

OrCAD 정합을 위한 수작업 도면 인식

The Hand-Drawn Diagram Recognition for OrCAD Matching

박 영 식, 김 진 홍

(Young-Sik Park and Jin-Hong Kim)

Abstract : CAD diagrams generally consists of many basic components: symbols, character, and connection lines. Thus, to recognize the diagrams, it is necessary to extract each components, and understand their meanings and relation among them. This paper describes a method for linking basic components extracted efficiently from hand-drawn diagrams to OrCAD data format. Experimental results with a hand-drawn diagrams of electronic and logic circuit show utility of the proposed method.

Keywords : preprocessing, feature extraction, pattern recognition, OrCAD link

I. 서론

1980년대에 CAD(Computer Aided Design)가 도입된 후 공학 분야에서 활용 범위가 점점 증가하는 추세이다. 특히 공학 설계 분야에서 중추적인 역할을 담당하는 CAD시스템은 설계분야의 생산성 향상에 많은 도움이 되고 있다. 그러나, 도면 관리 및 기타 관련 기술 업무는 종전 그대로 수작업 관리의 틀을 벗어나지 못하여 각종 기술 정보의 관리 및 이용에 많은 어려움이 있었다[1]-[3]. 설계 분야 관련 업계에서는 손으로 작성된 방대한 도면을 관리하는 데 문제점이 있으므로 도면의 효율적 설계 및 관리를 위해 설계 자동화를 시도하고 있다. 따라서 영상처리 기술과 패턴인식에 의해서 도면을 자동적으로 입력하여 처리하고자 하는 여러 가지의 방법을 연구해 왔다. 기존의 연구 사례를 간략하게 살펴보면 다음과 같다. Byun 등[2]이 제안한 방법은 공학적인 도면에서 기호의 특징들이 대부분 고리 구조로 되어 있으므로 틀 정합(template matching) 방식을 개선한 기하학적 틀을 사용하여 고리 구조를 정확하게 분석하였다. 또한, Beak 등[3]은 최소 경계면(minimum bounding rectangle)과 면적 정보를 이용하여 먼저 문자영역을 분리하고, 루프 심볼(loop symbol)과 비루프 심볼(non-loop symbol)은 각각 파라미터와 기하학적 특징을 이용하여 분리하였다. 여기에서 심볼의 인식은 추출된 심볼의 푸리에 표현자(Fourier descriptor)와 데이터 베이스화된 심볼간의 가장 가까운 인접방법을 이용하였다. Bley[4]는 세그먼트를 기본 요소에 적용하여 화상 그래프(picture graph)로 표현하였고, Groen 등[5]은 전자도면에서 심볼 인식을 확률적 정합에 기반을 두고 심볼의 세션화 과정을 그래프로 표현하였다. Fukada[6]는 논리 회로의 도면을 인식하기 위해 선분 센서(line sensor) 기법으로 신호선의 두께를 추적하는 알고리즘을 제안하였다. 또 Lin 등[7]은 배경 판단과 긴 선분의 세그먼트 추출을 제어 맵(control map)을 작성하여 실행하였다. Ishii 등[8]은 인쇄 회로 기판(PCB)의 도면에 발생하는 색 성분에 따라 요소를 추출하여 인식하는 방법을 연구하였다.

그리고 Kamei 등[9]은 도면의 이치 화상(binary image)중의 선 성분의 방향을 추출하는 방법으로써 벡터화(vectorization)하는 방법을 연구하였다. 최근에는 전자회로 도면의 자동 입력 시스템은 개발되었지만 손으로 작성된 전자회로 도면의 인식이나 전체 도면의 위상(topology) 관계와 전자공학 분야에서 널리 쓰이는 설계 틀인 OrCAD의 정합

(matching) 문제를 이용하지 못하고 있는 실정이다. 본 논문에서는 수작업으로 간략하게 작성된 전자회로 도면의 구조와 특성을 파악하고 OrCAD형태에 맞는 도면을 형성시키기 위해 다음의 4가지 단계로 분류하여 처리하였다. 첫번째 단계에서는 도면 화상의 전처리 단계로서 스캐너에서 받은 화상을 필터링 처리하여, 이치화, 세션화(thinning)과정을 거쳐 기본 요소 추출이 쉽도록 적절한 형태로 변환한다. 두번째 단계는 전자회로 도면의 패턴을 분리한다. 세번째 단계는 기본 요소를 인식하는 단계이다. 네번째는 OrCAD형태에 맞도록 정합시켜 CAD 데이터베이스에 저장한다. II장에서는 요소 추출이 쉽도록 하게 해 주는 영상처리 분야에서 널리 사용되는 전처리 과정의 방법에 대해서 기술하고, III장에서는 기본 요소를 분리하여 추출한 데이터로부터 패턴을 인식하는 과정에 대해서 설명한다. IV장에서는 인식으로부터 얻어진 정보로부터 실용적인 제품 생산에 유리한 OrCAD형태에 맞는 도면으로 연결해 주는 처리 부분에 대해 논의하고, 컴퓨터 시뮬레이션의 결과를 보이고, 끝으로 V장에서는 결론을 맺는다.

II. 전처리

수작업으로 작성된 도면이 카메라나 스캐너로부터 입력되는 과정에서 원래 도면 용지의 어긋남, 용지 위의 입력센서 위치에 따라 생기는 농도값의 차, 스캐너의 특성에 따라 발생하는 해상도의 차 등과 같은 원하지 않은 현상들이 발생할 수 있다. 일반적인 그레이 레벨의 화상은 대부분 잡음에 의해 손상되어 있고, 단순하게 보이는 영역에도 많은 세부적인 정보가 있어 광도의 불연속을 생성한다. 따라서 이러한 단점을 피하기 위해 수작업 도면 정보의 특징 추출이나 인식에 용이하게 하는 전처리 과정이 필요하다. 특징점을 검출하기 위해서는 평균화(averaging)이나 평활화(smoothing)와 같은 저역 통과 필터(low-pass filter)를 이용한 노이즈 제거가 필요하다.

1. 메디안 필터

메디안 필터링은 Turkey에 의해 개발된 비선형 신호 처리로서 화상에서 잡음제거에 유용하게 사용되고 있다[10]. 메디안 필터링의 알고리즘은 다음과 같다.

MDM(Median Method) 알고리즘:

algorithm MDM(m)

(1) array pixels within 3×3 windows.

(2) quick sort to array.

(3) replace median of pixels to center pixel.

2. 이치화

일반적으로 문자나 물체 인식을 할 때 대상이 되는 영상

은 대개의 경우 2 가지 영역으로 구성되어 있다. 즉, 물체 혹은 문자에 해당하는 영역과 그 배경에 해당하는 영역이다. 이러한 문자나 물체를 분리해내기 위해서 간단한 thresholding 방법을 사용한다. 그러나 화상이 복잡할 경우에는 이러한 간단한 방법을 사용할 수 없다. 이러한 문제점을 해결하기 위한 다른 방법으로는 region growing 방법과 boundary detection 방법을 사용하기도 한다. 입력된 수작업 도면의 화상은 그레이 레벨로 구성되므로 이것을 그대로 사용하면 배경과 기본 요소간에 구별하기 어려운 부분이 발생할 수 있다. 따라서 배경과 기본 요소간의 그레이 레벨의 히스토그램을 이용하여 thresholding 값을 선택한다. 영상으로부터 기본적인 요소를 제대로 찾아내기 위해서는 thresholding 값을 잘 결정해야 한다. 따라서 thresholding 값을 구하게 되는 과정은 다음과 같은 엔트로피 알고리즘을 사용하였다[11].

Bilevel process : algorithm HT(h)
algorithm HT(h)

$$① p_i = \frac{f_i}{N^2}, \quad \sum_{i=1}^m f_i = N^2, \quad i=1, 2, \dots, m$$

여기서 $p_i = i$ probability

$f_i =$ observed gray-level frequencies,

$N^2 =$ total number of pixels,

$i =$ number of gray level.

② Find maximum entropy(E_h)

$$E_h = \ln p_h(1-p_h) + \frac{h_h}{p_h} + \frac{(h_m - h_h)}{(1-p_h)}$$

여기서 $h_h = - \sum_{i=1}^h (p_i \ln p_i), \quad P_h = \sum_{i=1}^h P_i$

$$h_m = - \sum_{i=1}^m (p_i \ln p_i), \quad P_m = \sum_{i=1}^m P_i$$

③ Find threshold h .

3. 세션화

앞에서 기술한 thresholding 방법에 의해 처리된 결과는 이진화된 데이터로서 특정 추출을 위한 정보로 이용하기에는 너무 많은 경계점을 갖고 있다. 따라서 앞의 처리 과정에서 특징점을 추출하기 위해서는 세션화 과정이 요구된다. 세션화 과정은 패턴의 모양과 연결성을 유지할 수 있도록 경계점을 줄여 가면서 반복적으로 처리하여 골격을 추출함으로써 영상의 형태 분석을 쉽게 할 수 있도록 한다. 따라서 본 논문에서는 세션화 과정에서 두꺼운 부분을 처리하였고 선분 연결성의 유무는 이진 화소와 현재 화소간의 관계를 고려하여 현재 화소에 인접하는 화소를 추출해내는 수정된 Zhang[12] 방법을 사용하였다. 그림 1과 같이 3×3 윈도우를 이용하면 p1 화소를 중심으로 8개의 인접한 화소로 연결된다. 세션화과정에서 연결성을 유지하기 위해서 반복 과정을 다음 2가지의 부분으로 나누어 처리한다. 첫번째의 부분에서 윤곽점 p1은 다음과 같은 조건하에서 제거된다.

- ① $2 \leq B(p1) \leq 6$
- ② $A(p1)=1$
- ③ $p2 \times p4 \times p6=0$
- ④ $p4 \times p6 \times p8=0$

여기서, $A(p1)$ 는 p2, p3, ...,p9 순서로 '01' 패턴의 수,
 $B(p1)$ 는 p2, p3, ...,p9중 0이 아닌 수이다.

p9	p2	p3
p8	p1	p4
p7	p6	p5

그림 1. 3×3 윈도우.
Fig. 1. 3×3 window.

	2	
0	1	1
0		0
0	1	0
	1	

그림 2. 인접한 현재 화소의 '01' 패턴 수.
Fig. 2. '01' pattern number of the neighboring current pixel.

만약 그림 2와 같이 '01' 패턴을 가지고 있는 경우에는 위의 전체 조건은 ②항 때문에 만족하지 못한다. 그리고 만약 위의 조건중에서 어느 하나라도 만족하지 않으면 p1은 그대로 둔다. 두번째 부분에서는 ③, ④를 변형하여 다음과 같은 형태로 나타낸다.

- ⑤ $p2 \times p4 \times p8=0$
- ⑥ $p2 \times p6 \times p8=0$

즉 ①, ②, ⑤, ⑥중 하나라도 만족하지 않으면 p1은 위와 마찬가지로 그대로 둔다. 그림 3은 두번째의 부분에서 사용되는 위치의 화소를 나타낸다.

	p2	
p8	p1	p4
	p6	

그림 3. 두번째 조건에서 표시하는 위치점.
Fig. 3. Position point displayed in the 2th condition.

위의 과정으로 처리하면 특정 부분에 대해 세션화 처리가 정확하게 이루어 지지 않는 문제점이 발생한다. 따라서 이러한 문제점을 그림 4와 같은 조건을 부가하여 처리함으로써 해결하였다.

x	x	0
1	x	x
x	1	x

①

x	1	x
1	x	x
x	x	x

②

x	1	x
x	x	1
0	x	x

③

0	1	x
x	x	1
x	1	x

④

그림 4. 4가지 형태의 3×3 윈도우.
Fig. 4. Four types of 3×3 window.

반복 과정의 조건 ①, ②, ③, ④ 중 어느것 한가지라도 만족하면 p1=0이 되도록 처리하였다.

III. 기본요소 분리추출 및 패턴인식

전자회로 도면은 문자/숫자, 심볼, 연결선의 3가지 기본 요소로 구성되어 있다. 첫 단계로 도면내에 존재하는 문자/숫자를 분리한 후 도면내 심볼 및 연결선을 특징점으로 추출한다. 직선부분에 대해서는 정보량을 최대한 줄이기 위해 벡터화하고 하나의 심볼도 다양한 기하학적 구조 형태가 발생하므로 하나의 심볼에 대해 학습과정을 통해 후보 패턴을 데이터베이스화 한다. 그 다음에 학습된 심볼의 후보 패턴과 OrCAD에 존재하는 심볼의 데이터베이스인 표준 패턴과 비교하여 패턴 인식을 실현한다.

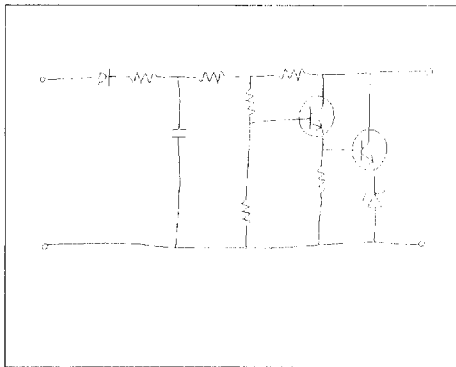
1. 문자/도면 분리

전자회로 도면에서 심볼의 추출과 도면의 벡터화를 위해 문자 분리과정이 우선 처리되어야 한다. 전자회로 도면의 기본요소들은 대부분 문자보다 크므로 쉽게 분리가 된다. 영역 레이블링 방식을 이용하여 도면의 좌측 상단에서 우측으로 스캔하다가 검은 화소를 만나면 그 화소를 최초의 시작점으로 하여 검은 화소를 추적하면서 영역들을 구분한다. 그리고 각 영역의 최소 외곽의 인접 사각형 정보를 만든다. 사각형 정보의 폭과 높이는 (1)에

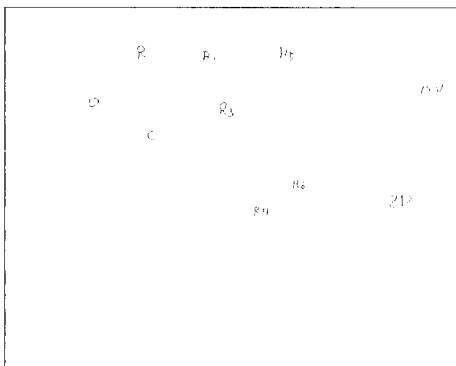
$$w = \Delta x = \max x_i - \min x_i$$

$$h = \Delta y = \max y_j - \min y_j \tag{1}$$

의해 계산한다. 사각형 크기가 100×100 이하의 영역을 문자 영역으로 판단하고 다른 도면과 구분하기 위해 색깔로 영역을 구분하였다. 5×5 이하의 영역에 있는 점은 잡음으로 간주하여 제거하였다. 그림 5(a), (b)는 도면과 문자/숫자를 분리하였다.



(a)



(b)

그림 5. 분리된 결과: (a) 도면, (b) 문자.

Fig. 5. Separation results :

(a) drawn diagram, and (b) characters.

2. 도면내의 선분과 심볼 요소 분리

도면에서 기본 요소를 분리하기 전에 특징점을 추출하고

특징점과 특징점 사이에 변화량이 작은 경우에 직선으로 근사화시키는 알고리즘[13]을 적용하였다. 이 알고리즘에 따라 특징점을 현재 화소를 중심으로 시작점, 분기점 그리고 끝점을 추출하였다. 특징점의 추출시 직선 근사화 알고리즘은 다음과 같다.

SLA(Straight Line Approximate) 알고리즘:

1. Move starting point to origin
2. $f_i = 0$
3. Calculate $\Delta x_i, \Delta y_i$
4. Move (x_i, y_i)
5. Δf_i is increment moving from the point P_{i-1} to next point P_i
 $f_i = f_{i-1} + \Delta f_i$
 where $\Delta f_i = x_i \cdot \Delta y_i - y_i \cdot \Delta x_i$
6. Calculate the length $L_i = \sqrt{x_i^2 + y_i^2}$
7. T is the maximum allowed area deviation per length unit of the approximation segment
 If $|f_i| \leq T \cdot L_i$ is satisfied
 increment i , goto 4
8. Start point for next segment.

위와 같은 과정으로 특징점을 추출한 결과를 그림 6에 나타냈고, 이 결과에서 직선 근사화 알고리즘을 적용하여 얻어진 결과를 그림 7에 나타내었다.

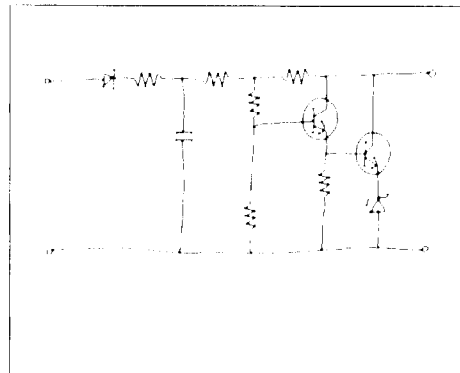


그림 6. 특징점 추출.

Fig. 6. Features extraction.

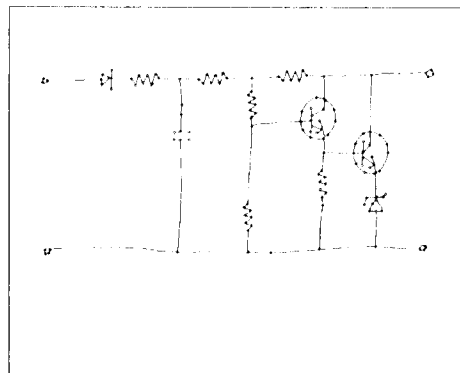


그림 7. SLA을 이용한 특징점 추출(T=1.0).

Fig. 7. The features extraction using SLA(T=1.0).

인접한 특징점을 처리하는 몇가지 방법에 따라 제거한 결과를 그림 8에 나타내었다. (a)는 분기점, 교차점, 끝점 정보를 가지고 처리한 결과, (b)는 직선 근사화로 처리한 결과, (c)는 인접한 특징점을 제거하는 조건으로 특징점이 끝점을

가지면서 특징점과 특징점사이의 화소 길이가 8이하인 경우를 한개의 특징점을 간주하여 처리한 결과이다.

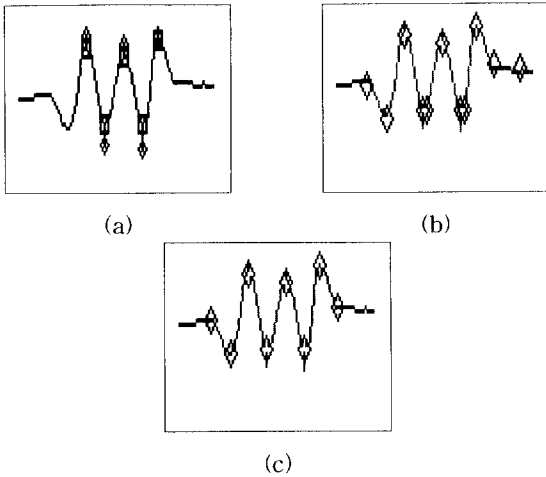


그림 8. 인접한 특징점 제거.
Fig. 8. Removal of neighboring feature points.

도면 화상 처리를 행하는 주된 목적중에 하나는 원 도면의 데이터량 보다 적은 양으로 원 도면의 데이터량과 같이 나타내고자 하는 것이다. 도면에서 선 구조를 좌표축으로 저장하면 데이터량을 줄일 수 있다. 따라서 이러한 목적을 위하여 벡터화(vectorization) 방법을 적용하였다[13]. 먼저 선의 시작점, 끝점 그리고 분기점을 구분하여 세그먼트한다. 각 세그먼트에 대해서 선 추적을 하고 각 점에서의 굴곡점을 구해서 굴곡의 변화가 임계각을 벗어나면 새로운 굴곡점을 설정한다. 특징점과 특징점 사이의 직선으로 간주해야 할 조건을 3가지로 분류해 보면 방향 코드가 일정하게 몇개 이상의 화소로 진행되는 경우, 방향 코드가 일정하게 진행하지 않아도 방향 코드의 규칙성이 있는 경우, 그리고 직선과 직선 사이가 몇개 이하의 화소로 이루는 경우를 직선으로 처리한다. 여기서 직선 구조 집합은 (2)로 표현한다.

$$L = \{l_i : i=1, \dots, n\} \quad (2)$$

여기서 l 의 각 요소는 다음 식과 같이 점의 집합으로 구성된다.

$$l = \{P_j = (x_j, y_j) : j=1, \dots, m\} \quad (3)$$

표 1. 특징점 정보.
Table 1. The information of feature point.

no	coordinate axis	br.no	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	D ₈
:	:	:	:	:	:	:	:	:	:	:
27	(821, 102)	3	0	110	0	0	121	0	0	122
28	(467, 103)	3	117	0	0	123	0	123	0	0
29	(294, 104)	3	108	0	0	124	0	124	0	0
30	(313, 104)	3	116	0	125	126	0	126	0	0
:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:

표 1에서 특징점 번호(no)의 속성은 좌표점, 특징점을 중심으로 가지(branch)가 뻗어나가는 수(br.no)와 방향 정보(D_n)를 갖도록 하였다.

표 2. 세그먼트 정보.
Table 2. Segment informations.

seg.no	node1	node2	length	θ
:	:	:	:	:
116	23	30	3	1.56
117	24	28	18	0.51
118	25	48	21	0.68
:	:	:	:	:
:	:	:	:	:

표 2는 특징점(node1)과 특징점(node2)를 하나의 영역(seg.no)으로 하고 그에 대한 길이 및 각도를 갖도록 하였다. 각도는 x축을 중심으로 이동한 후 node1과 node2간에 이루는 각 θ 를 arctan 값으로 나타낸다.

그림 9는 도면에 존재하는 각 심볼 영역의 부분을 나타낸다.

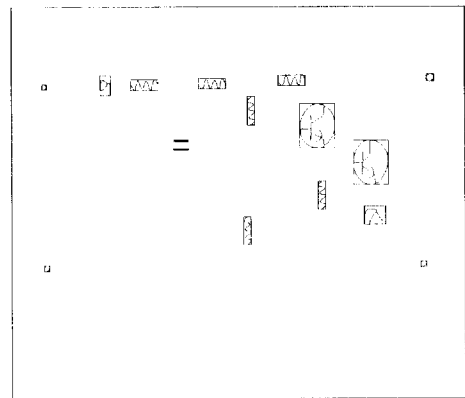


그림 9. 심볼 추출.
Fig. 9. Symbols extraction.

3. 심볼 패턴 인식

긴 선분과 특징점 성질을 고려하여 선분을 분리하면는 심볼들만 구성된다. 심볼인식 단계에서는 특징점의 존재 여부 및 특징점간의 연관관계를 이용한다. 각 심볼이 가지는 x, y축 길이의 정보를 각각 m, n 등분하여 각 등분에 교차하는 수를 특징 파라미터로 정한다. 특징 파라미터는 집합으로 표현하면 다음 식과 같이 나타낸다.

$$Scol = \{c_1, c_2, \dots, c_m\}$$

$$Srow = \{r_1, r_2, \dots, r_n\} \quad (4)$$

본 연구에서는 패턴 인식을 하기 위해 심볼과 외부 연결선 간의 절점 개수와 동일한 절점 개수라도 심볼의 특성에 따라 방향에 무관한 심볼이 있는가 하면 방향에 따라 다르게 표현되는 경우가 있으므로 방향 성분 및 심볼내 loop 개수를 구한다. 심볼내에 있는 전체 화소크기를 저장하여 패턴을 정합하는 경우에 탐색 시간이 많이 소요되므로 각 심볼 크기(B)를 $m \times n$ 크기로 분해하여 블록에 존재하는 심볼 데이터의 유무에 따라 특징량의 파라미터를 구성한다. 그 다음 단계로써 OrCAD에 구축된 표준 심볼 패턴의 특징량과 입력 패턴의 특징량을 비교하여 가장 가까운 패턴을 결정한다.

몇 가지 심볼에 대한 특징 파라미터를 표 3에 나타내었다. 그리고 그림 10은 패턴 인식을 하기 위해 결정 트리 구조를 나타내고 있다.

표 3. 각 심볼의 표준 패턴과 입력 패턴 비교.
Table 3. Comparison of standard and input pattern of each symbol.

패턴	심볼	x	y
표준 패턴	저항	1 1 1 1 1 1 1	6 6 5 4 5 4 6
	다이오드	2 2 2 2 2 2 2	2 3 2 2 3 2 2
	제너 다이오드	4 2 2 2 2 2 5	0 3 2 2 3 2 5
	TR	3 3 4 4 5 4 2	3 3 4 4 4 5 3
	콘덴서	7 0 0 0 0 0 7	2 2 2 2 2 2 2
입력 패턴	저항	1 2 1 1 1 2 1	5 6 5 4 5 4 5
	다이오드	1 2 2 2 2 3 1	2 3 2 2 3 3 1
	제너 다이오드	5 2 2 3 2 2 3	1 3 2 2 3 1 2
	TR	3 3 3 4 5 3 2	3 3 3 4 4 4 3
	콘덴서	6 0 0 0 0 0 7	1 1 2 2 2 2 1

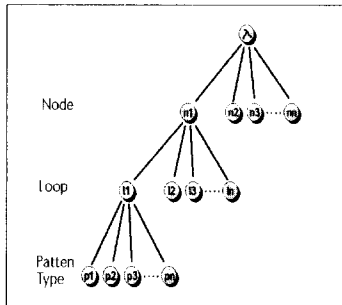


그림 10. 결정 나무 구조.
Fig. 10. Decision tree structure.

IV. 시뮬레이션 및 검토

작업 환경은 586 PC(66MHz)에 윈도우 3.1의 운영체제하에 Boland c/c++로 시뮬레이션하였으며 이러한 과정을 그림 11에 나타내었다.

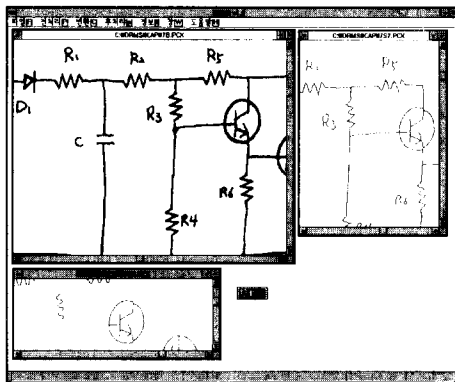
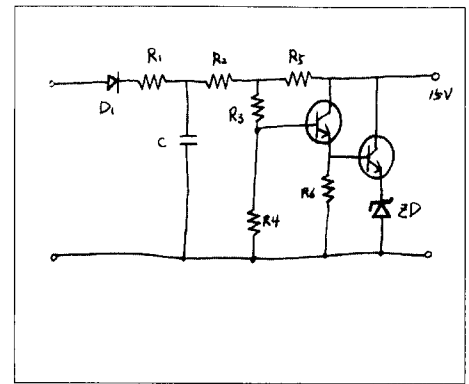


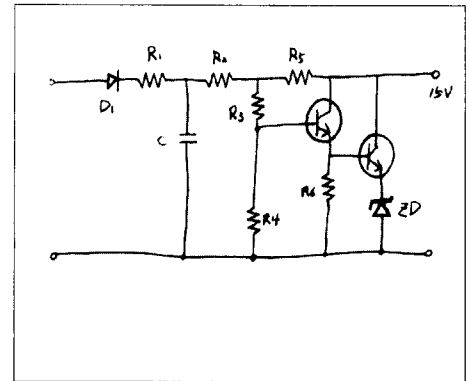
그림 11. 윈도우 환경에서 도면 처리 과정.
Fig. 11. Drawing diagram processing in Windows environment.

수작업으로 된 도면을 HP ScanJet IIcx 스캐너를 통하여 입력하였으며, 화상 정보의 크기와 해상도는 1048×578 pixels, 200 pixels/inch이다. 시뮬레이션 과정에서 처리된 결과를 그림 12에 나타내었다.

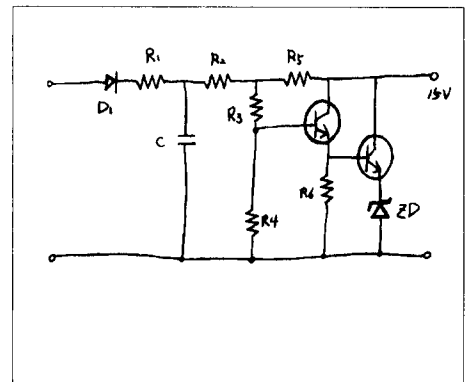
전자회로 도면을 이해하기 위해 심볼을 인식하는 것 이외에도 연결선 처리, 심볼과 선분간의 연결 관계가 명백하게 이루어져야 한다. 연결선에는 교차점, 결합점, 심볼에서 연결성을 내포하고 있다. 전자회로 도면의 위상 관계를 파악해야 한다.



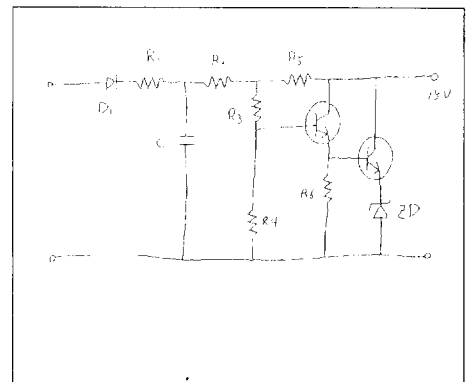
(a)



(b)



(c)



(d)

그림 12. 처리된 결과: (a) 원 화상, (b) 메디안 필터, (c) 이치화, (d) 세선화.

Fig. 12. The results: (a) original image, (b) median filter, (c) bilevel, and (d) thinning.

OrCAD와 연결하기 위해 2가지의 문제점을 해결해야 한다. 첫째, 파일 헤드(head)와 꼬리(tail) 부분 및 각 소자의 심볼의 형태를 분석해야 한다. 둘째, 입력 도면의 심볼의 크기가

가변적이므로 OrCAD 심볼의 크기를 고정시켜 입력 도면의 크기가 같도록 한다. 본 연구에서는 OrCAD와 정합시키기 위해 OrCAD 형식에 맞도록 구성하였고, 입력 패턴이 가변적이기 때문에 OrCAD 크기와 같도록 고정시켜야 하지만 정규화하지 않고 크기가 달라도 심볼 특성을 최대한 살려 인식할 수 있도록 x, y축을 $m \times n$ 하여 각 등분에 교차하는 수를 특징 파라미터로 하여 입력 패턴으로 정한다. 시뮬레이션 결과 탄력정합에 의해 최소 거리의 임계값이 3이하일 때 입력 패턴과 표준 패턴을 동일한 패턴으로 인식한다.

표 4. OrCAD 헤더 정보.

Table 4. Head informations of OrCAD.

Byte	내용	실제값
17	OrCAD 도면 파일	"Schematic", 0dh, 0ah, 1ah
5	명시되지 않음	00h, 00h, 00h, 00h, 00h
2	필요 메모리 크기	Variable value
2	명시되지 않음	00h,00h
2	도면 x 위치	Variable value
2	도면 y 위치	"
1	배율	"
2	명시되지 않음	00h,00h
2	예약	4fh,01h
2	Sheet number	Variable value
2	number of sheets	"
1	제목 & 도면 크기	"
1	날짜 문자열 갯수	"
18	날짜 문자열	"
1	Document number 문자열 갯수	"
36	Document number	"
1	Revision code 문자열 갯수	"
3	Revision code	"
1	title of sheet 문자열 갯수	"
44	title of sheet	"
1	Organization name 문자열 갯수	"
44	Organization name	"
1	1st Address line 문자열 갯수	"
44	1st Address line	"
1	2nd Address line 문자열 갯수	"
44	2nd Address line	"
1	3rd Address line 문자열 갯수	"
44	3rd Address line	"
1	4th Address line 문자열 갯수	"
44	4th Address line	"

표 5. 선분 정보 구조.

Table 5. The structure of line information.

Byte	내용	실제값		
		wire	bus	결합점
1	code	03h	04h	05h
1	길이	08h	08h	04h
1	예약	00h	00h	00h
2	시작 x 좌표	Variable value	Variable value	Variable value
2	시작 y 좌표	"	"	"
2	끝 x 좌표	"	"	"
2	끝 y 좌표	"	"	"

표 6. 회로 소자 정보 구조.

Table 6. The structure of circuit element information.

Byte	내용	실제값
1	code	02h
1	길이	Variable value
1	예약	00h
2	x 좌표	Variable value
2	y 좌표	"
2	부품심벌도 x 좌표	"
2	부품심벌도 y 좌표	"
2	부품 용량 x 좌표	"
2	부품 용량 y 좌표	"
1	방향	01h or 21h or 81h
4	부품 상세 정보	Variable value
7	부품 목록 코드 2	"
4	부품 순서 번호	"
1	심벌도 문자열 수	"
?	심벌도 문자열	"
1	부품용량 문자열 수	"
?	부품용량 문자열	"

표 4, 표 5, 표 6은 OrCAD의 헤더정보 및 기본요소 성분들의 형식을 몇가지 나타내 보였다. 그림 13에서 도면과 문자가 겹쳐져 있는 경우에는 특징점 추출에 문제가 발생하므로 편집 기능으로 재편성하였고, 문자/숫자는 인식하지 않은 상태에서 OrCAD형태로 변환하였다. OrCAD 틀은 부품에 대한 성질만 알면 부품명이 표시되고 문자/숫자 인식 부분을 본 연구에서는 하지 않았기 때문에 크기값에 대해서 미지수로 나타낸다.

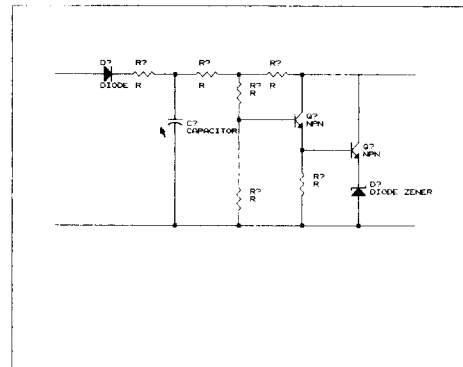


그림 13. OrCAD형태로 변환.

Fig. 13. Conversion to OrCAD type.

V. 결론

본 논문에서 손으로 그린 전자회로 도면을 제품생산에 보다 유리한 OrCAD에 적합한 형태로 변환하는데 목적을 두었다. 전처리 과정은 기존에 적용하는 방법을 조금 다른 측면에서 처리하였다. 도면내에 있는 심볼의 기본 요소와 문자/숫자를 분리하였고 도면내의 기본 요소들 추출은 먼저 선분을 추출하면 나머지 요소들이 심볼의 기본 요소들로 존재한다. 심볼의 기본 요소들은 각 심볼이 가지는 크기는 일정하지 않으므로 최대 x, y축 길이를 n등분하여 균등하게 분할한다. 각 심볼이 분할된 x성분의 교차점의 갯수와 y성분의 교차점의 갯수를 파악하여 표준 패턴과 입력 패턴간의 탄력 정합 패턴을 이용하여 인식하도록 하였다. 겹쳐진 부분은 기존 연구에서도 많은 어려움이 있어 본 연구에서는 편집 기능으로 재편성하여 처리하였다. 차후 패턴 정합을 신경회로망을 이용하여 학습시키는 방법으로 적용해 보고자

한다.

참고문헌

- [1] A. J. Filipski, R. Flandrena, "Automated conversion of engineering drawings to CAD form," *IEEE*, vol. 80, no. 7, pp. 1195-1209, July, 1992.
- [2] S. W. Byun, S. W. Lee, J. H. Kim, "A basic component extractor for hand-drawn diagram recognition," *KIASS*, vol. 20, no. 7, pp. 944-956, July, 1993.
- [3] Y. M. Beak, J. W. Seok, S. I. Chien, C. S. Hwang, "The vectorization and recognition of circuit symbols for electronic circuit drawing management," *KITE*, vol. 33, no. 3, pp. 176-185, March, 1996.
- [4] Heinrich Bley, "Segmentation and preprocessing of electrical schematics using graphs," *CVGIP*, vol. 28, no. 1, pp. 271-288, 1984.
- [5] Frans C. A. Groen, "Symbol recognition in electrical diagrams using probabilistic graph matching," *PRL*, vol. 3, no. 5, pp. 343-350, September, 1985.
- [6] Youji Fukada, "A primary algorithm for the understanding of logic circuit diagrams," *Pattern Recognition*, vol. 17, no. 1, pp. 125-134, 1984.
- [7] X. Lin, S. Shimotsuji, M. Minoh, T. Sakai, "Efficient diagram understanding with characteristic pattern detection," *CVGIP*, vol. 30, no. 1, pp. 84-106, 1985.
- [8] M. Ishii, Y. Ito, M. Yamamoto, H. Harada, "An automatic recognition system for color drawings of printed circuit board," *IEICE*, vol. J71-D, no. 4, pp. 685-692, 1988.
- [9] K. Kamei, Y. Nakamura, S. ABE, "Vectorization of drawings with raster operation," *IEICE*, vol. J72-D-II, no. 1, pp. 32-39, 1989.
- [10] William K. Pratt, *Digital Image Processing*, John Wiley & Sons, Inc., 1978.
- [11] Ahmed S. Abutaleb, "Automatic thresholding of gray-level picture using two-dimensional entropy," *CVGIP*, vol. 47, no. 1, pp. 22-32, 1989.
- [12] Zhang, T. Y and C. U. Suen, "A fast parallel algorithm for thinning digital patterns," *Comm. of the ACM*, vol. 27, no. 3, pp. 236-239, 1984.
- [13] V. Nagasamy, N. A. Langrana, "Engineering drawing processing and vectorization system", *CVGIP*, vol. 49, no. 1, pp. 379-397, 1990.



박 영 식

1953년 9월 17일생. 1979년 동대학교 전자공학과 졸업. 1990년 동대학원(박사). 1992년 ~ 현재 동의대학교 전산통계학과 교수로 근무. 주관심분야는 공장자동화, 신경회로망, 영상처리.



김 진 홍

1957년 11월 12일생. 1983년 동아대학교 전자공학과 졸업. 1992년 동대학원(박사) 1987년 ~ 현재 대학 전산학과 교수로 근무. 주관심분야는 영상처리, 압축, 암호화.