論文 92-29B-8-5

문자영상의 중심화소 추적 알고리듬 및 신경칩 설계

(The Tracing Algorithm for Center Pixel of Character Image and the Design of Neural Chip)

高 輝 鎭*, 呂 珍 璟*, 鄭 鎬 宣*

(Hwi Jin Ko, Jin Kyung Ryeu, and Ho Sun Chung)

要 約

문자영상의 중심화소를 추적하는 트레이싱(tracing) 알고리듬을 개발하였다. 문자영상의 입력장치로는 스캐너를 이용하였다. 트레이싱을 수행하면서 문자인식에 필요한 특징점인 가지점과 4방향의 스트록(stroke)을 추출하였다. 트레이싱 알고리듬은 기존의 문자인식의 전처리 과정인 세선화 과정을 비롯한 모든 과정을 한번에 수행하는 것으로, 전처리 과정의 수행시간을 5배 단축시켰다. 그리고 전처리 과정용칩을 신경 회로망 모델인 단층구조 퍼셉트론 알고리듬을 이용하여 설계하였다.

Abstract

We have presented the tracing algorithm for center pixel of character image. Character image was read by scanner device. Performing the tracing process, it can be possible to detect feature points, such as branch point, stroke of 4 directions. So, the tracing process covers the thinning and feature point detection process for improving the processing time. Usage of suggested tracing algorithm instead of thinning that is the preprocessing of character recognition increases speed up to 5 times. The preprocessing chip has been designed by using single layer perceptron algorithm.

I. 서 론

영어는 정보의 수집 및 교환수단으로 널리 사용되고 있으며, 이로 인해 처리해야 할 정보는 날로 늘어나고 있다. 그래서 정보의 처리과정을 사람에 의존하기보다는 이 과정을 기계로써 대행하려는 연구의 일환으로 영문자의 자동인식에 관한 연구가 활발히 진행되고 있다.[1~7] 카메라나 이미지 스캐너와 같은 영상입력 장치를 통하여 입력된 영상정보가 "1"이나 "0"으로 구성되어 있는 임의의 화상으로 주어졌을 때 이 화상에서 문자나 물체의 인식을 위하여 전처리 과정을 수행한다. 입력된 문자영상 정보에는 다수의 문자가 존재한다. 따라서 문자 분리과정에서 문자열로부터 개별문자를 분리한 후, 분리된 개별문자에 대해 잡음제거 이후의 모든 과정을 수행한다. 잡음제거 및 선형화 과정에서는 불필요한 잡음을 제거하고 문자영상의 모서리 부분에 존재하는 요철을 제거한다. 그리고 문자를 인식하기 위해서는 특징점들을 추출하여야 한다. 특징점 추출을 용이하게 하기 위해서는 문자를 하나의 화소폭으로 나타내는 세선화 과정이 필요하다. 위와 같은 영문자 인식 방법에서는 대부분의 수행시간을 세선화 과정에서 소모하게 된다.

본 논문에서는 이러한 인식속도의 한계성을 극복하기 위하여 시간이 많이 걸리는 세선화 과정[8] 대신에 문자영상의 중심화소만을 추적하면서 문자의 선폭을 하나의 화소폭으로 변화시키고, 동시에 특징점을 추출하는 트레이싱 알고리듬[9]을 개발하고 신경칩화함으로써 이를 해결

*正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)
接受日字 : 1990年 12月 26日

하고자 한다. 문자입력 장치로는 스캐너를 사용했으며 영문자 인식을 위해 사용된 특징점으로는 트레이싱 과정에서 추출한 가지점과 4방향 스트록 각각의 갯수이다. 영문자 52자 및 기본적인 문장 기호 29자를 그 대상으로 하였다. 영문자 인식을 위해 사용된 자형은 국정 교과서인 중학교 영어 교과서를 사용하였다. 전처리 과정용 칩을 신경회로망 모델인 단층구조 퍼셉트론 알고리듬을 이용하여 설계하였고, 설계된 회로에서는 데이타가 병렬처리됨으로써 보다 빠른 처리 속도를 얻을 수 있다.

Ⅱ. 영문자 인식과정

영문자를 인식하기 위한 전체적인 소프트웨어 구성은 그림 1과 같다. 입력 영상은 이미지 스캐너를 이용하여 200 DPI(Dot Per Inch)의 해상도로 받은 이진 영상이다. 그림 2는 입력된 문자 영상이다. 입력된 영상은 먼저 문자열에서부터 개별 문자로 분리하는 과정을 거치게 된다. 분리된 영상은 문자의 선폭이 두껍게 나타나므로 특징점들을 추출하기가 어렵다. 기존의 방법[6]으로는 세선화 과정을 수행하여 두꺼운 문자 선폭을 하나의 문자선폭으로 변화시킨 다음 마스크 매칭방법등을 이용하여 특징점

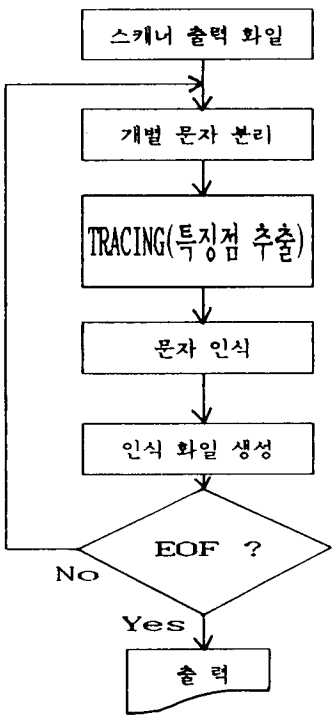


그림 1. 문자 인식을 위한 전체적인 흐름도
Fig. 1. Flowchart for character recognition.

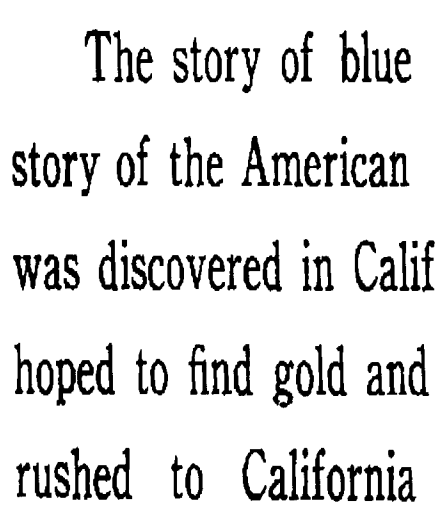


그림 2. 이미지 스캐너로부터 입력된 문자 영상
Fig. 2. Scanned character image by scanner.

을 추출하였으나 이와 같은 전처리 과정은 상당한 시간을 필요로 하므로 이진화된 문자영상의 중심 화소만을 추적하면서 문자의 선폭을 하나의 화소폭으로 변화시키고 동시에 특징점을 추출하는 트레이싱 과정을 수행한다. 트레이싱 과정에서 문자 인식에 필요한 특징점들을 추출한다. 특징점으로는 가지점과 4방향의 스트록이 있다.

Ⅲ. 문자 인식의 전처리 과정

1. 개별문자 분리과정

그림 2는 이미지 스캐너를 이용하여 해상도 200 DPI (Dot Per Inch)로 받은 문자 영상이다. 일반적으로 문서는 그림 2와 같이 문자열의 집합속에 개별 문자를 가지고 있다. 따라서 문자 인식을 위해서는 먼저 문자열에서부터 개별 문자를 분리한 후, 이들 개별 문자에 대해 순차적으로 문자를 인식하게 된다. 개별 문자를 분리하기 위한 알고리듬은 다음과 같다. 먼저 좌표점(0,0)에서부터 하나의 수평 라인에 문자를 형성하는 오브젝트 화소가 나타날 때까지 추적한다. 오브젝트를 이루는 화소가 처음으로 나타나는 점의 y 좌표값이 하나의 문자열이 시작되는 y축 좌표값이다. 이 좌표값을 iy라고 둔다. 그 다음에는 이 y좌표로부터 y축 좌표값을 증가시키면서 하나의 수평라인 전체에 걸쳐 문자를 이루는 오브젝트 화소가 하나도 없는 최초의 y 좌표까지 증가시킨다. 이때의 y 좌표가 하나의 문자열이 끝나는 y축 좌표값이다. 이 좌표값을 l_y라고 둔다. 이런 알고리듬으로 하나의 문자열이 존재하는 y축의 좌표(i_y, l_y)를 찾을 수 있다. 이렇게 하나

의 문자열이 분리되면 다음에는 수직으로 문자를 분리하여 문자열에서 개별 문자들을 분리한다. 먼저 문자열이 시작되는 y좌표, 즉 i_y에서 하나의 문자열이 끝나는 y좌표, 즉 l_y까지 문자열의 제일 왼쪽에서부터 y축 방향으로 추적한다. 이렇게 추적하다가 문자를 형성하는 오브젝트 화소가 처음으로 나타나면 이점의 x좌표값이 하나의 문자가 시작되는 x축 좌표값이다. 이 값을 ix라 둔다. 다음에는 이 x좌표로부터 x좌표값을 증가시키면서 i_y에서 l_y까지의 수직 라인 전체에 걸쳐 문자를 이루는 오브젝트 화소가 하나도 없는 최초의 x좌표를 찾는다. 이때의 x좌표가 이 문자가 끝나는 x축 좌표값이다. 이값을 lx라 둔다. 이렇게 하여 하나의 개별 문자가 존재하는 영역 (i_x, i_y)에서 (l_x, l_y)까지를 찾을 수 있다. 이상의 알고리듬을 입력된 문자 영상전체에 걸쳐 수행시키면 입력된 문자열에서 개별 문자를 분리하여 문자를 인식할 수 있다. 그림 3은 입력된 문자 영상에서 개별 문자를 분리한 경우이다.

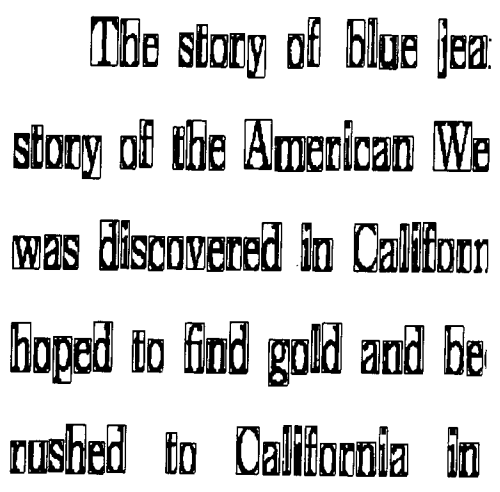


그림 3. 스캔된 영상에서 문자가 분리된 영상
Fig. 3. Separated character image from scanned image.

2. 트레이싱 과정

영상 입력 장치로 부터 입력된 문자 영상은 그림 2에서처럼 문자의 선폭이 두껍게 나타난다. 트레이싱 과정이란 이렇게 두꺼운 선폭으로 나타나는 문자의 폭을 그림 4의 5×5 위도우로 추적하면서 문자선폭의 중심화소를 찾아 하나의 화소폭으로 변화시키고 동시에 특징점을 추출하는 과정을 말한다. 이 알고리듬의 과정을 살펴보면 다음과 같다.

0	1	1	1	0
1	4	4	4	1
1	4	16	4	1
1	4	4	4	1
0	1	1	1	0

(a)

	10	11	12	
21	9	2	3	13
20	8	1	4	14
19	7	6	5	15
	18	17	16	

(b)

그림 4. Tracing 윈도우
(a) 5 · 5 윈도우
(b) 윈도우의 요소번호
Fig. 4. Tracing window.
(a) 5 · 5 window,
(b) Element number of window.

[과정 1] 이미지 스캐너로 부터 입력된 문자 영상은 문자 영역과 배경영역이 서로 분리된 이진 영상으로 화일이 생성된다. 이러한 이진 영상으로 부터 각 문자의 크기를 점검하여 전체 영상에서 하나의 문자만을 분리해 낸다.

[과정 2] 이미지 스캐너로 부터 받은 문자 영상의 화소값은 "0"과 "255"의 값을 가진다. 먼저 문자 영상의 화소값 "0"은 "1"로, "255"는 "0"으로 이진화시킨다.

[과정 3] 이진화된 문자 영상에서 5×5 윈도우의 가중치가 처음으로 "40"보다 큰 값을 가지는 화소를 트레이싱의 시작점으로 잡는다.

[과정 4] 5×5 윈도우를 8방향으로 각각 이동시켜 각 방향의 윈도우 가중치를 구한다.

[과정 5] 최대 윈도우 가중치 방향으로 5×5 윈도우를 이동시키고, 이 방향과 90°, 135°의 방향 차이가 나고, 이 방향들의 윈도우 가중치가 "35"보다 크면 이 방향들의 5×5 윈도우의 중심 화소를 가지점의 후보점으로 할당해 둔다. 트레이싱하는 5×5 윈도우의 중심 화소는 항상 트레이싱점으로 저장하고 5×5 윈도우가 지나온 문자 영상은 지운다.(화소 값="0")

[과정 6] 한 스트록을 트레이싱하면서 그 스트록의 방향을 저장하고 가장 많이 나타나는 방향을 그 스트록의 방향으로 한다. 한 스트록에 대한 트레이싱이 끝나면 할당해 두었던 가지점의 후보점으로 5×5 윈도우를 이동한다.

[과정 7] 가지점의 후보점의 5×5 윈도우의 가중치가 "30"보다 크면 이점을 가지점으로 저장하고 모든 문자 영상의 윈도우 가중치가 "40"보다 클때까지 과정을 반복한다.

[과정 8] 5×5 윈도우의 가중치가 "40"보다 큰 화소가 없을때까지 수행시키면 하나의 화소로만 연결된 트레이싱 영상을 얻게 된다.

그림 5. 영문자의 tracing 영상
Fig. 5. Traced image of English character.

그림 5는 문자 영상을 트레이싱한 결과이다. Parameter (시작점, 가지점의 후보점) 값을 작게하였을 경우 끝점부근에서 말리는 현상이 나타나고, 크게하였을 경우에는 tracing이 끝까지 되지않아 옳바른 특징점추출을 할 수 없다. 본연구에서 사용한 parameter는 어떤 일정 규칙에 의해 정해진 것이 아니라 많은 실험을 행한후 최종결정된 최적치이다. 그림 6과 7은 한글과 한자에 대한 트레이싱 영상이다. 표 1은 세선화 과정을 포함하는 전처리 과정과 본 연구에서 제안한 트레이싱 과정의 한 문자당 수행 시간을 비교한 것이고, 표 2는 전체 문자 163자에 대한 총 수행시간을 비교한 것이다. 표 1과 표 2에서 트레이싱 과정이 세선화 과정보다 수행 속도가 5배 정도 빠름을 알 수 있다.

3. 특징점 추출 과정

본 연구에서는 트레이싱 과정에서 특징점을 추출하여 이들로부터 문자를 인식하고자 한다. 인식을 위해 필요한 특징점들은 가지점(branch point)과 4방향의 스트록 갯수의 다섯가지 종류로 분류된다. 먼저 가지점은 트레이싱 과정에서 가지점의 후보가 가지점으로 인정된 점이고, 4방향의 스트록 갯수는 한 스트록을 트레이싱 하면서 가장 많이 나타나는 방향을 그 스트록의 방향으로 한다.

표 1. 하나의 문자에 대한 세선화와 트레이싱의 수행시간 비교
Table 1. Compare of processing time of thinning and tracing for single character.

전처리 과정	세선화 (초)	Tracing (초)
잡음 제거	0.2	
선 형 화	0.15	
세 선 화	0.8	0.32
특징점 추출	0.4	
총 수행시간	1.55	0.32

표 2. 163 문자에 대한 세선화와 트레이싱의 수행시간 비교
Table 2. Compare of processing time of thinning and tracing for 163 characters.

전처리 과정	세선화 (초)	Tracing (초)
잡음 제거	39.12	
선 형 화	29.34	62.60
세 선 화	156.48	
특징점 추출	78.24	
총 수행시간	303.18	62.60

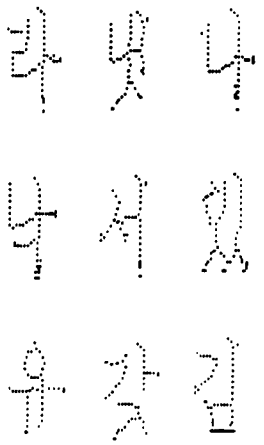


그림 6. 한글문자의 tracing 영상
Fig. 6. Traced image of Korean character.

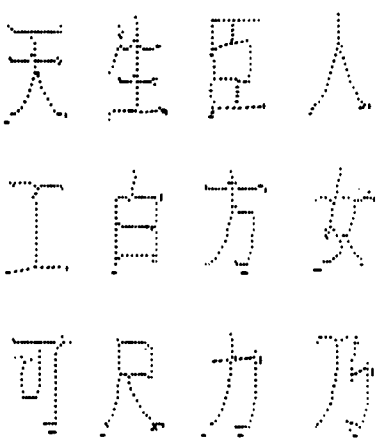


그림 7. 한자의 tracing 영상
Fig. 7. Traced image of Chinese character.

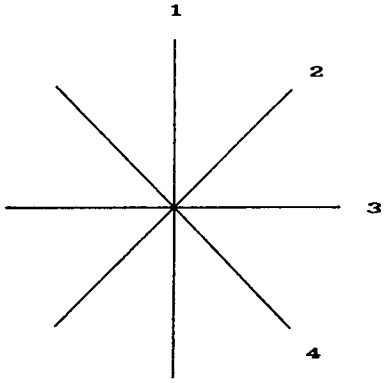


그림 8. Stroke의 방향
Fig. 8. Direction of stroke.

그림 9. 특징점 추출 결과
Fig. 9. Result of feature point extraction.

그림 8은 스트록의 방향을 표시한 것이고 그림 9는 특징점 추출 결과를 보여주는 것으로 가지점은 "B"로 표시되어 있다.

Ⅳ. 신경 회로망을 이용한 트레이싱 과정 회로설계

1. 트레이싱 과정의 회로동작 원리와 구성

트레이싱 과정의 회로 구성은 그림 10과 같다. 앞에서 설명한 트레이싱 과정에서 먼저 문자 영상을 이진화 시키고 트레이싱의 출발점을 구하고 출발점에서 1방향(출발점의 좌표값에서 y좌표값 2감소)으로 좌표값을 변화시킨다. 그림 11은 트레이싱의 8방향을 나타낸 것이다. 1번 윈도우의 가중치는 16이고 2번부터 9번까지의 윈도우의 가중치는 각각 4이다. 그리고 10번부터 21번까지의 윈도우의 가중치는 각각 1이다. 12 to 4 1's 카운터는 가중치가 1인 윈도우 요소번호 10번부터 21번까지에서 픽셀이 존재하는 갯수를 센다. 이 갯수가 곧 가중치가 1인 윈도우의 가중치 합이 된다. 8 to 4 1's 카운터는 가중치가 4인 윈도우 요소번호 2번부터 9번까지의 픽셀이 존재하는 갯수를 센 다음, 두 비트를 왼쪽으로 이동(2^2배)하여 가중치가 4인 윈도우의 가중치 합을 구하였다. 12 to 4 1's 카운터의 상위 두 비트와 8 to 4 1's 카운터의 하위 두 비트를 강제로 0의 값을 주어 두 카운터의 출력을 하위 3 비트를 더하는 3-비트 가산기와 상위 3 비트를 더하는 3-비트 가산기로 구성하여 두 카운터의 출력을 더하였다. 6-비트 가산기를 사용하지 않고 3-비트 가산기 둘로 회로를 구성한 이유는 신경 회로망은 자극과 억제에 해당하는 비트들을 무한히 늘일 수 없고, 비트 수가 늘어나면 입력 비교 단계에서 정밀도가 떨어지고 특히 공정에 있어서 채널의 폭과 길이(W/L) 값에 대한 오차와, 이동도에 의한 오차로 문턱치 값이 벗나갈 경우 실제 구현이 어렵기 때문이다. 이렇게 구한 값과 가중치

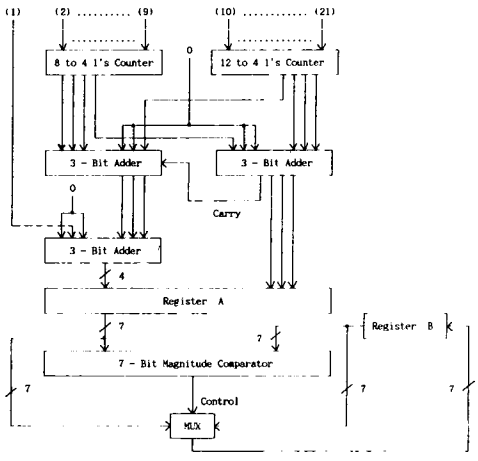


그림 10. Tracing 과정의 블럭도
Fig. 10. Block diagram of tracing processor.

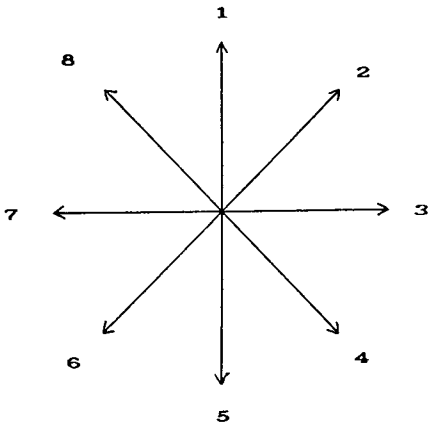


그림 11. Tracing 방향
Fig. 11. Direction of tracing.

가 16인 윈도우 요소번호 1을 최종적으로 더하고 결과를 레지스터 A에 저장 한다. (0000000)으로 초기화 되어있는 레지스터 B와 레지스터 A의 값을 불러와 7-비트 크기 비교기에서 두 레지스터의 값을 비교한다. 7-비트 크기 비교기의 출력은 레지스터 A의 값이 레지스터 B의 값보다 크거나 같으면 1을, 아니면 0을 발생시킨다. 이 값을 먹스(MUX)의 컨트롤로 주어 1이면 레지스터 A의 값을 레지스터 B로 전송하고 0이면 레지스터 B의 값을 그대로 레지스터 B로 전송한다. 레지스터 B의 값을 소프트 웨어에 저장한다. 다음 단계는 좌표값을 2의 방향으로 변화시키고 위의 과정을 반복 수행한다. 이 과정을 8방향까지 수행한 후, 가장 큰값의 5·5윈도우 가중치합을 갖는 방향으로 윈도우의 좌표값을 변화시키고 트레이싱을 수행한다.

2. 단일 방향성 귀환형태 신경 회로망 모델

신경 회로망은 다른 물리적인 회로와 마찬가지로 회로의 에너지가 감소하는 방향으로 회로의 상태가 수렴한다. 따라서 우리가 얻고자하는 해를 에너지 함수의 최소점에 위치시킴으로서 해를 얻을 수 있는 회로를 구현할 수 있다. 이러한 에너지 최소화 기술을 사용하여 병렬 신경회로인 Analog-to-Digital Converter(ADC)가 Tank와 Hopfield에 의하여 설계되었다. 이와같은 개념의 신경회로망은 그 구조와 특성면에서 현재의 디지탈 컴퓨터와는 근본적으로 다르다.

그림 12는 4-비트 ADC 배선도로서 인버팅 출력값만 갖는 4개의 증폭기로 이루어져 있다. 이 ADC 회로망은 한 증폭기의 출력과 다른 증폭기의 입력사이를 연결하는 피드백 저항들로 구성된다. 맨윗줄의 저항들은 각각 다른

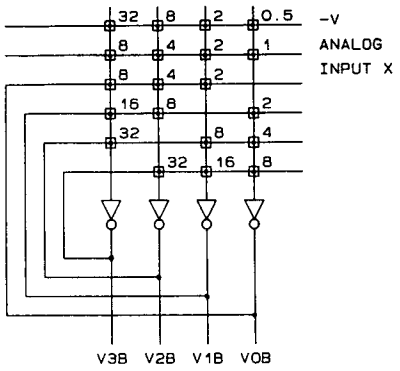


그림 12. 신경 회로망 4-비트 A/D 변환기 배선도
Fig. 12. Wire diagram of the 4-bit neural A/D converter.

값의 일정 전류들을 증폭기의 입력선에 공급한다. 둘째줄의 저항들은 이 회로에 의해 변환될 아날로그 입력전압 X에 비례하는 전류를 증폭기의 입력선에 주입하는 역할을 한다. 변환될 입력전압 X가 여러가지 값을 가지되 증폭기의 출력상태가 2진으로서 아날로그 입력전압에 수치적으로 대응될때 이 ADC는 정상적으로 동작한다고 할수 있다. 이를 수식적으로 표현하면 다음과 같다.

출력 V3_V2_V1_V0가 입력전압 X에의 가장 적합한 디지탈 표현이 되기 위해서는 각각의 V_i 값이 0또는 1의 값을 가지고 1과 0의 선택된 특정의 집합은 아날로그 신호를 표현하는 최적의 것이어야 한다. 따라서 다음의 에너지 함수 E가 최소가 되어야 한다.

$$E=\frac{1}{2}\left(x-\sum_{i=0}^{2} v_i * 2^i\right)^2$$

이 에너지 함수는 여러 최소값중에서 초기 상태에 가장 가까운 최소의점으로 수렴하기 때문에 항상 최상의 해를 낸다는 보장은 없다. 그리고 뉴론들 사이의 내부 연결은 VLSI 구현을 위해서는 다소 복잡하다. 본 논문에서는 트레이싱 회로를 VLSI로 구현하기 위하여 본 연구에서 개발한 단일 방향성 귀환형태(Unidirectional Feedback Type : UFT)의 특별한 목적의 신경 회로망을 제안한다. 그림 13은 UFT 신경회로 모델이다. 제안된 UFT 모델에서 귀환되는 연결수는 Hopfield 모델과 비교하여 절반으로 줄어든다.

3. 단일 방향성 귀환 형태 1's 카운터 설계

그림14는 단일 방향성 귀환 형태의 새로운 8 to 4 1's 카운터이다. IN1-IN8은 카운터의 입력이고 SO, S1, S2, S3은 이 회로의 출력을 나타낸다. 입력단위 PMOS의 컨덕턴스 비는 모두 1로 하고 VDD행의 NMOS는 각각 1, 2, 4, 8이며 귀환되는 NMOS는 2, 4, 8의 값을 가진다. 이 회로의 동작을 살펴 보면 초기값 입력이 모두

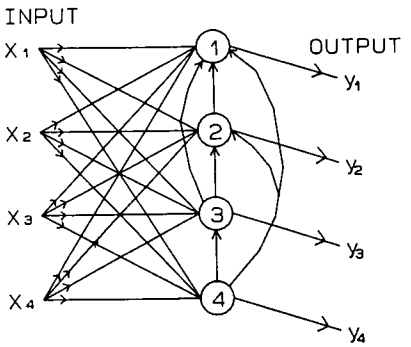


그림 13. 단일 방향성 귀환 형태 모델
Fig. 13. Unidirectional feedback type model.

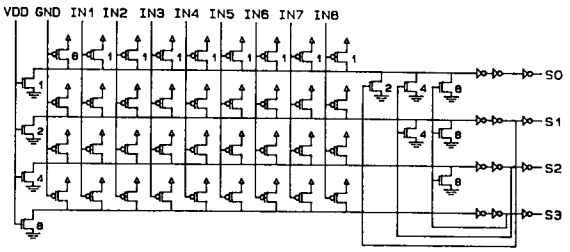


그림 14. 8 to 4 1's 카운터
Fig. 14. 8 to 4 1's counter.

0일때 NMOS에 의한 전류 싱크보다 PMOS에 의한 전류 공급이 크므로 증폭기의 입력단 전위는 문턱값 보다 높은 위치에 있게 된다. 따라서 증폭기의 출력은 1값이 되고 이 값이 귀환 되어 NMOS 게이트에 공급된다. 전체적으로 보면 8, 4, 2, 1 단위 전위만큼 높은 위치에 있게 된다. 최종 출력은 인버터를 거치므로 출력은(0000)이 된다. 입력값 중 어느 하나가 1의 값을 가지면 그 입력에 연결된 PMOS는 OFF되고, 증폭기의 입력단 전위는 7, 3, 1, 0이 되어 출력은 인버터를 거쳐서(0001)이 된다. 출력단 S1의 값이 NMOS Tr.에 귀환되어 컷 오프(cut off)됨으로써 컨덕턴스 2에 해당하는 전류가 증폭기의 입력단에 공급되는 것과 같은 효과가 나타난다. 따라서 각 증폭기의 입력단 전위는 6, 2, 0, 1이 되고 인버터를 거친 출력은(0010)가 된다. 이 회로는 8개의 입력이 4개의 뉴론에 병렬로 가해져 4개의 CMOS 인터터만 통해 결과를 얻게 되므로 회로 동작이 매우 빠르다는 것을 알 수 있다. 그림15은 12 to 4 1's 카운터이다.

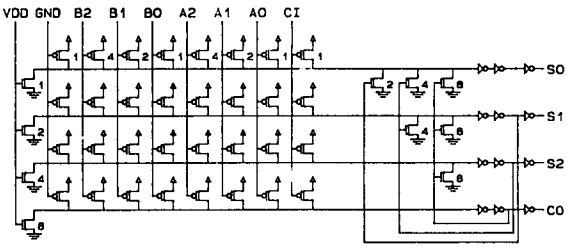


그림 15. 12 to 4 1's 카운터
Fig. 15. 12 to 4 1's counter.

4. 3-비트 가산기 설계

그림16은 3-비트 가산기이다. A2-A0, B2-B0는 가산기의 입력이고 CO, S2, S1, SO는 이 회로의 출력을 나타낸다. 입력단위 PMOS의 컨덕턴스 비는 모두 1로 하고 VDD행의 NMOS는 각각 1, 2, 4, 8이며 귀환되는 NMOS는 2, 4, 8의 값을 가진다. 이 회로의 동작을 살펴보면 입력값이(A2 A1 A0)가 (001)이고 (B2 B1 B0)가 (000)이면 1의 입력에 연결된 PMOS는 OFF되고 나머지 PMOS들은 ON되어 출력단위 값은 모두 "High" 상태가 된다. 이때 캐리 입력인 CI는 0이다. 이 값들이 버퍼를 통과하고 피드백 되어 2, 4, 8의 컨덕턴스를 가지고 NMOS를 구동시켜 SO단의 PMOS 컨덕턴스는 15이고 NMOS 컨덕턴스도 15가 된다. 그렇지만 PMOS의 W/L값은 5μm/2μm이고 NMOS의 W/L값은 2μm/2μm이여서 버퍼단의 출력값은 0이 되고 이 값이 인버터를 통하면 1의 값이 출력된다. 나머지(S1 S1 CO)의 값은 (000)이 된다. 따라서 최종적으로(0001)의 값을 출력하게 된다.

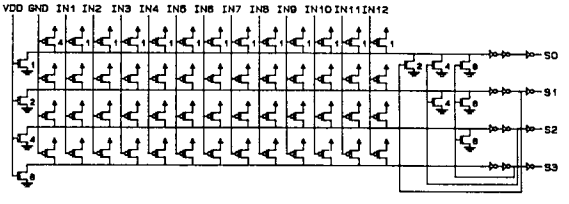


그림 16. 3-비트 가산기
Fig. 16. 3-bit adder circuit.

5. 비트 크기 비교기 설계

7-비트 크기 비교기의 설계는 앞에서도 설명한 것과 같은 신경 회로망의 특성 때문에 실제 구현이 어렵다. 그림 17는 7-비트 크기 비교기를 4-비트 크기 비교기

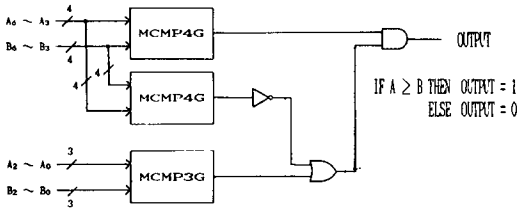


그림 17. 7－비트 크기 비교기
Fig. 17. 7－bit magnitude comparator circuit.

둘과 3－비트 크기 비교기 하나를 사용하여서 구현한 것이다. 레지스터 A의 상위 4비트인 A_6－A_3와 레지스터 B의 상위 4비트인 B_6－B_3를 4－비트 크기 비교기에서 비교하여 레지스터 A의 값이 레지스터 B의 값보다 크거나 같으면 4－비트 크기 비교기의 출력은 1을 나타낸다. 두개의 4－비트 크기 비교기에 레지스터 A의 상위 4비트와 레지스터 B의 상위 4비트를 입력 순서를 바꾸어 입력시켰을때 두개의 4－비트 크기 비교기의 출력이 모두 1이면 레지스터 A의 상위 4비트와 레지스터 B의 상위 4 비트는 같음을 알 수 있다. 이러한 경우 각 레지스터의 하위 3비트를 비교하여야 한다. 하위 3비트를 비교하는 3－비트 크기 비교기에서 레지스터 A의 값이 레지스터 B의 값보다 크거나 같으면 1을 출력하여 최종 OUTPUT은 1이 되고 레지스터 B의 값이 크면 3－비트 크기 비교기의 출력은 0이고 최종 OUTPUT은 0이 된다. 3－비트 크기 비교기로 회로의 동작을 살펴보면, 입력으로(A2 A1 A0)가 (010)이고 (B2 B1 B0)가 (001)이면 PMOS의 컨덕턴스는 2이고 NMOS의 컨덕턴스는 1로서 OUT는 1이 되고, 입력으로(A2 A1 A0)가 (010)이고 (B2 B1 B0)가 (111)이면 PMOS의 컨덕턴스는 2이고 NMOS의 컨덕턴스는 7로서 OUT는 0이 된다. 그림18은 3－비트 크기 비교기이고 그림19은 4－비트 크기 비교기이다.

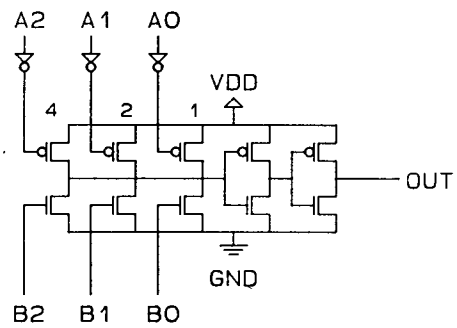


그림 18. 3－비트 크기 비교기
Fig. 18. 3－bit magnitude comparator circuit.

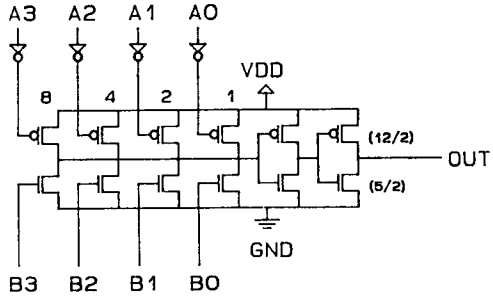


그림 19. 4－비트 크기 비교기
Fig. 19. 4－bit magnitude comparator circuit.

V. 수행결과 및 고찰

영상입력 장치를 통하여 입력된 문자의 인식을 위한 전처리과정으로 기존의 방법은 특징점 추출을 용이하게 하기 위해 문자를 하나의 화소폭으로 나타내는 세선화 과정을 거친 다음 세선화된 문자 영상에서 마스크 매칭 방법등을 이용해 특징점을 추출하였다. 그러나 이러한 방법은 대부분의 수행시간을 전처리 과정인 세선화 과정에서 소모하게 된다. 본 연구에서는 이렇게 시간이 많이 걸리는 기존의 세선화 과정 대신 문자 영상의 중심 화소만을 추적하면서 문자의 선폭을 하나의 화소폭으로 변화시키고 동시에 특징점을 추출하는 트레이싱 알고리듬을 개발하였고 이를 신경 칩화함으로써 수행속도를 개선하였다. 세선화 과정과 트레이싱 과정의 수행시간 비교를 표 1에 나타내었으며 트레이싱 과정을 이용한 전처리 과정이 세선화 과정을 이용한 전처리 과정보다 수행속도가 5배 정도 빠름을 알 수 있었다. 이 방법에서는 잡음제거 및 선형화 과정이 없어도 마스크가 크고, 또한 2픽셀씩 이동하므로 어느정도의 잡음에는 영향을 받지 않는다. 트레이싱 알고리듬을 한글과 한문에 대해서도 적용하여 좋은 결과를 얻을 수 있었다. 그러나 종획과 횡획의 문자 선폭의 차가 큰 특수한 한글과 한문에서 문자부분이 뭉쳐진 부분에 대해서는 좋은 트레이싱 결과를 얻을 수 없었다. 또한 5×5 트레이싱 윈도우의 가중치를 등고선 모양으로 주고 윈도우의 중심에서 가장자리로 멀어질수록 윈도우의 가중치를 현격하게 작게 함으로써 문자 영상의 선폭의 영향을 줄여서, 문자 영상의 선폭이 2~6화소인 모든 문자 영상에 대한 트레이싱을 가능하게 하였다. 이 알고리듬을 이용하여 추출한 특징점과 기존의 세선화 알고리듬으로 추출한 특징점은 거의 같다.

V. 결 론

본 연구에서는 스캐너를 통해 입력된 영문자 영상으로부터 영문자 및 기호의 인식에 필요한 전처리 과정으로 시간이 많이 걸리는 기존의 세선화 과정 대신 수행속도가 5배 정도 빠른 트레이싱 알고리듬을 제안하였다. 트레이싱 알고리듬은 문자 인식을 위한 전처리 과정을 한번에 수행하는 알고리듬으로서 세선화 과정을 포함하는 기존의 전처리 과정보다 수행시간이 빠르다. 본 연구에서는 또한 이 알고리듬을 신경 회로망 모델인 단층 구조 퍼셉트론 알고리듬을 이용하여 신경 칩으로 설계하였다. 회로는 병렬 처리함으로써 보다 빠른 처리 속도를 가진다. 그러나 트레이싱 과정 회로는 5×5 윈도우의 각 방향으로의 이동과, 각 방향에대한 윈도우의 가중치합을 소프트웨어에 의존하고 있다. 앞으로의 연구 과제는 방향 성분을 포함한 윈도우의 가중치합을 저장하는 방법에 대한 것이다.

參 考 文 獻

[1] W.H. Abdulla, A.O.M. Saleh, A.H. Morad, "A preprocessing Algorithm for Hand-written character Recognition," Pattern Recognition Letters: pp. 13-18, vol. 7, no. 1, 1988.

[2] C.Y. Suen, M. Berthod, and S. Mori, "Automatic recognition of Hand-printed Character-the State of the Art" *Proc. IEEE*, vol. 68, no. 4, pp. 469-489, 1980.

[3] N.J. Naccache and r. shingnal, "SPTA: A proposed algorithm for thinning binary patterns", *IEEE Trans. Systems, Men, and Cybernetics, SMC14*, pp. 409, 1984.

[4] G.T. Toussaint, "The use of context in pattern recognition", Pattern Recognition, Bol. 10, pp. 189-204, 1978.

[5] J.J. Hull and s.N. Srihari, "Experiments in Text Recognition with binary n-gram and viterbi algorithms", IEEE Trans. Pattern Anal. Machine Intel, vol. PAMI-4, pp. 520-530, 1982.

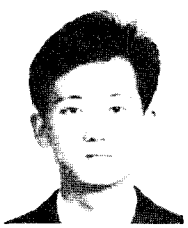
[6] 류종필, "신경 회로망을 이용한 문자인식용 칩 설계", 경북 대학교, 석사 학위 논문, 1989.

[7] 남호원, "영문자 인식 및 전처리용 신경 칩의 설계", 경북대학교, 석사 학위 논문, 1990.

[8] A. Rosenfeld, L.S. Davis, "A note on thinning," *IEEE Trans. on Systems, Men, and Cybernetics:* pp. 226-228, March, 1976.

[9] H. Takakubo and K. Shono, "Recognition of hand-written drawing image using CMOS neural m odules," 제2회 일·한 공동 세미나 논문집, pp. 111-126, 1990, 2.

著 者 紹 介



高 輝 鎭 (正會員)
1966年 10月 5日生. 1989年 2月 경북대학교 전자공학과 졸업. 1991年 2월 경북대학교 대학원 전자공학과 공학석사학위 취득. 1991年 2月～현재 삼성전관(주) 정보연구소. CAD실 연구원. 주관심분야는 영문자인식및 신경회로망 등임.

呂 珍 璟 (正會員) 第28卷 B編 第10號 參照
현재 경북대학교 전자공학과 박사과정수료(1991年)

●

鄭 鎬 宣 (正會員) 第28卷 B編 第9號 參照
현재 경북대학교 전자공학과 교수