

# 한글 VECTOR 表記를 위한 最小情報 構成에 關한 研究 (A Study on Organizing the Least Informations for Vector-Drawing Korean Characters)

金 三 卯\*  
Kim, Sam Myo

## 要 約

한글을 vector 表記하는데 필요한 最小情報를 構成하기 위하여 한글 字素를 주어진 display raster 上에 그리는 最短經路 및 始作點을 비교적 간단한 algorithm을 이용하여 찾았고, 이 經路를 따라 그리는 데 필요한 vector 制御 情報를 最小로 要하는 기본 vector 群은 raster 거리의 1 및 3 배의 크기를 갖는 vector 들이며, 9 개의 重子母音字를 포함한 33 개의 한글 字素를 그리는 데 필요한 總 vector 制御 情報는 horizontal format로 7 bits 463 words 즉 3,241 bits로 構成될 수 있었다.

## Abstract

To organize the least control information for vector-drawing Korean characters, relatively simple 'Shortest Route Searching Algorithms' which can be applied to Korean character patterns were constructed. The shortest drawing routes and the common starting point for all characters were found with this algorithm.

It is shown that the vectors, which needs the least control informations to follow the shortest drawing route, are those vertical and horizontal vectors which have the size of 1 and 3 times of raster distance. The vector control information for drawing 33 Korean characters including 9 double consonants and vowels can be organized with 7 bits 463 words of horizontal format informations which are corresponding to 3,241 bits.

## 1. 序 論

한글 情報의 入出力을 위한 研究가 많이 進行되었으 며<sup>1)3)5)6)</sup> 印刷機에의 出力의 경우 그 속도와 글자의 모양에 문제점이 있으나 drum, chain 또는 dot matrix 등의 방식에 의하여 처리되고 있다<sup>9)10)11)</sup>. 그러나 出力의 다른 主 裝置의 하나인 display의 경우 아직 한글처리를 위한 실용화된 system이 이용되지 못하고 있다. 이는 한글 display가 현재 印刷機만큼 절실히 요구되지 않은데도 그 원인이 있겠으나 곧 microcompter와 같은 소형 system이 널리 보급되고 이들이 대부분

display를 이용하게 된다는 것을 고려할 때 비교적 저렴하고 효율적인 display system 개발이 필요하다. 물론 既存 system에 한글 字素의 모양에 관한 情報와 이를 處理하는 algorithm에 의하여 한글을 display할 수 있겠으나<sup>9)</sup> 그 pattern 情報 및 algorithm은 되도록 적고 간편하게 구성되어야 할 것이다.

현재 주로 이용되고 있는 display 방법으로는 주어진 dot pattern을 display하는 point plotting display 방법, 두 끝점의 좌표에 의하여 display하는 line generation 방법, 기준점과 이 점을 起點으로한 x 및 y 方向의 增分 vector  $\Delta x$  및  $\Delta y$ 에 의하여 display하는 vector generation 방법 등이 쓰여지고 있다<sup>7)9)</sup>. point plotting 방법은 그 裝置가 비교적 간단하여 저렴하며

\* 正會員 慶北大學校 電子工學科  
接受日字: 1977年 8月 12日

pattern을 위한 情報 또한 비교적 적게 구성될 수 있겠으나 모아쓰기 된 한글을 쉽게 識別할 수 있도록 하기 위하여는 dot pattern이 커야하므로 주어진 畫面에 出力될 수 있는 文字數에 보다 많은 制限을 받게되며 글자가 아름답지 못한 缺點이 있다<sup>6)</sup>.

line generation에 의한 display方法은 주어진 두 좌표로부터  $\Delta x$  및  $\Delta y$ 를 구하기 위하여 adder가 필요하며 이를 거치는 시간 지연 때문에 display control이 늦어질 뿐 아니라 raster수가 많아짐에 따라 각 點을 表示하는 bits數가 增加하여 文字와 같이 많은 짧은 線分으로 構成된 pattern일 경우 필요한 情報量이 다른 方法보다 많아지게 된다<sup>7)</sup>.

vector generation方法은 pattern을 그리기 위한 一連의 vector情報를 만들어야 하는 복잡성이 있으나 基準座標에 대하여 制禦情報가 獨立이기 때문에 反復되어 나오는 pattern은 일종의 subroutine으로 利用될 수 있는 便利한 點이 있으며 線의 增分  $\Delta x$ ,  $\Delta y$ 가 주어지므로 adder를 必要로 하지 않는 利點이 있다<sup>8)</sup>.

本 研究은 vector generation方法에 의하여 한글을 表記할 경우 必要한 最小 vector制禦情報 및 그 情報量의 算出을 그 目的으로 하였으며 이를 위하여 다음의 研究課題들을 履行하였다.

- 1) 한글 字素를 그리는 最短經路를 찾는 algorithm의 構成
- 2) 最短表記經路 및 그 始作點의 發見
- 3) 最小制禦情報를 要하는 vector群의 發見
- 4) 한글 字素들을 그리기 위한 制禦情報의 構成

本 研究은 한글 字素의 크기를 그 위치에 따라 變化시킬 수 있고 그 모양은 vector表記의 성질상 認識을 어렵게 하지 않는 범위에서 原型으로부터 다소 變形시킬 수 있다는 前提條件下에서 이루어졌으며 현재 한글 印刷裝置에 많이 쓰이고 있는 24個의 기본 한글 字母音과 9個의 重字母音(ㅍ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ)을 합한 33個의 字素를 택하였다. 현재 이용되고 있는 display 裝置가 2~3가지로 기본 raster의 크기를 dynamic하게 制禦할 수 있는 function을 갖고 있기 때문에 기본 vector들에 對한 制禦情報만 만들어 주면 한글 字素의 위치에 따른 크기조절은 software의 으로 쉽게 制禦할 수 있을 것이다.

## 2. 한글 字素를 그리는 最短經路를 찾는 algorithm의 構成

vector generation에 의한 文字 pattern을 display하는 方法은 beam의 初期位置가 位置 function에 의하여 software의 으로 주어지고 이를 起點으로 하여 주

어진 文字 pattern을 vector的으로 그려나가기 때문에 beam의 初期位置는 pattern에 관계없이 文字 pattern이 그려질 raster matrix의 어느 한 點으로 주어져야 하며 이 基準點의 位置와 文字의 모양에 따라 必要한 制禦情報量이 달라진다. 英文字의 경우 이 始作點은 raster matrix의 左下端點으로 되어있지만 한글 字素의 경우 pattern을 그리는 最短經路를 쫓 수 있는 始作點은 다를 것이라는 가정하에  $7 \times 7$  raster matrix에 33개의 한글 字素를 最短經路로 그릴 수 있는 始作點 및 그 經路를 찾는 algorithm을 構成하였다.  $7 \times 7$  raster matrix를 택한 것은 현재 이용되고 있는 대부분의 display가 이 정도 크기의 raster matrix에 複雜한 特殊文字도 쉽게 識別할 수 있도록 display할 수 있는 resolution을 갖고 있기 때문이다.

最短經路를 찾기 위하여 우선 주어진 始作點으로부터 vector가 이동할 수 있는 모든 經路를 追跡하여 그 經路의 長이를 比較할 수도 있겠으나 한글 字素의 構造에서 볼 수 있는 일정한 pattern에 쓸 수 있는 다음에 記述되는 것과 같은 4개의 algorithm에 의한 subroutine을 이용한 프로그램을 만들었다. 電子計算機에 의하여 한글 pattern을 주어진 경로를 따라가다 남은 경로가 algorithm이 적용될 수 있는 pattern으로 축소됐을 경우 바로 해당 algorithm에 의하여 분석적으로 나머지 pattern의 最短經路가 決定된다.

algorithm에 이용되는 記號는 그림 1에 의하여 다음과 같이 定義한다.

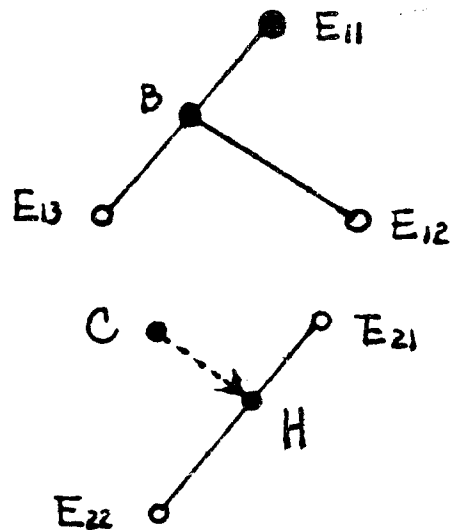


그림 1 algorithm用 pattern記號

$E_{ij}$ :  $i$ 번째 獨立 pattern의  $j$ 번째 끝점

$C$ : beam의 현재 位置

$H$ :  $C$ 에서 가장 가까운 pattern上的 raster점

**Algorithm I**: 끝점이나 마디가 없을 경우(閉回線)

- ① go to  $H$
- ② generate end point
- ③ follow the line
- ④ if end point then stop else go to ③

**Algorithm II**: 두개의 끝점만 있을 경우(1개의 線分)

① if  $\{(\overline{CH} + \overline{E_{11}E_{12}}) \geq \min(\overline{CE_{11}}, \overline{CE_{12}})\}$  then go to end point giving  $\min(\overline{CE_{11}}, \overline{CE_{12}})$  else go to  $H$  and generate end point

- ② follow the line
- ③ if end point then(if end point left then go to ① else stop) else go to ②

**Algorithm III**: 4개의 끝점만이 있을 경우(2개의 線分)

① go to end point which gives  $\min[\{\overline{CE_{21}} + \min(\overline{E_{22}E_{12}}, \overline{E_{22}E_{11}})\}, \{\overline{CE_{22}} + \min(\overline{E_{21}E_{12}}, \overline{E_{21}E_{11}})\}, \overline{CE_{11}} + \min(\overline{E_{12}E_{21}}, \overline{E_{12}E_{22}})\}, \{\overline{CE_{12}} + \min(\overline{E_{11}E_{21}}, \overline{E_{11}E_{22}})\}]$

- ② follow the line
- ③ if end point then go to Algorithm II else go to ②

**Algorithm IV**: 3개의 끝점과 1개의 마디가 있을 경우

- ① go to closest end point
- ② follow the line till hit a node
- ③ go to Algorithm II

### 3. 最小表記經路 및 그 始作點의 發見

위에서 構成된 algorithm을 FORTRAN으로 encode 한후, CDC 3200電子計算機를 이용하여 raster matrix의 各點을 始作點으로 했을 경우 1개의 한글 字素를 最단經路로 그리는데 소요되는 blank vector의 移動 平均길이를 조사한 結果 表-1을 얻을 수 있었다. 모든 한글 字素의 總 線分길이가 일정하므로 最短 blank vector 移動길이를 주는 始作點이 한글을 그리는 最適 始作點으로 raster matrix의 右下端에 있음을 알 수 있다. 英文表記에서 主로 쓰이는 左下端은 右下端에서 보다 平均 1.36의 blank vector移動을 더 要求하고 있어 많은 文字가 display될 경우 display 속도에 큰 影響을 미칠 수 있음을 알 수 있겠다. raster matrix의 右下端을 起點으로한 33개의 한글 字素의 最短經路는 부록과 같다.

6	4.82	5.09	5.13	5.33	5.34	5.45	5.13
5	5.14	5.23	5.38	5.78	5.65	5.49	5.35
4	5.27	5.43	5.60	5.98	5.85	5.73	5.46
3	5.58	5.79	6.05	6.48	6.08	5.83	5.44
2	5.47	5.69	5.88	6.07	5.71	5.46	5.07
1	5.58	5.63	5.68	5.88	5.35	4.96	4.72
0	5.59	5.62	5.39	5.59	5.09	4.76	4.23
$y \backslash x$	0	1	2	3	4	5	6

表-1 각 Raster점에서의 平均 Blank Vector 移動 經路

$\Delta X \backslash \Delta Y$	0	1	2	3	4	5	6
0			12	15	2		40
1	3	7		1			
2	15		3	1			
3	18			7			
4	2						
5							
6	37		8				

表-2 線分 分布

### 4. 最小制禦情報을 要하는 vector群의 發見

한글 字素를 그리는 最短經路가 주어지더라도 線分 길이의 分布 및 선택된 基本 vector群에 따라 制禦情報量이 달라진다. 表-2의 線分分布와 같이  $\Delta x, \Delta y$ 의 크기가 2,3,6에 偏重되어 있어 이와 같은 크기를 가진 vector들이 基本 vector群을 이룰 것임을 예상할 수 있겠다.

일반적으로 raster matrix에 있는 pattern들의 直線分 最大 길이가  $m$ 일 경우 이들을 表記하기 위하여 고려될 수 있는 基本 이동 vector群의 數는  $\sum_{n=1}^m mC_n$  이다. 한글의 경우  $m=6$ 이므로 63개의 基本移動 vector群이 고려된다. 그러나 直線分 길이가 1일 경우 이보다 큰 vector로 表記할 수 없기 때문에 크기 1을

가진 vector는 移動 vector群에 포함시켜야 하며, pattern에 있는 斜線의 기울기를 줄 수 있는 vector쌍이 移動vector群내에 있어야 한다. 예를 들면  $\Delta x=2$ ,  $\Delta y=6$ 으로 주어진 斜線分  $\vec{\Delta x} + \vec{\Delta y}$ 는 1과 6의 크기를 가진 수직 수평 vector들로 表記할 수 없다.

이상의 제한조건을 고려하면 표-2와 같은 線分 分布를 가진 33개의 한글 字素를 그리는데 쓸 수 있는 移動vector群은 {1,2}, {1,3}, {1,2,3}, {1,2,6}, {1,3,6}, {1,2,3,6} 6개이다.

beam의 明/減 및 方向制禦數는 표-3과 같고 여기에 각 移動 vector群이 33개의 한글 字素를 最短經路를 따라 가는데 필요한 移動制禦數를 합한 總 制禦數는 표-4의 둘째欄과 같다. 總 制禦情報量은 각 制禦에 필요한 制禦 word의 bits數에 의하여 결정되는데 制禦 word의 horizontal format는 그림-3과 같이 만들 수 있다. 總 制禦情報는 制禦 word의 길이에 總 制禦數를 곱한 값이 되는데 制禦word는 基本移動 vector數

B	V <sub>c</sub>	H <sub>c</sub>	Vertical Vector Group	Horizontal Vector Group
---	----------------	----------------	-----------------------	-------------------------

Vertical Vector Group  $\subseteq \{V_1, V_2, V_3, V_4, V_5, V_6\}$   
 Horizontal Vector Group  $\subseteq \{H_1, H_2, H_3, H_4, H_5, H_6\}$   
 단, vector  $V_1$  및  $H_1$ 의 크기는 raster 거리와 같고  $V_n = V_1 \times n$ ,  $H_n = H_1 \times n$

그림-3 Horizontal 制禦 Word

를  $N$ 이라 하면 horizontal format의 경우  $3+2N$ , vertical format의 경우 모든 制禦경우의 數 즉  $7+2N+N^2$ 을 encode할 수 있는 최소 bits數가 된다. 각 vector群에 대한 必要制禦情報量은 표-4와 같다.

B	V <sub>c</sub>	H <sub>c</sub>	V <sub>c</sub> B	H <sub>c</sub> B	V <sub>c</sub> H <sub>c</sub> B	Total
85	10	8	38	26	3	170

· 단, {1, 3} vector群을 택했을 때는 H<sub>c</sub> 제어가 1개 더 추가되어 171이 됨

표-3 明/減 및 vector 方向制禦

표-4에서 볼 수 있는 바와 같이 horizontal format를 택했을 경우 基本移動 vector群을 {1,3}으로 택할 경우 總制禦情報量이 3,241 bits로 가장 적으며 vertical format의 경우 {1,2,3,6}의 vector群이 1,725 bits의 最小情報量을 주고 있음을 알 수 있다. vertical format를 이용할 경우 {1,2,3,6}의 vector群을 택하면 5×32의 비교적 큰 decoder가 있어야 하며 LSI memory chip을 이용할 경우 制禦 word 5 bits가 한 byte에 맞지 않으므로 생기는 word 割當問題를 고려한다면 情報量은 다소 많지만 4 bits로 構成될 수 있는 {1,3}

기본이동 vector群	총제어수	제어 word 길이 (수평 format)	총제어 bits	제어 word 길이 (수직 format)	총제어 bits
(1,2)	553	7	3,871	4	2,212
(1,3)	463	7	3,241	4	1,852
(1,2,3)	430	9	3,870	5	2,150
(1,2,6)	391	9	3,519	5	1,955
(1,3,6)	386	9	3,474	5	1,930
(1,2,3,6)	345	11	3,795	5	1,725

표-4 33개의 한글 字素의 vector 表記 情報量

vector群을 택하는 것이 바람직하다고 볼 수 있다. {1,3} vector群을 선택했을 때 '츠'字 表記의 일부인 線分 vector ( $\Delta x=3$ ,  $\Delta y=2$ )의 表記가 불가능하나 마침 이 vector가 blank vector이므로 ( $\Delta x=0$ ,  $\Delta y=2$ ), ( $\Delta x=3$ ,  $\Delta y=0$ )의 두 線分 vector로 분리시켜 그릴 수 있으며 이 때문에 순해보는 blank vector의 길이는 평균 0.03 밖에 되지 않으므로 표-1의 분석에 영향을 주지 않음을 알 수 있다.

### 5. Vector表記 制禦情報의 構成

이상의 研究에서 얻어진 基本 移動 vector群 {1,3} 즉  $V_1, V_3, H_1, H_3$  및 beam 明/減 制禦 B, 移動 vector의 수평 및 수직 方向制禦 V<sub>c</sub>, H<sub>c</sub> 각각에 1개의 bits를 주어 構成된 制禦 word format는 그림-4와 같다.

B	V <sub>c</sub>	H <sub>c</sub>	V <sub>1</sub>	V <sub>3</sub>	H <sub>1</sub>	H <sub>3</sub>
---	----------------	----------------	----------------	----------------	----------------	----------------

그림-4 {1,3} 移動 vector群에 의한 제어 word format

이 制禦 word를 이용 부록의 33개의 한글 字素를 표시된 경로를 따라 그리는 基本 移動 vector에 대한 制禦情報는 표-5와 같다. 便利上 各 制禦 word는 set된 bit名만으로 表示하였다. vertical format로 制禦情報를 表示하려면 나열된 各 制禦情報에 해당하는 vertical code를 주기만 하면 된다.

### 5. 結 論

이상의 研究結果를 종합하여 다음과 같은 結論을 얻을 수 있었다.

1) 33개의 한글 字素를 그리는 最短經路를 주는 display beam의 如作點은 文字 表記用 raster matrix의 右下端 raster點이다.

2) 수직, 수평 모두 1과 3의 基本 vector群을 선택했을 때 7 bits 463制禦 words (horizontal format)

즉 3,241 bits의 最小制御情報을 만들 수 있었다.

3) hardware에의 응용시 주로 이용되는 ROM의 1 byte 中 7 bits를 制御 bits로, 나머지 1 bits를 EOR을 위한 flag bit로 이용한다면 463 bytes의 ROM영역에 本 研究에서 선택한 33개의 한글을 vector表記하기 위한 制御情報을 넣을 수 있을 것이며, 制御 word를 4 bits로 encode된 vertical format로 만든다면 word當 4 bits가 소요되어 33개의 EOR까지 포함하여 248 bytes의 ROM 容量이 소요될 것이다.

- ㄱ : B, V<sub>3</sub>, V<sub>3</sub>, H<sub>3</sub>, H<sub>3</sub>, B
- ㅋ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, B
- ㆁ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, B
- ㄴ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, B
- ㄷ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, Vc, V<sub>3</sub>, V<sub>3</sub>, B
- ㄹ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>3</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, VcB, V<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㅁ : B, V<sub>3</sub>H<sub>3</sub>, B, V<sub>3</sub>, BVc, V<sub>3</sub>, V<sub>3</sub>H<sub>3</sub>, B
- ㅇ : V<sub>1</sub>H<sub>1</sub>, VcB, V<sub>1</sub>H<sub>1</sub>, H<sub>1</sub>, H<sub>1</sub>, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>, V<sub>1</sub>, Hc, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>H<sub>1</sub>, H<sub>1</sub>, H<sub>1</sub>, Vc, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>, V<sub>1</sub>, V<sub>1</sub>H<sub>1</sub>, B
- ㅂ : B, V<sub>3</sub>H<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>3</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, VcHcB, V<sub>3</sub>H<sub>3</sub>, HcB, V<sub>3</sub>H<sub>3</sub>, B
- ㅅ : B, V<sub>3</sub>H<sub>3</sub>, B, H<sub>3</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, HcB, V<sub>2</sub>, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>H<sub>3</sub>, B
- ㅇ : B, V<sub>3</sub>, V<sub>3</sub>, H<sub>3</sub>, H<sub>3</sub>, VcB, V<sub>3</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B
- ㅈ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>1</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, B, V<sub>1</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B
- ㅊ : B, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, B, V<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㅋ : V<sub>1</sub>, VcB, V<sub>1</sub>H<sub>1</sub>, H<sub>3</sub>, H<sub>1</sub>, Vc, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>, V<sub>1</sub>, Hc, V<sub>1</sub>H<sub>1</sub>, H<sub>3</sub>, H<sub>1</sub>, Vc, V<sub>1</sub>H<sub>1</sub>, V<sub>1</sub>, V<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, V<sub>1</sub>, B, H<sub>3</sub>, B
- ㆁ : V<sub>1</sub>, V<sub>1</sub>, B, H<sub>3</sub>, H<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B
- ㄴ : B, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>3</sub>, B, H<sub>3</sub>, H<sub>3</sub>, B
- ㄷ : B, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, B, H<sub>3</sub>, H<sub>3</sub>, B, V<sub>1</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B
- ㄹ : B, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>3</sub>, B, V<sub>3</sub>, V<sub>3</sub>, B
- ㅁ : B, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, V<sub>3</sub>, V<sub>3</sub>, B, H<sub>1</sub>, H<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㅂ : H<sub>3</sub>, B, V<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>3</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>
- ㅇ : H<sub>1</sub>, H<sub>1</sub>, B, V<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, HcB, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㅅ : V<sub>3</sub>, B, H<sub>3</sub>, H<sub>3</sub>, B
- ㅇ : H<sub>3</sub>, B, V<sub>3</sub>, V<sub>3</sub>, B
- ㅈ : B, V<sub>3</sub>, V<sub>3</sub>, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B

- ㅊ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, Hc, H<sub>3</sub>, H<sub>3</sub>, HcB, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㅋ : B, H<sub>3</sub>, H<sub>3</sub>, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>3</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B, V<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㆁ : B, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, Vc, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, VcHcB, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, Vc, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, B
- ㄴ : B, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, Vc, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, HcB, H<sub>1</sub>, H<sub>1</sub>, VcHcB, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, Vc, V<sub>3</sub>H<sub>1</sub>, V<sub>3</sub>H<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, HcB, H<sub>3</sub>, B
- ㄷ : B, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>3</sub>, B, H<sub>3</sub>, H<sub>3</sub>, VcB, V<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, B
- ㄹ : B, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, B, H<sub>3</sub>, H<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, H<sub>3</sub>, B
- ㅁ : B, V<sub>3</sub>, V<sub>3</sub>, B, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>3</sub>, H<sub>3</sub>, H<sub>3</sub>, B
- ㅇ : B, V<sub>3</sub>, V<sub>3</sub>, B, H<sub>3</sub>, VcB, V<sub>3</sub>, V<sub>3</sub>, VcB, V<sub>1</sub>, V<sub>1</sub>, B, H<sub>3</sub>, B, V<sub>1</sub>, V<sub>1</sub>, HcB, H<sub>3</sub>, B

표-5 Vector Control 정보

參 考 文 獻

- (1) 李柱根, 한글 文字의 電子計算組織에 適應하기 위한 特徵抽出에 關한 研究, 대한전자공학회지, 6, 198, (1969)
- (2) 강린구, 이행세, 한글자체의 특징추출의 한 방식, 대한전자공학회지, 6, 175, (1969)
- (3) 安秀楛, 電子計算機의 한글 入出力에 關한 方案 案, 대한전자공학회지, 10, 74, (1973)
- (4) 李柱根, 한글文字의 認識에 關한 研究(IV), 대한전자공학회지, 9, 197, (1972)
- (5) 安秀楛, 한글문자 모아쓰기 display方案, 대한전자공학회지, 12, 27, (1975)
- (6) 박승규, 컴퓨터에 의한 한글기계화연구, 한국과학기술 학위논문, (1976)
- (7) Newman, Spraul, Interactive Computer Graphics, McGraw-Hill, pp.21~80. (1973)
- (8) 李柱根, 朱薰, 한글 文字의 認識을 위한 代數的 構造, 대한전자공학회지, 12, 2, (1975)
- (9) CDI, Visual Displays, CDC, pp.5-1~7-9, (1967)
- (10) IBM Korea, Programmers Guide Using Hangl, 20, (1969)
- (11) UNIVAC, 한글 Data처리에 대한 설명서, KD-7201, 40, (1968)

부록: Vector 표기 최단 경로

