

다중 인식기 및 검증기를 갖는 거버문자 인식 시스템

A Gerber-Character Recognition System with Multiple Recognizers and a Verifier

오혜원 · 박태형

Hye-Won Oh and Tae-Hyoung Park

충북대학교 제어계측공학과 및 컴퓨터 정보통신 연구소

요 약

인쇄회로기판 제작에 사용되는 국제표준규격의 거버 파일로부터 부품 위치 이름을 자동으로 추출하기 위한 문자인식 시스템을 제안한다. 거버 파일은 벡터형식의 그림파일로서, 각종도형 및 기호가 문자와 혼합되어 있으며, 가로쓰기와 세로쓰기 및 역 세로쓰기가 병용된다. 거버문자인식 시스템은 거버 파일에서 문자패턴을 추출하여 분리하는 전 처리 단계와 추출된 패턴을 인식하는 인식단계 및 인식된 문자와 숫자를 조합하여 부품위치이름을 구성하는 후 처리단계로 구성된다. 특히 인식을 향상을 위하여 신경회로망에 의한 다중인식기 및 구조적 특징을 이용한 검증기를 개발한다. 본 논문에서 개발된 거버 문자 인식시스템은 인쇄회로기판 조립 및 검사 장비를 위한 자동 프로그래밍 시스템에 사용되어, 전자제품 제조시스템의 생산성 향상에 기여할 수 있다.

Abstract

We propose the character recognition system for Gerber files. The Gerber file is the vector-formatted drawing file for PCB manufacturing, which includes various symbols, figures and characters. Also, the characters are written in horizontal, vertical, and reverse-vertical directions. In this paper, we newly propose the Gerber-character recognition system to recognize all of component names located in PCB. To improve the performance, we develop the multiple recognizers by neural networks and the verifier considering the structural features. The developed system has been installed to the auto-programming software for PCB assembly and inspection machines.

Key Words : Character recognition, Electronic manufacturing system, Gerber file, Neural networks

1. 서 론

인쇄회로기판에 저항, 컨덴서 등의 부품을 조립하는 전자 조립라인은 스크린 프린터, 납 도포 검사기, 칩마운터, 디스펜서, 납땜 검사기 등 다양한 장비로 구성된 생산시스템이다. 최근 휴대폰, 컴퓨터 등 정보통신기기의 수요가 급증함에 따라 전자조립라인의 생산성 향상이 매우 중요하게 인식되고 있다. 특히 다품종 소량생산 체계의 확산에 의하여 생산모델의 교체주기가 단축되고 있으며, 이에 대응하는 작업 준비시간의 단축이 라인 생산성 향상에 크게 영향을 준다.

전자조립라인을 구성하는 기기의 작업 준비를 위해서는, 우선적으로 인쇄회로기판에 존재하는 수백-수천 개의 부품에 대한 위치데이터가 입력되어야 한다. 현재 사용되고 있는 위치데이터를 입력하는 방식은 다음의 네 가지로 구분할 수 있다. 첫째, 장비 내부에 장착된 교시용 카메라를 사용하여, 각 부품 위치로 카메라를 수동으로 일일이 이동시켜 위치데이터를 교시하는 방식이다. 교시에 많은 시간이 소요되며, 특히 장비 가동을 멈추고 작업하여야 하므로 장비 가동률을 크

게 저하시킨다. 둘째, 인쇄회로기판에 대한 스캐너 이미지를 사용하여 오프라인 컴퓨터에서 위치를 입력하는 방식이다. 장비 가동 중에 위치 입력이 가능하나, 화면상에서 수동 입력 작업을 하여야 하므로 시간이 많이 소요되고 위치 정확도 또한 낮다. 셋째, 인쇄회로기판을 설계한 CAD 파일로부터 자동으로 위치데이터를 추출하는 방식이다. 오프라인에서 전자동으로 정확한 데이터 추출이 가능하나, CAD 파일의 규격이 수백 개의 CAD 기종별로 상이하므로 일반적 적용이 어렵다. 또한 설계보안 문제로 인하여 조립라인 또는 외주업체에서 CAD 파일을 확보하는 것이 용이하지 않다. 넷째, 인쇄회로기판 제조를 위한 거버 (Gerber) 파일로부터 부품들의 위치데이터를 추출하는 방식이다. 거버 파일은 국제표준규격이며, 조립라인에서의 확보가 용이하다. 그러나 텍스트 형식의 CAD 파일과 달리 거버 파일은 그림파일이며, 그림파일에서 자동으로 부품 위치 데이터를 추출하기 위해서는 별도의 지능소프트웨어가 필요하다. 이상 네 가지의 부품 위치 데이터 입력방식 중 거버 파일을 사용하는 방법이 점차 확산되고 있다. [12][13]

거버 파일로부터 인쇄회로기판 내의 부품위치데이터를 자동으로 추출하는 소프트웨어를 거버 인터페이스 프로그램이라 한다. 이 프로그램은 작업기판에 대한 부품의 위치좌표와 부품위치이름, 납땜 패드의 위치 좌표, 크기 및 면적 데이터 등을 거버 파일에서 자동으로 추출한다. 이때 'R1', 'C2', 'U3' 등의 부품위치이름을 자동으로 추출하기 위해서는 별도의 문자인식기능이 필요하다. 본 논문은 거버 인터페이스

접수일자 : 2003년 10월 15일

완료일자 : 2004년 1월 14일

감사의 글 : 본 연구는 한국산업기술재단 2003년 대학보유기술이전사업 제품화 연구과제의 지원으로 수행되었습니다.

프로그램을 위한 문자인식시스템을 새로이 제안한다.

문자인식에 대한 연구는 대부분은 이미지 스캐너, 카메라 등 광학입력장치를 통하여 입력되는 데이터를 분석하여 문자를 인식하는 광학문자인식 (optical character recognition, OCR) 에 집중되어 왔다.[1 4][14] 최근에는 우편물 자동 분류, 차량 번호판 인식, 금융기관에서의 전표 또는 수표의 자동입력 등 응용 분야를 넓혀가고 있다. 특히 최근 디지털 카메라 등 광학 입력 장치가 보편화되고, 개인 휴대용 단말기 (PDA) 의 보급으로 많은 연구와 상품화가 진행 중이다. 거버문자의 인식에 관하여 발표된 연구는 단일 신경회로망을 거버문자인식에 적용한 연구[15]가 유일하다. 본 논문은 거버문자의 인식을 향상을 위하여, 다중 신경회로망 및 검증기를 적용한 결과를 제시한다.

반면 거버 파일에서 부품위치이름을 추출하는 거버 문자 인식의 경우, 광학문자인식과 여러 가지 점에서 차별화된다. 우선 입력패턴이 비트맵형식이 아닌 벡터형식으로 주어지며, 문자뿐 아니라 각종 도형 및 기호가 포함되어 있다. 또한 부품위치이름은 가로쓰기, 세로쓰기 및 역 세로쓰기 등 다양한 방향으로 표기되어 있으므로 각도를 고려하여 문자를 인식해야 한다. 즉 기존의 광학문자인식을 위한 방법을 거버문자인식에 직접 적용하기 어려우며, 전 처리, 특징추출, 분류 및 후 처리 등 모든 문자인식단계를 새로이 개발하여야 한다.

본 논문은 거버문자 인식을 위한 하나의 시스템을 제안하고자 한다. 거버 파일을 분석하여 이로부터 문자패턴을 분리하고, 여러 가지 특징점을 추출한다. 문자패턴의 효과적 분류를 위하여 다중 인식기와 검증기를 사용한다. 이때 다중 인식기는 특징벡터를 입력으로 하는 신경회로망으로 구성되며, 검증기는 문자의 구조적 특징 값을 사용하여 오 인식을 최소화 시킨다. 분류된 문자들에 대한 후처리 과정을 통하여 부품위치이름을 자동으로 인식한다. 다양한 거버 파일에 대한 실험을 통하여 제안된 방법의 유용성을 검증한다.

2. 거버 문자 인식 시스템

거버 파일은 PCB 의 제작을 위한 파일로서, 포토 플로터, 드릴링 머신 등 PCB 제조기의 구동을 위하여 사용된다. 회로설계용 CAD 파일로부터 생성되며, 하나의 기관에 대하여 부품 면, 납땜 면, 마스크 면, 실크 면 등 여러 개의 거버 파일이 필요하다. 거버 파일은 1980년 Gerber System 사에 의하여 처음 제안되었으며, 이후 국제표준규격 (RS-274D, RS-274X) [5] 으로 채택되어 널리 사용되고 있다.

거버 파일의 데이터는 툴의 형상, 크기 및 이동방식을 정의한 D 코드, 위치 보간 방식을 정의한 G 코드 및 XY 위치 값으로 구성된다. 그림 1은 문자 'A'에 대한 거버 파일 데이터의 일부를 보여준다. 예에서 보는 바와 같이 거버 파일은 벡터형식으로 정의된 일종의 그림파일이다. 툴의 형상 및 크기는 aperture 데이터로서 별도로 정의되나, 본 문자인식에서는 별도의 세션화 과정을 생략하기 위하여 이를 사용하지 않는다.

거버 인터페이스 프로그램은 칩마운터, 납땜검사기, 디스펜서 등 PCB 조립 및 검사장비에 필요한 부품위치 데이터를 거버 파일에서 추출하는 소프트웨어이다. 부품위치 데이터는 'R1', 'C2' 등의 부품위치이름과 위치좌표 및 각도를 포함하며, 이를 오프라인에서 정확하게 추출할 수 있으므로 생산현장에서 널리 사용된다. 거버 인터페이스 프로그램 사용 시, 위치좌표 및 각도는 자동으로 추출가능 하나, 부품위치이름의 경우

G01*	직선 보간 운동
G54D10*	10 번 툴로 교체
X1000Y1000D01*	(1000,1000) 위치로 이동
X3000Y5000D02*	(3000,5000) 위치까지 선 그리기
X5000Y1000D02*	(5000,1000) 위치까지 선 그리기
X2000Y3000D01*	(4000,3000) 위치로 이동
X4000Y3000D02*	(2000,3000) 위치까지 선 그리기

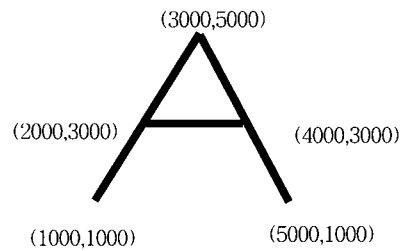


그림 1. 거버 데이터의 예.

Fig. 1. An example of Gerber data.

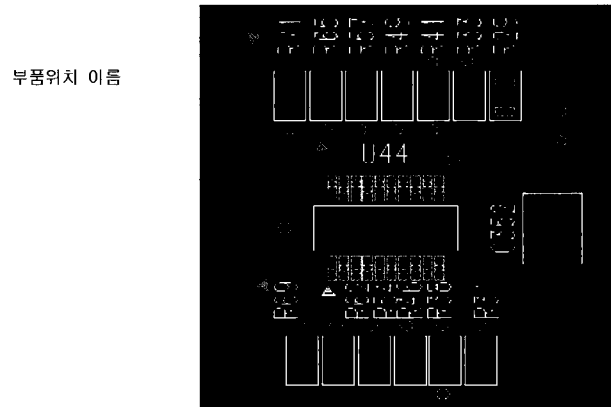


그림 2. 거버 파일의 그림.

Fig. 2. Drawing of a Gerber file.

사용자가 일일이 수동으로 입력하여야 한다. 하나의 기관에는 수백-수천 개의 부품이 존재하므로, 상당한 시간이 소요되며 사용자가 잘못 입력하는 경우도 발생할 수 있다. 거버 파일 내에 존재하는 부품위치이름을 자동으로 추출할 수 있으면, 부품위치데이터 추출에 소요되는 시간을 대폭 단축할 수 있으며 또한 오 입력에 의한 생산 손실도 방지할 수 있다.

그림 2는 거버 파일을 읽어서 화면에 표시한 예이다. 여러 가지 도형, 기호와 함께 섞여있는 부품위치이름을 볼 수 있으며, 가로 및 세로, 역 세로 방향으로 쓰여 진다. 부품위치이름은 A-Z 의 영문 대문자와 0-9의 숫자의 조합으로 구성된다. 거버 파일에 사용되는 문자 데이터를 거버 문자라 정의하며, 그림 3은 거버 문자의 예를 보여준다. 거버 문자는 다양한 폰트와 크기를 갖으며, 0도, 90도 및 -90도 방향을 갖는다.

거버 문자 인식 시스템은 거버문자의 자동인식을 통한 부품위치 이름의 자동추출을 목표로 한다. 그림 4는 본 논문에서

서 제안하는 거버문자 인식시스템이다. 전체 시스템은 크게 전 처리 단계, 특징 추출 단계, 인식 단계 및 후 처리 단계로 구분할 수 있다. 전 처리 단계는 전체 기판에 대한 거버 파일을 읽어드리고, 기호와 도형을 제외한 문자와 숫자를 분리(segmentation) 하여 패턴을 만드는 과정이다. 특징 추출 단계에서는 분리된 각 문자 패턴에 대한 화소 값 특징, 방향성분 특징 및 교차점 특징을 추출하는 과정이다. 인식 단계에서는 특징 값들을 사용하여 주어진 패턴을 A-Z, 0-9 중 어느 하나로 분류하는 과정이다. 신경회로망을 사용한 3개의 인식기를 사용하며, 문자의 구조적 특성을 고려한 검증기를 별도로 사용한다. 후처리 단계에서는 분류된 문자 및 숫자를 조합하여 부품위치이름을 추출하는 과정이다.

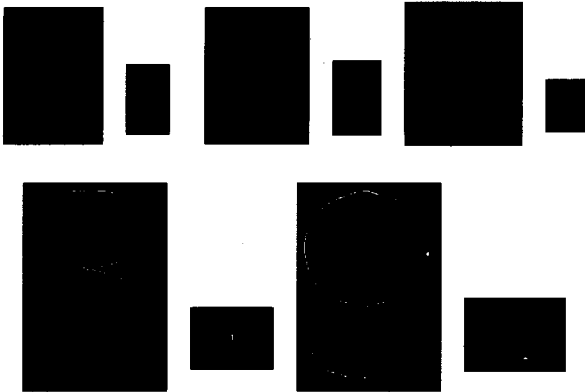


그림 3. 거버 문자 패턴의 예.
Fig. 3. Patterns of Gerber characters.

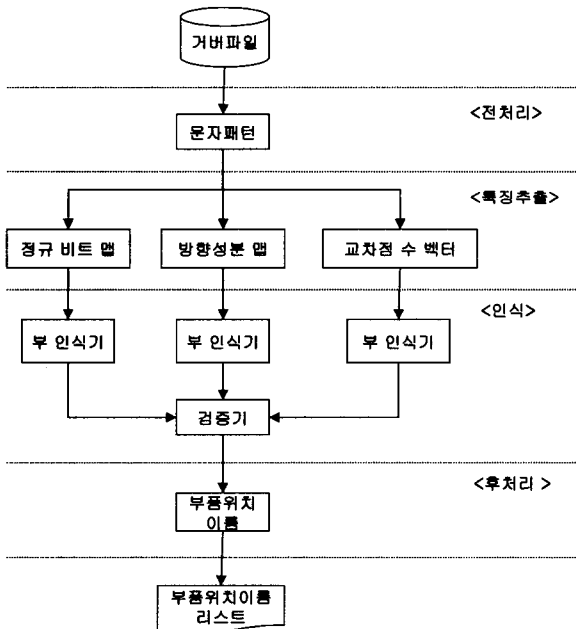


그림 4. 거버 문자 인식 시스템.
Fig. 4. Gerber-character recognition system.

3. 문자인식 알고리즘

3.1 전 처리(Pre-Processing)

벡터형식의 그림파일인 거버 파일에서 문자, 숫자, 기호,

도형을 분리하여 비트맵 패턴을 생성하는 단계이다. 즉 그림 2와 같은 거버데이터에서 그림 3 과 같은 부품문자이름의 인식을 위한 문자 패턴 $P \subset \{0,1\}^{M \times N}$ 을 생성한다. 이때, 문자의 크기에 따라 패턴의 크기가 달라지며, 문자의 두께는 모두 1화소를 차지한다. 일반적인 OCR의 경우와 달리, 별도의 이진화, 필터링 및 세션화 과정이 불필요하다. 전 처리 알고리즘은 다음과 같다.

- S1. D01 코드 (이동)로부터 시작하여 다음의 D01 코드가 나오기 전까지의 D02 코드 (선 굵기)로 구성된 벡터 집합 $\{v_{x+1}, v_{x+2}, \dots, v_{x+k}\}$ 을 하나의 세그먼트 s_i 로 구성한다.
- S2. 각 세그먼트 $s_i (i=1, \dots, n)$ 에 대하여 영역의 폭 W_i 및 길이 L_i 를 구하고, $W_i \cap W_j \neq \emptyset$ 또는 $L_i \cap L_j \neq \emptyset$ 인 다음의 두 세그먼트를 합쳐서,

$$s_i = v_{x+1}, v_{x+2}, \dots, v_{x+k}$$

$$s_j = v_{y+1}, v_{y+2}, \dots, v_{y+l}$$

다음과 같은 하나의 세그먼트를 만든다.

$$s_i = v_{x+1}, v_{x+2}, \dots, v_{x+k}, v_{y+1}, v_{y+2}, \dots, v_{y+l}$$

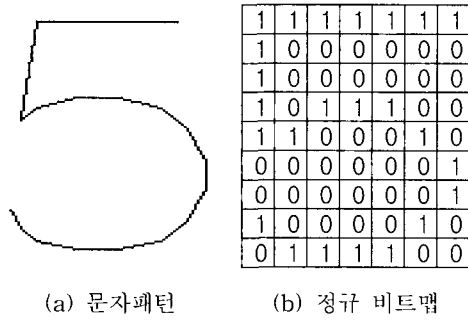
- S3. 각 세그먼트 $s_i (i=1, \dots, n)$ 에 대한 W_i 및 L_i 가 미리 정의된 최대 값 이상이거나 최소 값 이하이면, 해당 세그먼트를 제거한다.
- S4. 모두 동일한 벡터로 구성된 세그먼트를 모아서 하나의 패턴으로 등록한다.
- S5. 벡터로 구성된 패턴을 비트맵으로 변환시킨다.

3.2 특징 추출 (Feature Extraction)

인식단계에 필요한 문자패턴의 특징 값을 추출한다. 특징 값은 각 문자패턴이 유일하게 갖는 값으로 설정하는 것이 바람직하다. 또한 거버문자가 갖는 다양한 폰트 및 크기, 문자의 회전을 고려하여야 하며, 특히 기호, 도형 등 비 문자와의 구분에도 유리하여야 한다. 일반적으로 하나의 특징 값을 사용하는 것보다는 여러 가지 특징 값을 사용하는 것이 인식을 향상에 유리하다. 본 논문에서는 정규 비트맵, 교차점 수 벡터 및 방향성분 맵의 세 가지 특징 값을 사용한다. 일반적 문자인식과 달리, 수직 및 역 수직 방향의 문자를 함께 인식하여야 하고 또한 기호 및 도형도 처리하여야 하므로, 실험을 통하여 인식에 유리한 특징 값을 선택하였다.

정규 비트맵 $F_B \subset \{0,1\}^{m \times n}$ 은 다양한 크기를 갖는 문자패턴을 $m \times n$ 의 일정한 크기로 정규화 시킨 2차원 벡터이다. 맵의 각 원소는 0 또는 1의 값을 갖는다. 거버 문자의 크기가 다양하므로, 문자의 인식을 위하여 일정한 크기로 정규화 시킬 필요가 있다. 맵의 크기가 큰 경우 다양한 패턴정보를 표현할 수 있으나, 인식단계에서 많은 시간이 소요될 수 있으므로 그 크기를 적절히 결정하여야 한다. 그림 5는 숫자 '5'의 문자패턴에 대한 9×7 의 정규 비트맵이다.

방향성분 맵 $F_D^i \subset \{0,1\}^{p \times q} (i=1,2,3,4)$ 은 문자패턴에 포함된 방향성분을 추출하여 표현한 특징 값이다. 그림 6(a)는 수평, 수직, 사선, 역사선 성분에 대한 네 가지의 방향 코드를 나타낸다. 패턴의 좌상단점에서 출발하여 화소 값이 1 인 모든 화소에 대한 방향코드를 구한다. 네 가지 방향코드 별로 0 또는 1의 값을 갖는 맵을 구성하며, 이 때 맵의 크기를

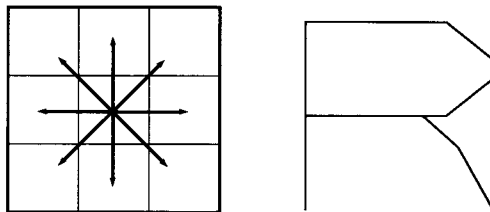


(a) 문자패턴 (b) 정규 비트맵

그림 5. 정규 비트맵 ($m=9, n=7$).

Fig. 5. Normalized bitmap.

$p \times q$ 로 정규화 시킨다. 결과적으로 문자패턴에 대한 수평, 수직, 사선, 역 사선의 네 가지 방향성분 맵이 추출된다. 그림 6(b)는 'R' 문자패턴을 보여주고, 그림 6(c)-(f)는 'R' 문자패턴에 대하여 추출된 네 가지 방향성분 맵을 보여준다.



(a) 방향 코드 (b) 문자패턴

1	1	1	0	0
0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(c) 수평성분 맵

0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

(e) 사선성분 맵

1	0	0	0	0
1	0	0	0	1
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0

(d) 수직성분 맵

0	0	0	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	1	0
0	0	0	0	1
0	0	0	0	0

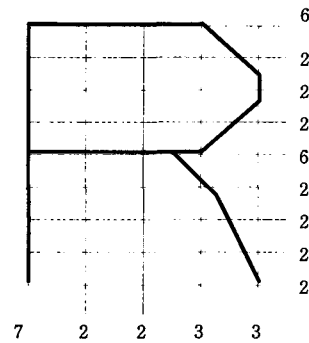
(f) 역 사선성분 맵

그림 6. 방향성분 맵 ($p=7, q=5$).

Fig. 6. Direction map.

교차점 수 벡터 $F_C \subset I^{h \times v}$ (단, $I = \{1, 2, \dots, k\}$)는 원 문자패턴에 수평으로 h 개, 수직으로 v 개의 주사선을 투영시켜서, 각 주사선과 문자패턴이 만나는 교차점의 수를 기준으로 생성시킨 벡터이다. 이 때 문자패턴의 크기가 다양하여, 교차점의 수에 편차가 많이 발생할 수 있으므로, 이를 k 단계로 정규화 시킨다. 그림 7은 수평으로 9개, 수직으로 5개의

주사선을 사용하고, 교차점의 수를 7단계로 정규화 하여 얻은 교차점 수 벡터의 예를 보여준다.



(a) 문자패턴

6	2	2	2	6	2	2	2	2	7	2	2	3	3
---	---	---	---	---	---	---	---	---	---	---	---	---	---

(b) 교차점 수 벡터

그림 7. 교차점 수 벡터 ($h=9, v=5, k=7$).

Fig. 7. Cross count vector.

3.3 인식 (Recognition)

인식 단계에서는 추출된 특징 값들로부터 입력된 문자패턴을 A-Z의 영문 대문자와 0-9의 숫자로 분류하며, 이에 속하지 않는 경우 기호로 판정한다. 또한 0도, 90도 및 -90도의 문자방향을 결정한다. 본 논문은 다중인식기와 검증기를 사용하여 인식단계를 구현한다.

3.3.1 다중 인식기

문자인식에 있어서 패턴의 특징 값으로부터 문자를 분류하는 인식기는, 미리 등록된 표준값과 비교하는 방법 [6][16]과 신경회로망을 사용하는 방법 [7] 등으로 구분할 수 있다. 거버 문자 인식의 경우, 다양한 폰트 및 크기에 대응하여야 하므로 문자별 표준값 생성이 어려우며, 따라서 신경회로망을 사용하는 것이 보다 적합하다. 신경회로망을 인식기로서 사용하는 경우, 단일 인식기에 의한 방법 [8]과 다중 인식기에 의한 방법 [9][17][18]으로 구분할 수 있다. 여러 개의 특징 값을 입력으로 하는 단일 인식기의 경우, 신경회로망으로 구성하는 연결강도의 수가 증가하여 장시간의 학습이 요구된다. 그러나 각 특징 값을 입력으로 하는 여러 개의 부 인식기들로 구성된 다중 인식기는, 각 부 인식기 별로 학습이 이루어지므로 학습시간이 단축될 수 있다. 또한 부 인식기 결과를 종합하여 검증기를 거쳐 최종 판정하므로, 각 부 인식기의 약점을 상호 보완할 수 있는 장점이 있다.

그림 8은 본 논문에서 제안하는 거버 문자 인식을 위한 다중 인식기의 구조이다. 정규 비트맵, 교차점 수 벡터 및 방향성분 맵을 각 입력으로 하는 세 개의 부 인식기로 구성된다. 각 부 인식기는 입력 층, 은닉 층 및 출력 층으로 구성된 다층 신경회로망이다. 다중 인식기의 학습은 표준 문자패턴을 사용하여 각 부 인식기 별로 독자적으로 수행된다. 학습 알고리즘은 다층 신경회로망의 학습에 널리 사용되는 오류 역전파 알고리즘 [7]을 사용한다.

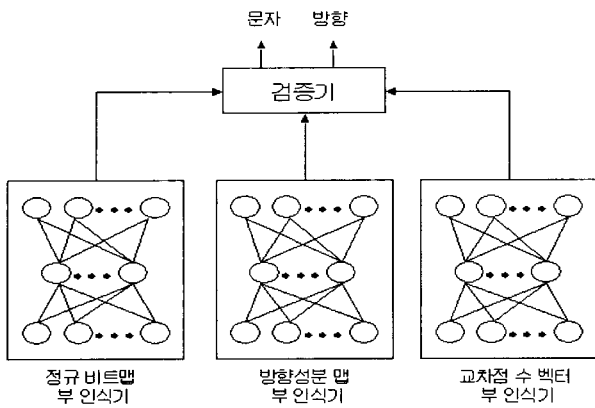
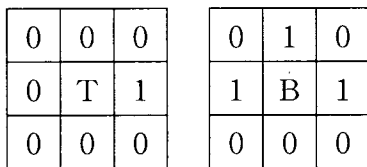


그림 8. 다중 인식기 및 검증기.
Fig. 8. Multiple recognizers and verifier.

정규 비트맵 부 인식기의 입력 층 각 노드에는 정규 비트맵 F_B 의 원소인 $b_{xy} \in \{0, 1\}$ ($1 \leq x \leq m, 1 \leq y \leq n$)가 입력되며, 총 $m \times n$ 개의 노드를 갖는다. 방향성분 맵 부 인식기의 입력 층 각 노드에는 방향성분 맵 F_D^i ($i=1, 2, 3, 4$)의 원소인 $d_{xy} \in \{0, 1\}$ ($1 \leq x \leq p, 1 \leq y \leq q$)가 입력되며, 총 $4 \times p \times q$ 개의 노드를 갖는다. 교차점 수 벡터 부 인식기의 입력 층 각 노드에는 교차점 수 벡터 F_C 의 원소인 $c_x \in \{1, \dots, k\}$ ($1 \leq x \leq h+v$)가 각각 r 비트의 이진수로 변환되어 입력되며, 입력 층은 총 $(h+v) \cdot r$ 개의 노드를 갖는다. 모든 부 인식기의 출력 층 A-Z 및 0-9를 판정하는 36 개의 노드와, 기호임을 판정하는 1 개의 노드, 문자의 방향 (0, 90, -90) 을 판정하는 3 개의 노드를 포함하여 총 40 개의 노드로서 동일하게 구성된다.

3.3.2 검증기

검증기는 학습이 완료된 부 인식기의 출력 결과를 입력으로 하여, 최종적으로 입력패턴의 문자 및 각도를 판정한다. 세 가지 부 인식기의 출력이 모두 일치하는 경우에는, 그 값으로 판정한다. 그러나 이 중 상호 불일치하는 결과가 존재하는 경우 별도의 검증 알고리즘을 통하여 최종 판정한다. 검증 알고리즘의 수행을 위하여, 앞서 정의된 세 가지 특징 값 이외에 끝점 수, 분기점 수, 중단 및 횡단 획의 수, 홀(hole)의 유무 등 여러 가지 특징 값을 추가로 정의한다.



(a) 끝점 T (b) 분기점 B

그림 9. 끝점 및 분기점의 예.

Fig. 9. Examples of the terminal and branch point.

끝점은 이웃 값 중 1이 하나 만 존재하는 점이며, 분기점은 이웃 값 중 1이 수평 또는 수직 방향으로 3개 존재하는

점이다. 그림 9는 각 끝점 및 분기점의 예를 보여준다. 3×3 의 마스크를 사용하여, 문자의 골격선을 따라가며 구할 수 있다. 은 분기중단 및 횡단 획은 문자패턴 전체를 중단 또는 횡단하는 직선의 수를 의미한다. 앞서 정의된 교차점 수 벡터에서 개수와 크기 및 위치를 판별할 수 있다. 홀은 문자패턴에 존재하는 폐곡선을 의미한다.

혼동 문자 쌍 검증 알고리즘은 다음과 같다.

- S1. 정규 비트맵 부 인식기의 출력, 방향 성분 맵 부 인식기의 출력, 교차점 수 벡터 부 인식기의 출력이 모두 일치하는 경우, 부 인식기의 결과를 출력한다. 두 개의 부 인식기 결과가 동일하고, 다른 하나의 부 인식기 결과가 다른 경우 S2로 간다. 세 개의 부 인식기 결과가 모두 다른 경우 S3로 간다.
- S2. (혼동 문자 쌍 검증)
 - 두 문자가 혼동 문자 쌍 리스트에 없으면 S3 로 간다.
 - 있으면, 혼동 문자 쌍 리스트에 존재하는 검증 방법에 의거, 각 문자패턴에 대하여 해당 특징 값을 구하고 이를 표준 값과 비교 판정하여, 그 결과를 출력한다.
- S3. (가중치 검증)
 - S3.1. 문자패턴에 홀이 존재하는 경우, 방향성분 맵 부 인식기의 모든 출력노드 값에 1 보다 큰 가중치를 곱한다.
 - S3.2. 문자패턴에 중단 및 횡단 선이 존재하는 경우, 교차점 수 벡터 부 인식기의 모든 출력노드 값에 1 보다 큰 가중치를 곱한다.
 - S3.3. 세 개의 부 인식기에 대한 모든 출력노드 중 그 값이 가장 큰 노드의 문자 및 각도로 판정한다.

각 부 인식기에서 서로 다른 값이 자주 발생하는 문자 쌍을 모아서 미리 정의한 것이 혼동 문자 쌍 리스트이다. 표 1은 혼동 문자 쌍 리스트의 예를 보여준다. 예를 들어 문자 'O'와 'D'는 혼동이 발생되기 쉬우며, 이 때 중단 및 횡단획수를 구하여 그 값이 0이면 'O'로, 1이면 'D'로 판정한다. 많은 실험을 통하여 혼동 문자 쌍 리스트를 작성한다. 특히 (6, 9), (J, 7), (-, 1) 등 회전 시 혼동을 일으킬 수 있는 문자 쌍을 포함시킨다.

표 1. 혼동 문자 쌍 리스트의 예.
Table 1. The couples of confusing characters.

혼동 문자 쌍	끝점	분기점	중단 및 횡단 획 수
(O, D)			(0,1)
(0, D)			(0,1)
(8, B)			(0,1)
(N, S)		(2,0)	(2,0)
(E, F)			(3,2)
(L, F)	(2,3)	(0,1)	
(C, G)		(0,1)	

가중치 검증은 혼동 문자 쌍 검증이 불가능한 경우 수행한다. 이것은 세 개의 부 인식기 결과에 가중치를 부여하여 최종 판정하는 방법이다. 문자패턴에 홀이 존재하는 경우 (0, 6, 8, 9, A, B, D, O, P, Q, R)에는 방향성분 맵 부 인식기의 출력을 보다 신뢰할 수 있으므로, 이에 가중치를 부과한다. 중단 및 횡단 선이 존재하는 경우 (1, 2, 4, 5, 7, E, F, H, I,

K, L, M, N, T, Z)에는 교차점 수 벡터 부 인식기의 출력 값을 보다 신뢰할 수 있으므로, 이에 가중치를 부여한다. 혼동 문자 쌍 검증 및 가중치 검증은, 문자의 구조적 특성을 활용하여 인식률을 높이는 경험적 방법이다.

3.4 후 처리 (Post-Processing)

인식단계에서 출력된 각 패턴의 문자 및 각도 값으로부터, 부품위치이름을 생성하는 단계가 후 처리이다. 거버 파일에 존재하는 모든 문자가 부품위치이름에 사용되는 것이 아니므로, 불필요한 문자열은 제외되어야 한다. 부품위치이름은 'R124', 'RA23', 'U67' 등과 같이 반드시 영문 대문자와 숫자의 조합으로 구성된다. 이 특성을 사용하여 부품위치이름을 생성할 수 있다.

전 처리과정에서 생성된 모든 세그먼트에 대하여, 인식단계의 결과로부터 문자 및 각도 값을 부여한다. 임의의 영문자를 갖는 세그먼트로부터 시작하여, 각도에 따른 최 근접 세그먼트를 탐색한다. 0도 문자의 경우 오른쪽, 90도 문자의 경우 위쪽, 90도 문자의 경우 아래쪽이 탐색방향이 된다. 각도가 동일한 문자가 발견된 경우, 앞 문자와의 간격이 기준치 이내이면 부품위치이름에 이 문자를 추가한다. 더 이상의 문자가 발견되지 않을 때까지 문자를 추가하여 부품위치이름을 생성한다.

4. 실험 및 고찰

본 논문에서 제안한 거버 문자 인식시스템을 구현하고 성능평가할 위한 실험을 수행하였다. IBM PC 호환기종 Pentium-IV 급 (2GHz) 컴퓨터의 MS-Windows XP 환경에서 Visual C++ 로 프로그래밍 하였다. 정규 비트맵의 크기 ($m \times n$)는 $9 \times 7 = 63$ 비트로 하였으며, 네 개의 방향성분 맵의 크기 ($p \times q$)는 각 $7 \times 5 = 35$ 비트로 설정하였다. 교차점 수 벡터의 크기 ($h + v$)는 $9 + 5 = 14$ 자리로 하였으며, 단계 수 (k)는 7로 설정하였다. 정규 비트맵 부 인식기는 63개의 입력노드, 63개의 은닉노드, 40개의 출력노드로 구성하였다. 방향성분 맵 부 인식기는 140개의 입력노드, 70개의 은닉노드, 40개의 출력노드로 구성하였다. 마지막으로 교차점 수 벡터 부 인식기는 42개의 입력노드, 42개의 은닉노드, 40개의 출력노드로 구성하였다.

생산현장에서 실제 사용되는 거버 파일을 사용하여 실험을 수행하였다. 기호 및 회전된 문자를 포함하여 총 841 개의 표준 문자패턴을 사용하여 부 인식기를 학습하였다. 사용된 문자패턴은 CAM350 [10], CADSTAR [11] 등 다양한 CAD 소프트웨어에서 여러 가지 라이브러리를 사용하여 생성되었다. 학습에 소요된 시간은 정규 비트맵 부 인식기 16분, 방향성분 맵 부 인식기 31분, 교차점 수 벡터 부 인식기 13분이다. 학습이 완료된 후, 입력 문자에 대한 인식 시간은 문자 당 평균 약 0.15초이다.

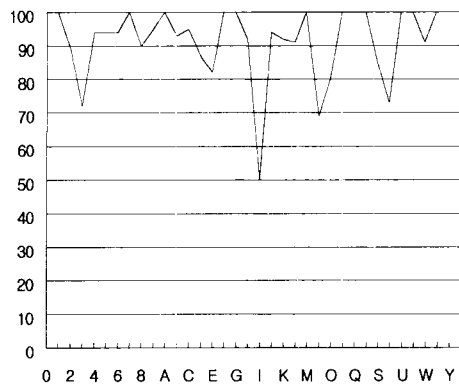
표 2는 본 논문에서 제안한 인식 방법의 결과를 비교하여 보여준다. 방법 1,2,3은 단일 인식기를 사용한 경우의 결과이며, 방법 4는 다중 인식기와 가중치 검증기를 사용한 경우이다. 방법 5는 다중 인식기와 혼동 문자 쌍 및 가중치 검증기를 모두 사용한 경우이다. 단일 인식기를 사용하는 경우보다는 다중 인식기를 사용하는 것이 유리하며, 특히 혼동 문자 쌍 검증기를 사용하여 인식률이 많이 향상되었음을 확인할 수 있다.

그림 10은 표 2의 방법 1, 방법 2, 방법 3, 방법 5에 대하여, 0-9 및 A-Z의 문자별 인식률을 보여준다. 방법 1 및 2의 경우, 문자 'I'에 대한 인식률이 낮으며, 이는 숫자 '1'과 혼동되는 경우가 많이 발생되기 때문이다. 방법 3의 경우 문자 'D'의 인식률이 특히 낮으며, 이는 숫자 '0'과 혼동되는 경우가 많이 발생하는 경우이다. 방법 5와 같이 검증기를 추가한 경우, 문자 별 인식률이 전반적으로 향상됨을 확인할 수 있다. 그림 11은 0도 및 90도, 90도의 문자 각도 별 인식률을 종합하여 보여준다. 전체적으로 0도에 대한 인식률이 높으며, 단일 인식기를 사용하는 경우 그 편차가 큼을 볼 수 있다. 그러나 다중 인식기 및 검증기를 사용하는 경우 90도, -90도에 대한 인식률의 향상으로 편차가 상대적으로 작아짐을 확인할 수 있다.

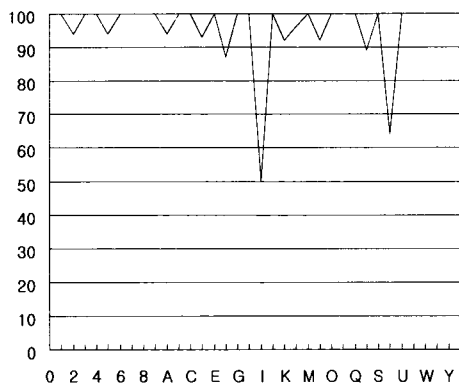
5. 결 론

본 논문은 전자조립 및 검사장비의 오프라인 프로그래밍을 지원하는 거버 인터페이스 프로그램에 사용되는 문자인식 시스템을 새로이 제안하였다. 광학문자와는 다른 거버문자의 특성을 고려하여 전 처리, 특징 추출, 인식, 후처리를 포함한 문자인식 방법을 제시하였다. 특히 다중 인식기 및 검증기를 사용한 인식방법은, 단일 인식기를 사용하는 경우에 비하여 2-3%의 인식률 향상 효과가 있었으며, 결과적으로 96.8%의 인식률 결과를 얻을 수 있었다. 적용 시스템의 특성 상, 본 연구의 인식률 결과는 직접 상용화 프로그램에 적용될 수 있는 수준이다.

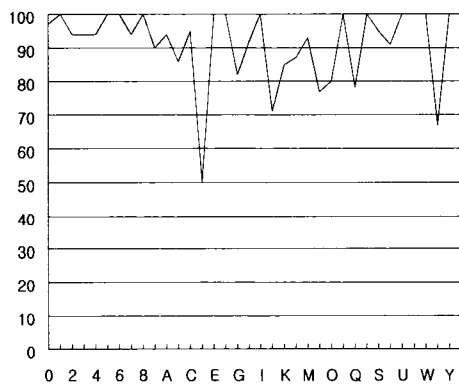
현재의 인식률 수준을 향상시키기 위해서는 검증기 기능을 보다 강화시킬 필요가 있으며, 부품위치이름의 문맥정보 등을 활용하는 방법이 적용될 수 있다. 90도 및 90도 회전된 문자에 대한 인식률이 상대적으로 낮으며, 회전 특성을 잘 반영할 수 있는 특징 값들을 추가하여 인식성능을 향상시킬 예정이다. 본 논문의 결과는 거버 인터페이스 프로그램에 직접 적용될 수 있으며, 전자조립 및 검사 장비의 프로그래밍 시간을 단축하여, 궁극적으로 전자제품 제조라인의 생산성 향상에 기여할 수 있으리라 기대된다. 또한 문자인식시스템의 응용 영역을 새로이 확대하였다는 점에서 본 연구의 의의를 찾을 수 있다.



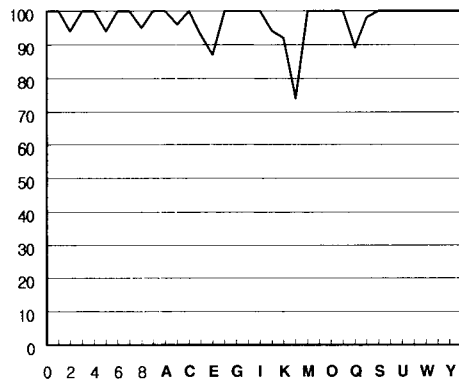
(a) 방법 1



(b) 방법 2

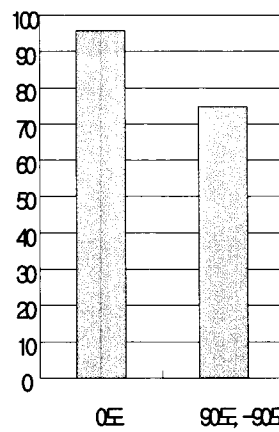


(c) 방법 3

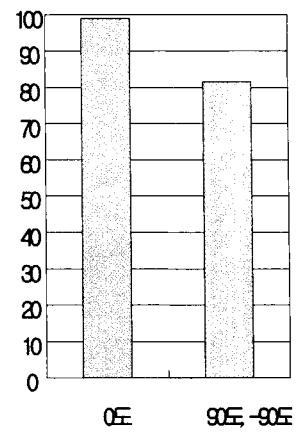


(d) 방법 5

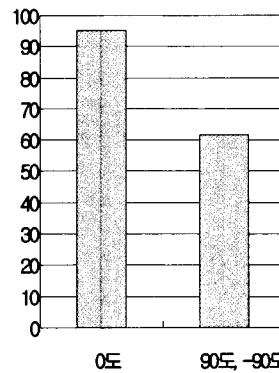
그림 10. 인식방법 별 인식률 (문자 기준).
Fig. 10. Recognition rates for characters.



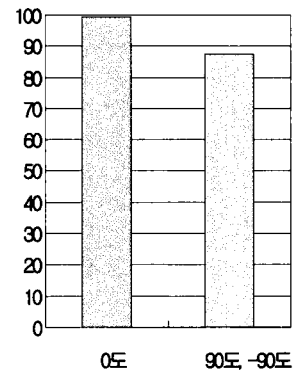
(a) 방법 1



(b) 방법 2



(c) 방법 3



(d) 방법 5

그림 11. 인식방법 별 인식률 (각도 기준).
Fig. 11. Recognition rates for rotations.

표 2. 거버 문자 인식 결과 비교.

Table 2. Recognition results.

방법	인식기	검증기	인식률 (%)
1	단일 (정규 비트맵)	없음	93.3
2	단일 (방향성분 맵)	없음	95.2
3	단일 (교차점 수 벡터)	없음	93.0
4	다중 (정규 비트맵 + 방향성분 맵 + 교차점 수 벡터)	가중치	95.9
5	다중 (정규 비트맵 + 방향성분 맵 + 교차점 수 벡터)	혼동 문자 쌍 + 가중치	96.8

참고문헌

[1] S. Kahan, T. Pavlidis, H. S. Baird, "On the Recognition of Printed Characters of Any Font and Size," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 2, pp. 274-288, 1978.

[2] L. O's Gorman, R. Kasturi, Executive Briefing Document Image Analysis, IEEE Computer Society, 1997.

[3] S. Mori, H. Nishida, H. Yamada, Optical Character Recognition, John Wiley & Sons, Inc. 1999.

[4] H. Bunke, P. S. P. Wang, Character Recognition and Document Image Analysis, World Scientific, 2000.

[5] The Gerber System Corporation, Gerber RS-274X Format Guide, 1997.

[6] L. Lam, C. Y. Suen, "Structural Classification and Relaxation Matching of Totally Unconstrained Handwritten ZIP-Code Numbers," Pattern Recognition, Vol. 21, No. 1, pp. 19-31, 1998.

[7] S. Haykin, Neural Networks, Prentice Hall, 1999.

[8] H. I. Avi-Itzhak, T. A. Diep, H. Garland, "High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2, pp. 218-224, 1995.

[9] L. S. Oliveira, R. Sabourin, F. Bortolozzi, C. Y. Suen, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1438-1454, 2002.

[10] CAM350 Technical Manual, <http://www.downstreamtech.com>, Down Stream Technology Co., 2003.

[11] CADSTAR Express Manual, <http://www.dmss.co.kr>, (주)대명시스템즈, 2003.

[12] 박태형, "전자조립용 CAM 시스템의 개발 동향," 전자 공학회지, 제 36권, 3호, pp. 272-280, 1999.

[13] 박태형, 송중석, 김철한, "PCB 조립장비용 장착점 추출 프로그램 개발," 제어계측·자동화·로보틱스 연구회 합동 학술 발표회 논문집, pp. 242-247, 2000.

[14] 이성환, 문자인식, 이론과 실제 : 제 1권, 제 2권, 홍릉 과학 출판사, 1994.

[15] 김철한, 박태형, "PCB조립 장비를 위한 거버 문자 인식 알고리즘 개발," 2000 한국자동제어 학술회의 논문집, 2000.

[16] 강현모, "다양한 활자체 한글 영상에서의 한글 문자 인식 알고리즘 연구," 서울대학교 공과대학 공학 석사 논문, 1997.

[17] 박창순, 김두영, "오프라인 필기체 숫자 인식을 위한 다양한 특징들의 성능 비교 및 인식률 개선 방안," 한국정보처리학회 논문지, 제 3권, 4호, pp. 915-925, 1996.

[18] 김원우, 백중현, 이관용, 변혜란, 이일병 "구조적 특징 분석기를 이용한 무제약 필기 숫자 인식기의 결합," 한국인지과학회 논문지, 제 7권, 1호, pp. 37-56, 1996.

저 자 소개



오혜원 (Hye-Won Oh)

2001년 : 충북대 제어계측공학과 졸업 (공학사)

2001년~현재 : 충북대 제어계측공학과 석사과정.

관심분야 : 영상처리, 문자인식, 신경회로망
E-mail : jamtongi@hotmail.com



박태형 (Tae-Hyoung Park)

1988년 : 서울대 제어계측공학과 졸업 (공학사)

1990년 : 서울대 제어계측공학과 졸업 (공학석사)

1994년 : 서울대 제어계측공학과 졸업 (공학박사)

1994년~1997년 : 삼성테크윈(주) 정밀기기 연구소 선임연구원

2000년~2001년 : University of Toronto 방문교수
1997년~현재 : 충북대학교 제어계측공학과 교수

관심분야 : 전자 조립 시스템, 로봇 동작계획, 자동 프로그래밍 시스템, 최적화 알고리즘

Phone : 043-261-3240

Fax : 043-268-2386

E-mail : taehpark@chungbuk.ac.kr