의 합이 최소가 되는 경로를 찾는 것으로 현재의 위치인 MATRIX[i, j]에서 가능한 이동은 [정의1]과 [정의2]와 같이 정의 한다. 여기서 삽입이동은 표준패턴 하나의 획에 다수개의 입력 획이 대응된 것이고, 삭제 이동은 다수의 표준패턴의 획들이 하나의 입력 획에 대응된 것이다. 정합이동은 입력 패턴과 표준패턴의 획이 1대 1로 대응된 경우이다.

[정의1] 정합행렬에서의 이동

- ① 삽입이동 : MATRIX[i, j] ---> MATRIX[j+1, j]
- ② 삭제이동 : MATRIX[i, j] ---> MATRIX[i, j+1]
- ③ 정합이동 : MATRIX[i, j] ---> MATRIX[i+1, j]

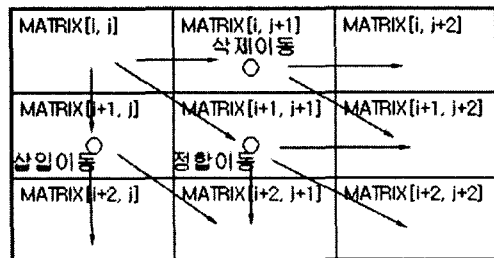
경로 찾기에서 현재의 위치에서 다음 상태로의 이동은 [그림-11]에서와 같이 3 X 3 원도우를 사용한다. 이것은 단순히 최소가 되는 값으로 이동을 하게 되면 국부적인 최소

에 빠지기 쉽고 윈도우를 크게 하면 탐색시간이 많이 걸리기 때문에 사용하였다. 3 X 3 이 윈도우는 [정의2]를 이용하여 이동방향에 따라서 예상되는 별점을 계산하고 최소의 별점을 갖는 경로를 택하여 이동한다.

[정의2] 최적 경로 탐색을 위한 별점 계산

- ① 삽입이동 별점 :  $(Pel(MATRIX[i+1, j]) + \text{Min}(Pel(MATRIX[i+2, j]), Pel(MATRIX[i+2, j+1]))) \times 1.5$
- ② 삭제이동 별점 :  $Pel(MATRIX[i, j+1]) + \text{Min}(Pel(MATRIX[i, j+2]), Pel(MATRIX[i+1, j+2]))$
- ③ 정합이동 별점 :  $(Pel(MATRIX[i+1, j+1]) + \text{Min}(Pel(MATRIX[i+1, j+2]), Pel(MATRIX[i+2, j+2]), Pel(MATRIX[i+2, j+2]))) \times 1.5$

$Pel(MATRIX[i, j])$  : 정합행렬에서 i번째 행과 j번째 열의 별점



[그림-11] 최적 경로 탐색

[그림-12]은 3X3원도우를 이용하여 '#'에 대한 최적 경로 탐색의 한 예를 나타낸 그림으로, 실선은 실획(tablet이 down인 상태)이고, 점선은 가상획(tablet이 up인 상태)을 나타내며, S는 표준 정합 벡터를, I는 입력정합 벡터를 나타낸다[5].

## 6. 실험 및 고찰

본 논문에서 제안한 알고리즘의 실험에 사

S	I	1	10	1	5	8	5	13	13	4	13	13	4	20	20	20	20	20	20	20	20	
1	0	9	0	4	7	4	12	12	3	12	12	3	19	19	19	19	19	19	19	19	19	19
2	1	8	1	3	6	3	11	11	2	11	11	2	18	18	18	18	18	18	18	18	18	18
11	6	1	6	6	3	6	2	2	7	2	2	7	9	9	9	9	9	9	9	9	9	9
10	7	0	7	5	2	5	3	3	6	3	3	6	10	10	10	10	10	10	10	10	10	10
9	8	1	8	4	1	4	4	4	5	4	4	5	11	11	11	11	11	11	11	11	11	11
1	0	9	0	4	7	4	12	12	3	12	12	3	19	19	19	19	19	19	19	19	19	19
16	1	6	1	5	8	5	3	3	4	3	3	4	4	4	4	4	4	4	4	4	4	4
1	0	9	0	4	7	4	12	12	3	12	12	3	19	19	19	19	19	19	19	19	19	19
6	5	4	5	1	2	1	7	7	2	7	7	2	14	14	14	14	14	14	14	14	14	14
8	7	2	7	3	0	3	5	5	4	5	5	4	12	12	12	12	12	12	12	12	12	12
13	4	3	4	8	5	8	0	0	7	0	0	7	7	7	7	7	7	7	7	7	7	7
5	4	5	4	0	3	0	8	8	1	8	8	1	15	15	15	15	15	15	15	15	15	15
4	3	6	3	1	4	1	9	9	0	9	9	0	16	16	16	16	16	16	16	16	16	16
5	4	5	4	0	3	0	8	8	1	8	8	1	15	15	15	15	15	15	15	15	15	15
13	4	3	4	8	5	8	0	0	7	0	0	7	7	7	7	7	7	7	7	7	7	7
13	4	3	4	8	5	8	0	0	7	0	0	7	7	7	7	7	7	7	7	7	7	7
14	3	4	3	7	6	7	1	1	6	1	1	6	6	6	6	6	6	6	6	6	6	6
5	4	5	4	0	3	0	8	8	1	8	8	1	15	15	15	15	15	15	15	15	15	15
4	3	6	3	1	4	1	9	9	0	9	9	0	16	16	16	16	16	16	16	16	16	16

[그림-12] '#'의 정합경로

용한 시스템은 CPU 1.7GHz, RAM 512M 이고, 악상기호 입력은 와콤 ArtPad II tablet 을 사용 하였으며, Visual Basic6.0 으로 알고리즘을 구현하였다. 표준 패턴은 총 13개 로 하였고 각각의 입력 정합 벡터에 대한 악상 기호의 입력은 tablet에서 자연스럽게 필기 하듯 각 악상기호 당 20개씩 tablet을 통해 입력하였다. [그림-13]은 인식의 예를 보이고 있는데, 그림에서 보는 바와 같이 최적 경로 탐색을 통해 구해진 별점의 합과 음표를 1~7번 까지 오름차순으로 나열하고 있다. 최종적으로 별점의 합이 최소가 되는 1번 즉, 8분음표로 인식을 하게 된다.

 입력 기호 특징점 추출	1: 16. 8분음표 (♩)
	2: 30. 16분음표 (♫)
	3: 175. 2분음표 (♩)
	4: 993. 온음표 (♩)
	5: 1555. 8분음표 (♩)
	6: 2071. 4분음표 (♩)
	7: 6582. 4분음표 (♩)
<인식 결과>	

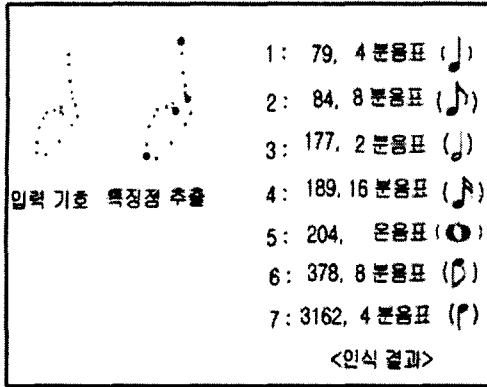
[그림-13] 인식 예

[그림-14]는 표준 패턴과 입력 패턴과의 매칭시 DP매칭의 장점인 입력패턴의 변화가 있는 경우에도 유효하게 인식 되는 것을 볼 수가 있다.


[그림-14] 인식 이 잘된 예

[그림-15]는 오인식의 예를 나타낸 것으로, tablet에서 악상기호를 입력할 때 빠르게 입력 하면 좌표가 제대로 입력이 되지 않게 된다. 따라서 특징점 추출이 천천히 입력 했을 때 보다 인식에 유효한 특징점이 적어지게 되므로 의도하지 않은 악상기호로 오인식 되었다.





[그림-15] 오인식의 예

제안한 방법의 유용성을 평가하기 위한 실험 결과, [표-3]와 같이 음표인 경우에는 75.7%의 인식률을 얻었고, 비음표인 경우에는 82.5%의 인식률을 얻었는데 음표의 경우는 악상기호들의 유사성 때문에 비음표보다 조금 낮은 인식률을 얻었다. 인식 시간은 한 악상 기호 당 1초의 인식 속도를 보였다. [표-3]은 각 악상기호 당 20개씩 입력받아 실험한 인식 결과표이다.

[표-3] 인식 결과

	음표	비 음표
총 악상기호 입력 수	140	120
인식 수	106	99
오 인식 수	34	21
인식률	75.7%	82.5%

## 7. 결론

본 논문에서는 실시간으로 악보를 인식하고, 사용자의 편의를 도모하기 위해 온라인 상에서 입력된 악상기호를 DP매칭 기법을 이용하여 인식하는 방법을 제안하였다. DP매칭에서 정합행렬의 최적 경로 탐색을 통해

입력정합 벡터와 표준정합 벡터 사이의 별점에 대한 최소 합을 구하기 위해 먼저 tablet을 통해 실시간으로 악상 기호를 그릴 때 생기는 x, y좌표를 이용하여 16방향으로 부호화를 하였다. 이때 tablet펜이 down상태 뿐만 아니라 up상태 일 때, 입력되는 좌표에 대한 부호화를 함으로써, 악상 기호를 특징을 지을 수 있는 방향을 구할 수 있었다. 그리고 부호화를 할 때 모든 좌표에 대한 방향성을 추출하게 되면 동일한 방향이 중복되어 계산된다. 따라서 계산량을 줄이기 위해, 이전에 구한 방향과 현재의 방향을 비교하여 같은 방향이면 합침으로써, 추출되는 방향의 개수를 줄일 수 있었다. 정합행렬에서는 3X3 윈도우를 이용하여 삽입, 삭제, 정합 이동을 통해 입력 정합 벡터와 표준 정합 벡터 사이의 별점으로 다음의 이동 경로를 결정 하였고, 이동 경로에서의 합계를 구한다음 최소가 되는 값을 인식으로 하였다.

본 논문에서 표준패턴들을 구분함으로써, 입력 패턴에 대응 하는 표준패턴 전체에 대한 계산을 할 필요가 없으므로 계산 시간이 단축되며, 인식의 범위를 줄이는 효과로 인해 좋은 인식률을 얻을 수 있었고, 제안한 DP매칭법으로 국소적인 변형이 있는 패턴과 특징의 수가 다른 패턴의 경우에도 좋은 인식률을 얻었다.

향후 연구 과제는 악상 기호 입력 시 너무 흘려 쓴다거나 빠르게 쓸 때의 오인식률을 줄이는 연구와 표준패턴수를 늘리고, 그에 따른 여러 가지 조건을 구축하여 인식률을 높이는 연구가 필요하다.

## 참고 문헌

- [1] 김정규, “음악기호의 계층구조를 이용한 인쇄악보 인식”, 한국화상학회지, Vol. 3, NO. 1, 1997, pp.66-68
- [2] 황영길, 원영주, 김성천, “악보 인식 기술 현황”, 정보과학회지 제11권 제 5

- 호, 1993, pp.35-40
- [3] 조경은, “골격선과 런 길이 정보를 이용한 피아노 악보 인식”, 동국대학교 대학원 컴퓨터 공학과, 1994, pp.2-7
- [4] 이성기, 신재욱, “신경망을 이용한 악보 인식”, 한국정보과학회논문지 제21권 제7호, 1994, pp.1358-1360
- [5] 이은주, 박진열, 박재성, 김태균, “구조 정보의 DP 정합에 의한 흘려 쓴 한글의 온라인 인식”, 전자공학회논문집 제31권 B편 제4호, 1993, pp.516-524
- [6] 이성환, 문자인식 이론과 실제 1권, 서울 : 홍릉과학출판사, 1993, pp116-125
- [7] 安居院 猛, 長菴 智晴, 畫像の處理と認識, 昭晃堂, 1994, pp.134-140