

컴퓨터 비전에 의한 PCB 검사를 위한 검사 정보 생성 시스템 개발*

이철수** · 고은희***

Generating a Modified RLC(MRLC) from Gerber File for the PCB Inspection

Cheol Soo Lee · Eun Hee Go

〈Abstract〉

For the PCB inspection by computer vision, in some cases, the MRLC file should prepared. The MRLC file contains a RLC(Run Length Code) and a direction flag. In this paper, a generating method of MRLC is described. It is composed of two procedure as followings; (i) rasterizing Gerber file which is a vectorized image of PCB panel, and (ii) calculating a MRLC that is useful for the inspection as a template image. The suggested procedures are written in C-language and executable on Windows 95 and Windows NT.

1. 서 론

에칭에 의한 PCB 제작은 동적충판에 에칭 레지스트를 부착하고 마스크 필름에 의하여 레지스트를 현상하여 구리가 필요 없는 부분의 에칭 레지스트를 제거하고 그것을 에칭함으로써 만들어진다. 에칭된 기판은 세척되고 다층 기판인 경우에는 이와 같은 과정을 반복하여 만든다. 그 후에 드릴 가공을 하고 검사를 거치면 하나의 PCB 기판으로 완성된다[1]. 〈그림 1〉은 PCB 제작 과정과 검사 공정과의 관계를 도시한 것이다.

위와 같은 PCB 설계와 검사에 있어서 CAD데이터는 중요한 역할을 하게 된다. CAD에서 이루어지는 PCB 설계는 주어진 회로에 의하여 구리가 남아 있어야 할 부분과 없어야 할 부분을 구분하고 소자가 삽입될 위치에 구멍이 뚫어지도록 하는 데이터를 만드는 것이다. 설계된 데이터는 PCB 제작을 위한 데이터로 만들어져서 전달되는데 이에 대한 정보는 일반적

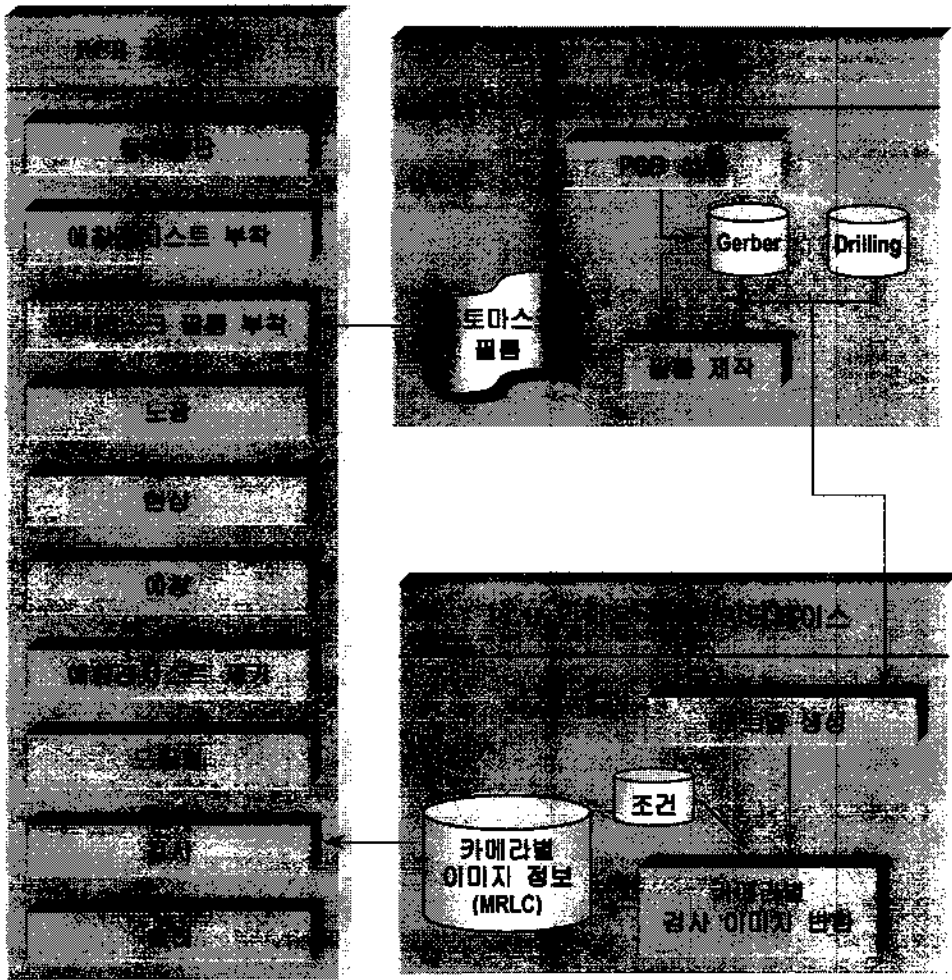
으로 Gerber 파일 형식으로 되어있다. Gerber 파일은 PCB의 이미지를 벡터 형식으로 갖고 있다. Gerber 파일은 광원의 크기와 이동 궤적 등을 일반적인 NC 코드의 형태로 표현하고 그것에 의하여 필름이 제작되도록 하는 것이다. 마스크 필름을 제작할 때 정해진 크기의 빛을 필름에 쏘여 주게 되고 쏘여진 부분에 대해서는 후에 에칭이 일어나지 않는다. 에칭 과정에서 PCB 이미지의 선 폭이 줄어들 수 있으므로 에칭 조건에 따라 Gerber 파일에서 광원의 크기를 조정할 필요가 있다.

초기에 CAD에서 설계된 것은 PCB 한 장에 대한 정보만을 담고 있다. 실제로 PCB의 제작 공정에서는 하나의 판넬에 여러 장의 PCB가 배치되어져서 생산될 수 있으므로 이들에 대한 배치 작업이 필요하게 된다. 이러한 작업은 PCB 제조의 바로 전단계에서 이루어지게 되고 여기에서의 결과도 Gerber 파일 형태로 만들어진다.

* 본 논문은 LG전자와의 산학협동 연구에 의해 이루어졌음.

** 전남대학교 산업공학과

*** 전남대학교 자동차연구소



〈그림 1〉 PCB 제작과 검사 공정

제조된 PCB는 원래의 설계 데이터와 비교하여 불량 여부를 판단하여야 한다[2][3][4]. 이와 같은 검사에 컴퓨터 비전을 이용한 검사의 자동화를 이루고 있다. 검사를 위해서는 기준이 되는 이미지를 가지고 있어야 하는데 이것은 CAD에 의하여 설계되고 배치된 정보를 이용하여 만들어질 수 있다. 본 논문에서 사용한 시스템의 경우, 컴퓨터 비전에 의하여 읽어 들인 데이터는 RLC(Run Length Code)에 의하여 표현되므로 검사의 기준이 되는 이미지도 같은 형식으로 표현되어야 한다. 카메라에서 읽어 들인 데이터와 본 논문에서 제시된 아이디어로 생성된 데이터는 실시간으로 비교되어 PCB의 양/불량을 판단하게 된다.

효율적인 검사를 위해서 RLC에 덧붙여서 유효정보가 포함되는데 본 논문에서는 이것을 MRLC(Modified RLC)라고 한다. 설계된 PCB는 Gerber 파일 형식[5]으로 되어 있기 때문에 이것을 MRLC 형식으로 변환하여야 자동화된 검사에 이용할 수 있다.

본 논문은 컴퓨터 비전에 의한 자동 검사작업을 위해서 CAD로 설계되고 배치된 PCB의 산출물인 Gerber 파일을 카메라가 읽어 들인 정보와 같은 형식인 MRLC로 만드는 방법과 검사를 위한 유효 정보를 생성하는 방법에 대하여 다루고 있다.

2. 검사 정보 생성 시스템의 구성 및 흐름도

컴퓨터 비전에 의한 PCB 판넬 검사 시스템의 주요 기능은 CAD에서 만들어진 PCB와 판넬 정보를 입력으로 하여 그것을 검사 시스템에서 원하는 MRLC 파일 형식으로 변환하여 주는 것이다. 세부적으로는 다음과 같은 기능이 필요하다.

(1) 상이한 크기와 패턴을 갖는 PCB의 결합을 영상 시스템에 의해서 검출하기 위해서 설계 단계의 CAD 데이터와 영상 처리를 위한 데이터 형식과의 인터페이스를 구성한다.

(2) PCB 제작을 위한 설계단계의 CAD 데이터는 Gerber 형식으로 제공되는데 이것은 먼저 이미지 확인용으로써 비트맵 이미지를 만들어주어야 하고, 결합 검사를 위해서 다시 MRLC 형식으로 변환되어야 한다. 비트맵 이미지는 카메라의 해상도 및 PCB의 크기에 맞게 조정되어야 하고 MRLC는 결합 검출에 유용한 별도의 정보를 제공하여야 한다. 이를 위해서 사용자가 MRLC 생성방법을 조정할 수 있도록 한다.

(3) Gerber 파일의 비트맵 이미지 및 MRLC 형식으로의 변환 후, 작업자의 확인이 가능하도록 화면으로 디스플레이 해준다.

(4) 검사 작업시 해상도의 변화에 따라 한 개의 카메라가 대응할 수 있는 크기가 달라지기 때문에 이러한 변화에 대응할 수 있는 기능이 필요하며 출력되는 MRLC 양과 영역의 양이 가변적이어야 한다. 이를 위하여 개발된 시스템은 이미지의 정밀도와 카메라의 스캔 능력에 맞도록 자동으로 MRLC 영역이 결정된다.

(5) 조건 선택 및 기능 수행과 그 결과의 디스플레이 등에 관한 부분을 그래픽 유저 인터페이스로 구성하여 최소한의 입력으로 이를 수행할 수 있도록 이러한 일련의 과정을 체계화한다.

<그림 2>는 검사정보 생성 시스템의 흐름도로써 Gerber 데이터로부터 비트맵 이미지를 생성하고, 이를 기준으로 MRLC 데이터를 만드는 과정을 보이고 있다.

<그림 2>의 검사정보 생성시스템의 단계별 기능은 다음과 같다.

(1) 주어진 Gerber 파일을 이용하여 포토 플로팅될 이미지를 생성한다. 이 이미지는 비트맵 형태로 만들어진다.

(2) 드릴링 정보를 이용하여 드릴 작업에 대한 이미지를 생성한다.

(3) 포토 플로팅 정보와 드릴링 정보를 합성하여 실제 생산될 판넬 이미지를 만든다.

(4) Gerber 파일을 포토 플로팅하였을 때 생길 것으로 예상되는 이미지를 비트맵으로 변환하였으므로 그것을 확인하고 검사 방법을 검토한다.

(5) 검토된 방법을 토대로 하여 검사 정보를 생성하기 위한 조건들을 지정한다.

(6) 카메라 정보와 검사 조건을 기준으로 카메라별 MRLC 정보를 생성한다.

3. Gerber 파일의 래스터라이징(Rasterizing)

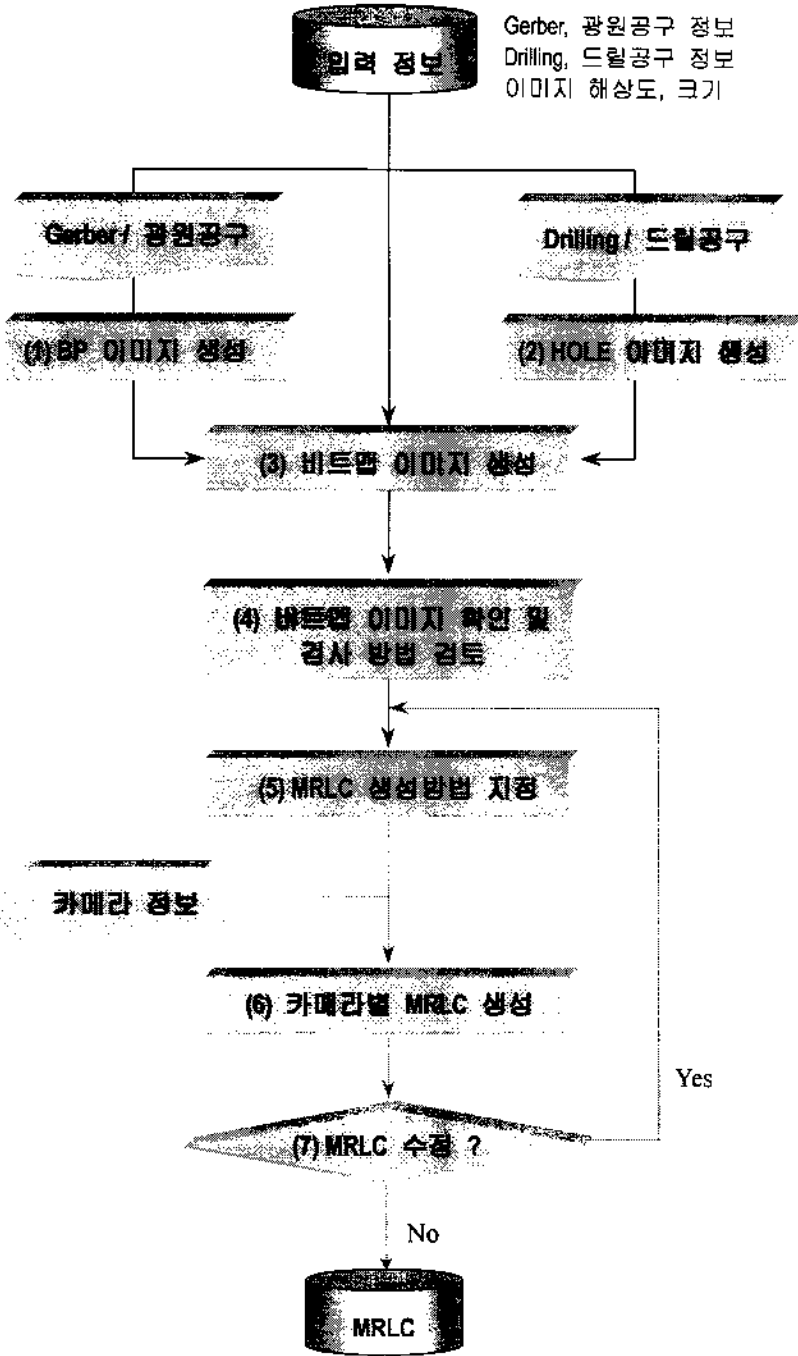
3.1 개요

Gerber 데이터 G54*D10*X10Y10D03* 는 ‘공구 10번을 사용하여 (10,10) 위치를 표현하라’는 명령이다. 비트맵 이미지의 정밀도가 0.25mm라면 기준 단위 1mm는 4개의 pixel로 표현될 것이다. 공구 10번은 지름이 2mm인 원형이라고 하자. 공구의 이동은 항상 xy평면에 수직을 유지하며 회전은 하지 않는다. 이 경우 공구 위치 $p(x,y) = (10,10)$ 에 대한 비트맵 이미지는 공구에 의해서 가리워지는 부분이 되며 결과는 <그림 3>의 (b)와 같다. <그림 3>의 (b)는 판넬 크기만큼의 이미지를 표현할 수 있는 메모리를 나타낸다. 공구에 의해 가리워지는 부분을 ‘패턴’이라 하자. 결국 비트맵 이미지는 Gerber 파일에 정의된 점, 직선, 곡선등의 경로를 따라 만들어지는 패턴 이미지들의 합이 될 것이다.

Gerber 파일로부터 만들어지는 비트맵 이미지는 일반적인 BMP 형식을 따르지 않고 별도의 이진화된 코드 형태로 관리된다. <그림 4>는 <그림 3>의 비트맵 이미지의 메모리 형태를 보여준다.

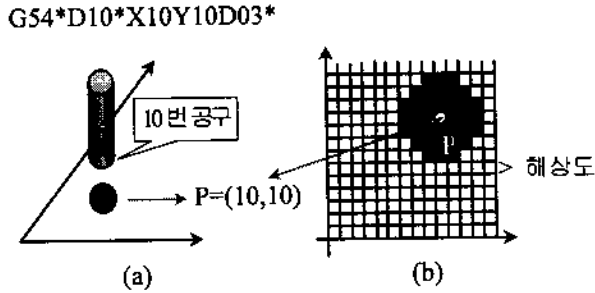
메모리상의 1bit는 정밀도 만큼의 거리를 나타내며 행은 y좌표, 열은 x좌표를 의미한다. 따라서 1mm를 표현하기 위해서는 $1mm / 0.25mm = 4bit$ 가 필요하다. $600mm \times 200mm$ 크기의 판넬에 대한 이미지를 표현하기 위해서는 가로가 $600 \times 4bit = 2400bits$, 세로로 $200 \times 4bit = 800bits$ 가 필요하므로 총 $(2400 \times 800 = 1920000bits)$ 의 메모리가 필요하다. 이미지는 마지막 행을 시작으로 하여 행 단위로 저장된다. 따라서 비트맵 이미지는 y축에 대칭인 이미지로 관리된다. 공구 좌표 10mm는 메모리상에서 $10mm \times 4 = 40$ 번째 bit에 해당하기 때문에 [40,40] pixel을 중심으로 공구 반지름인 1mm 거리 안에 존재하는 모든 pixel들이 패턴으로 처리되었다.

메모리의 1bit가 한 픽셀을 나타낸다고 하였다. 이것을 고려

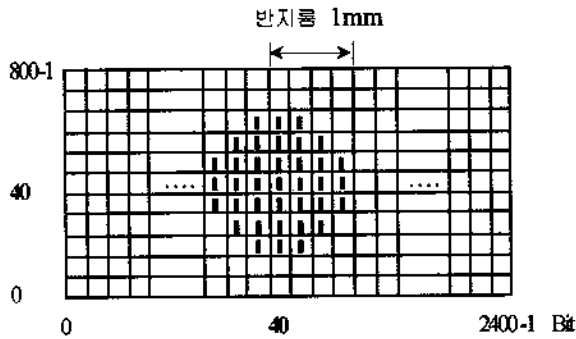


〈그림 2〉 검사 정보 생성 시스템 흐름도

하여 실제좌표 (x,y)의 이미지를 참조하거나 수정할 수 있다. 실제좌표 (x,y)에 해당하는 비트맵 메모리내 위치는 기준 단위



〈그림 3〉 비트맵 표현



〈그림 4〉 600mm × 200mm 이미지의 메모리 상태

1mm를 표현하는데 필요한 pixel 개수를 실좌표값에 승산하여 계산하면 된다. 계산된 bit단위 위치에서의 값이 1인 경우는 패턴, 아니면 공백을 의미한다. 비트맵 이미지용 메모리가 1바이트 크기의 2차원 배열로 표현되었다면 실제 좌표 (x,y)에 해당하는 pixel 즉, 메모리 주소는 y에 대하여는 그대로 참조하여 쓰고 x에 대해서는 1byte에 8개의 픽셀을 표현할 수 있으므로 (x/8)번째 바이트의(segment) (x % 8)번째 비트(offset)이다.

3.2 래스터라이징

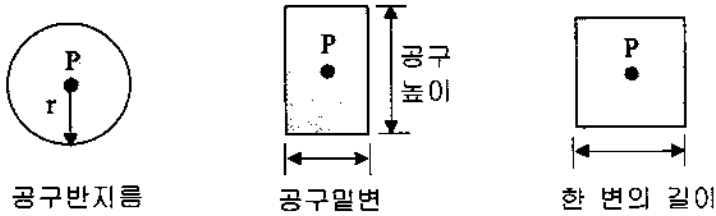
3.2.1 점 이미지의 래스터라이징

광원에 대한 점 이미지는 임의 점으로 이동한 후 빛이 점멸함으로써 만들어진다. 따라서 만들어지는 이미지는 광원 공구와 같은 모양을 하게 된다. Gerber 파일에서 사용하는 광원 공구는 〈그림 5〉와 같이 원형, 직사각형, 정사각형 등의 형상을

갖는다. 공구 정보는 원인 경우는 반지름, 직사각형인 경우에는 밑변과 높이, 정사각형인 경우에는 한 변의 길이가 주어진다. Gerber 파일에서의 공구 이동 좌표값에 광원 공구의 중심을 위치시킨다.

이들 공구에 대하여 임의의 위치 P에서 공구의 비트맵 이미지는 다음과 같이 구해진다.

- (1) 원형 공구인 경우에는 비트맵 상의 한 점에서 P와의 거리가 공구 반지름 r 이하인 모든 점들은 ON된 것으로 한다.
- (2) 직사각형 공구에 대해서는 비트맵 상의 한 점 P에서 x 방향으로 (공구 밑변 / 2), y 방향으로 (공구 높이 / 2) 이내인 영역 내 모든 점들을 ON된 것으로 한다.
- (3) 정사각형 공구인 경우에는 P에서 x와 y 방향으로 (공구 한 변의 길이 / 2) 이내에 존재하는 모든 점들을 ON된 것으로 한다.



〈그림 5〉 Gerber 파일의 광원

3.2.2 선분의 래스터라이징

직선은 점의 연속이동으로 생기는 자취라고 할 수 있으므로 직선의 비트맵 이미지는 점 비트맵 이미지의 합으로 표현이 가능하다. 〈그림 6〉은 공구 형상에 따른 선분의 비트맵 이미지화 과정을 보여주는 것으로 (a)와 (c)는 공구의 이동에 따라 생기는 점 이미지를 표현한 것이다. 완전한 선분의 비트맵 이미지는 〈그림 6〉의 (b)와 (d)같이 P₁에서 P₂까지의 점의 연속 이동에 의한 경계영역과 시점과 종점에서의 점 이미지의 합으로 표현된다.

3.2.3 원호의 래스터라이징

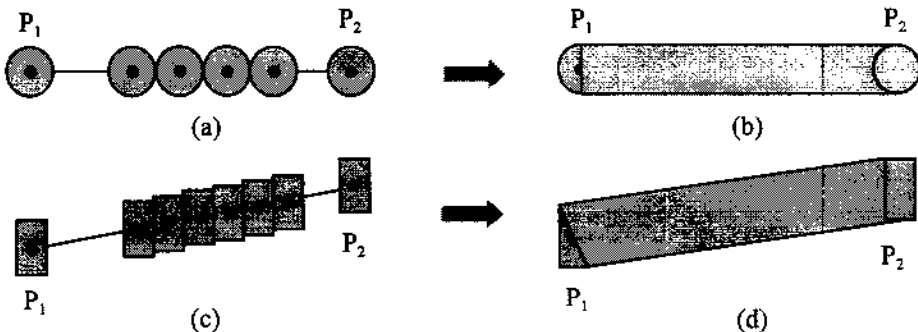
원호는 허용오차에 맞추어 여러 개의 작은 선분으로 쪼개어 표현할 수 있다. 원호를 래스터라이징하는 경우에 이 방법을 사용한다. 즉 원호를 허용오차에 맞는 작은 선분으로 나누고 나누어진 선분을 앞에서 설명한 선분의 래스터라이징 처리를 수행하면 원호를 래스터라이징할 수 있다.

이 때 곡선을 구성하는 직선 요소(Line Segment)의 개수는 생성되는 원호의 비트맵 이미지 정밀도를 결정하는 주요 변수이다. 이 직선 요소를 세분화할수록 곡선은 부드럽게 표현되지만 이미지 생성에 소요되는 시간은 증가한다. 따라서 이를 적절히 고려해야 할 필요성이 있다. 원호의 사이각과 반지름

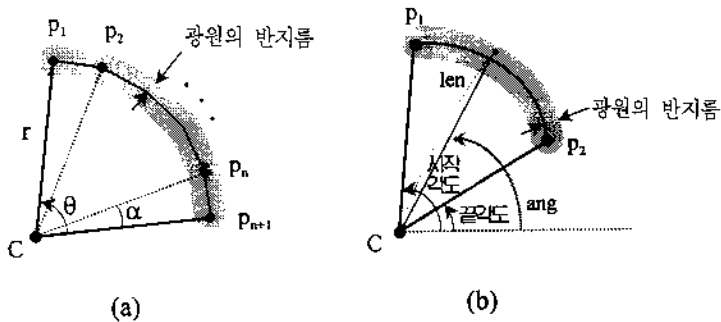
을 고려하여 직선 요소의 개수 n이 결정되면 원호의 사이각을 n등분하는 각 점들의 좌표를 구한다. 〈그림 7〉의 (a)와 같이 원호를 표현하는 직선 요소 개수가 n이면 원호 상의 점 P_i는 다음과 같이 구할 수 있다.

$$\begin{aligned}
 &r : \text{원호의 반지름} \\
 &\theta : \text{원호의 사이각} \\
 &\alpha = (\theta / n) \\
 &\text{for } (i = 0; i <= n; i++) \\
 &\{ \\
 &\quad \alpha_i = \text{시작각} + (\alpha \times i) \\
 &\quad P_i = (cx + r \times \cos(\alpha_i), cy + r \times \sin(\alpha_i)) \}
 \end{aligned}$$

선분에 의한 원호의 근사 표현은 선분의 개수를 결정하는 것이 주요 변수이지만 원호의 식을 그대로 이용한다면 그와 같은 문제는 없을 것이다. 〈그림 7〉의 (b)와 같이 광원 공구가 원호를 따라간다면 비트맵 중에서 원호의 내부에 속하는 픽셀을 찾는 것은 쉬운 일이다. 즉, 한 픽셀이 원호의 중심점을 중심으로 x축과 이루는 각도를 ang이라고 하고, 원호의 중심점과 픽셀 간의 거리를 len이라고 하자. 이 때 광원 공구가 지나가게 될 픽셀은 ang이 시작 각도와 끝 각도의 사이에 있고, len이 (원호의 반지름 · 광원의 반지름)과 (원호의 반지름



〈그림 6〉 선분의 비트맵 이미지



〈그림 7〉 원호의 비트맵 이미지

+ 광원의 반지름) 사이에 있는 것이다.

이 방법은 광원 공구가 원형인 경우에는 간단히 적용할 수 있지만 사각형 공구에 대해서는 간단하지 않기 때문에 본 논문에서는 원호를 선분의 연속으로 분할하여 계산하는 방법을 채택하였다.

3.2.4 드릴 홀의 래스터라이징과 포토 플로팅 정보의 합성

드릴의 홀 정보는 드릴의 반지름을 광원 공구의 반지름으로 하여 앞에서 설명한 점 이미지의 래스터라이징에 의하여 만들 수 있다. PCB 검사는 드릴 작업 이후에 하기 때문에 MRLC 정보를 생성할 때는 드릴링 이미지가 포함된 비트맵 이미지가 필요하다. 따라서 드릴링 이미지를 포토 플로팅 이미지와 합성할 필요가 있다. 〈그림 8〉은 두 이미지가 합성되는 과정을 보이고 있다. 〈그림 8〉의 (a)는 Gerber 데이터에서 만들어진 판넬 이미지의 일부분이고, (b)는 드릴 작업에 대한 홀 이미지의 일부분이다. 판넬 이미지에서 홀 이미지를 삭제함으로써 (c)와 같은 합성된 이미지를 얻을 수 있다.

4. MRLC 생성

4.1 RLC와 MRLC

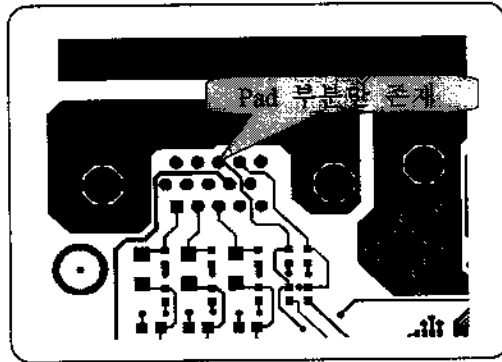
RLC는 비트맵의 y축 방향으로 진행하면서 색깔이 같은 것에 대해서는 하나의 RUN으로 처리하여 데이터를 표현하는 방식이다. 하나의 RUN은 시작 위치와 동일한 색깔을 갖는 y축 방향의 픽셀의 개수, 색깔 등이 하나의 레코드로 표현된다. 따라서 한 개의 y축 방향의 스캔라인에 한 개 이상의 RUN이 존재하게 된다.

위와 같이 RUN을 표현한다면 한 개의 RUN만으로는 주변의 이미지를 알 수 없는 상태이다. 그러나 PCB 검사에 있어서는 주변 이미지에 대한 정보가 필요하다. 예를 들면 현재의 RUN이 PCB 상의 패턴과 평행한 RUN의 일부라는 것을 안다면 PCB 검사에 중요한 정보가 되고 검사를 효율적으로 할 수 있을 것이다. 따라서 RLC에 주변 이미지와의 관계를 나타내는 방향 플래그(flag) 정보를 포함시킬 필요가 있다. 본 논문에서는 MRLC(Modified Run Length Code)라고 하여 하나의 레코드에 시작 위치, 색깔, 방향 플래그, 홀 플래그 등의 정보를 알려준다.

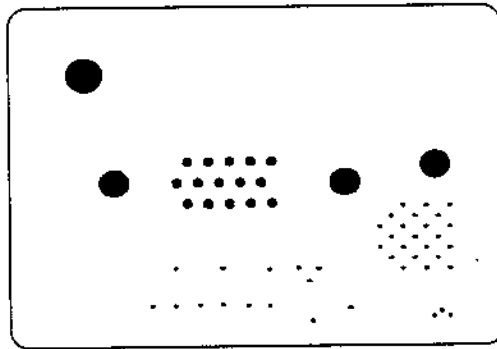
PCB 판넬의 실제 검사영역은 가이드 격쇠 내부이다. Gerber 파일에는 판넬의 네 모서리에 있는 십자형 마크를 경계로한 사각형 내부의 데이터가 모두 들어가 있다. 이 십자형 마크는 다층기판을 적용한 후에 판넬의 가장자리를 절단하는 트리밍 공정의 기준이 된다.

〈그림 9〉는 PCB 가관상에서 볼 수 있는 주요 기준 마크들을 보여준다. 판넬 이미지의 상하 구별은 두개의 인쇄 guide hole로 하며, 이는 좌측 상단 A 또는 B의 자리에 위치한다. Guide 격쇠를 경계로한 영역의 긴 쪽을 장축, 짧은 쪽을 단축이라 하며 NC guide hole은 장축의 상하에 나타난다.

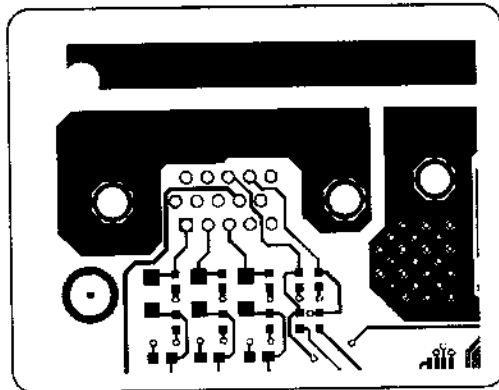
위와 같은 기준 마크들에 대한 패턴이미지는 Drill Gerber 파일에서 제공된다. 이 파일은 트리밍 절단선, 인쇄 Guide hole, photo guide hole, NC guide hole, drill check guide hole과 같은 각종 hole에 대한 위치와 판넬 완성 후 수행할 드릴작업에 대한 위치정보를 Gerber 형식으로 제공하는 텍스트 파일이다. 이 파일에서 Guide hole에서 실제 검사 시작 위치까지의 거리나 좌표, 복수개의 카메라 검사시스템에서 각 카메라에 작업을 할당하기 위한 실제 검사영역의 시작과 끝 좌표, 기구부의 제어를 위한 각종 기준홀에 대한 좌표를 알 수 있다.



(a) 판넬 이미지



(b) 홀 이미지



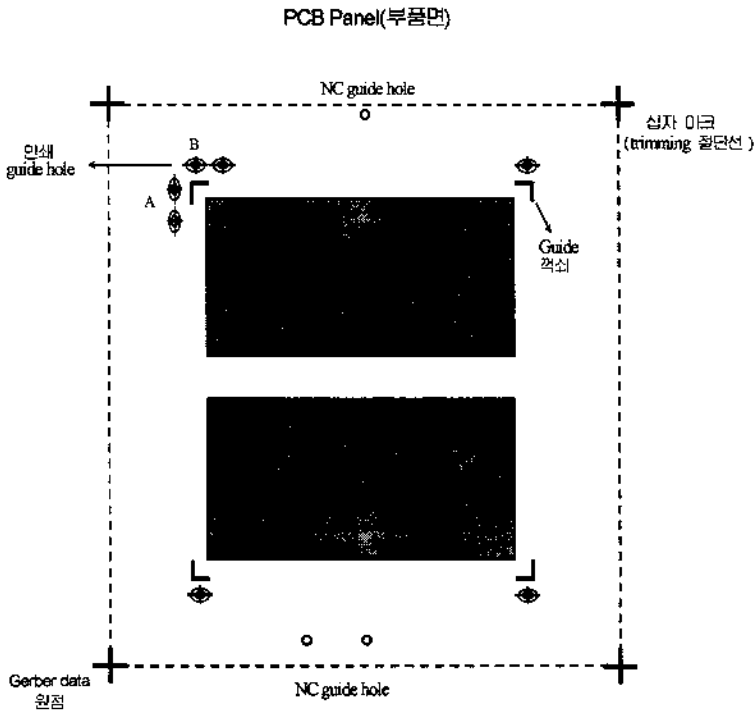
(c) 완성 이미지

〈그림 8〉 비트맵 이미지 생성 과정

4.2 PCB SCAN 방향

〈그림 10〉은 검사대에 놓여지게 될 PCB 판넬과 카메라의

관계를 보여준다. PCB 판넬은 두개의 NC guide hole이 오른쪽에 위치하고, 한 개의 NC guide hole이 왼쪽에 오도록 배치한다. 카메라의 위치는 고정되어 있고 PCB가 D 방향으로 이



〈그림 9〉 PCB 기판상의 기준 마크

송되면서 검사를 하기 때문에 PCB 스캔 방향은 d가 된다. 스캔 용량을 고려하여 카메라별 검사영역이 결정되면 스캔 방향 d를 따라 이동하면서 한 라인씩 스캔된다. 이미지의 위쪽에서 아래쪽 방향으로 입력되는 한 개의 스캔라인은 몇 개의 RUN으로 구성되어 있고, 각 RUN에 대해서 MRLC 정보를 만들어 준다.

4.3 MRLC 생성 절차

비트맵 이미지에서 MRLC 정보를 만드는 절차는 아래와 같다. 만들어지는 MRLC는 검사 과정에서 기준 이미지(reference image)로 사용될 것이다.

(1) 검사 방법을 결정한다.

수평 패턴 이미지로의 분류는 어느 정도까지 허용할 것인지 결정하고, 카메라의 스캔 용량을 조정한다.

(2) MRLC 생성 영역 계산

비트맵 이미지에서 ' ' 모양의 가이드 객체를 찾는다. 판넬의 네모서리에서 일정 영역 만큼의 이미지를 탐색하여 객체를 찾으면 된다. MRLC 생성 영역은 가이드 객체를 포함하지

않은 최대 사각형 영역이다.

(3) 스캔 방향과 스캔 원점을 결정한다.

MRLC를 만드는 시점에서의 비트맵 이미지는 장축이 가로 방향으로 놓이며 두개의 NC guide hole이 오른쪽에 위치하도록 조정되어 있는 상태이다. 판넬의 스캔 작업은 두개의 NC guide hole에서 한 개의 NC guide hole쪽으로 진행된다. 스캔 원점은 비트맵 이미지에서 2개의 NC Guide Hole이 있는 오른쪽/상단 모서리 위치이다. 이 스캔 원점은 1번 카메라의 첫 번째 스캔 라인의 시점이 된다.

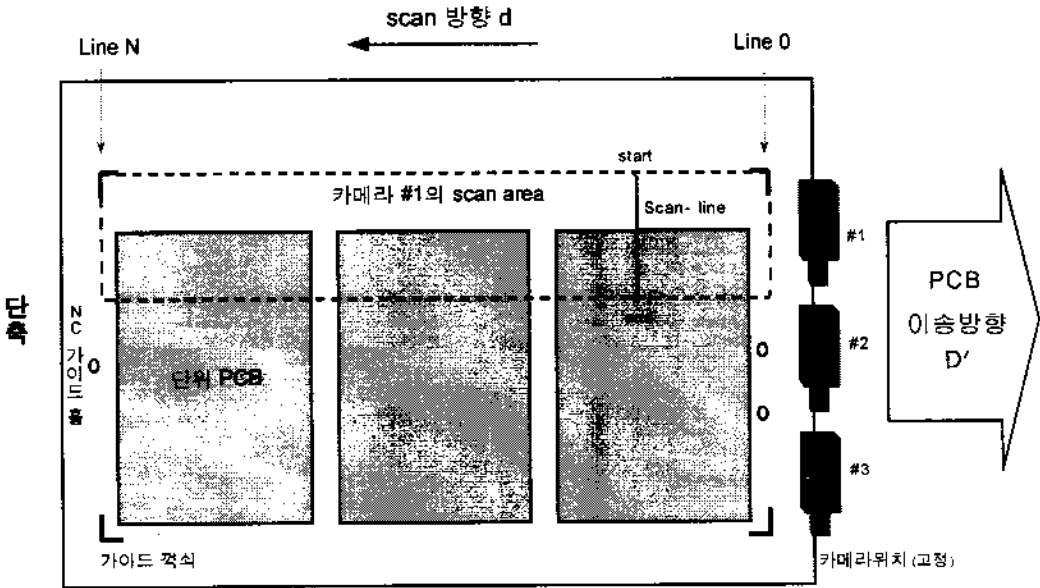
(4) 카메라별 스캔 영역을 계산한다.

스캔 영역은 장축 방향으로 오른쪽과 왼쪽 가이드 객체 내부이고, 단축 방향으로 카메라의 스캔 용량에 따라 결정된다. 예를 들어 카메라의 라인 스캔 용량이 4096 pixel이고, 정밀도가 0.2mm인 경우에는 최대 $4096 \times 0.2\text{mm} = 81.92\text{mm}$ 까지 허용된다. 카메라별로 중복되는 영역이 존재할 수 있다.

(5) 카메라별 MRLC를 생성한다.

한 개의 라인 이미지를 스캔하여 각 RUN에 대한 다음과 같은 MRLC 정보를 생성하게 된다.

a. RUN의 시작 위치



PCB panel 감촉

〈그림 10〉 PCB SCAN

- b. 패턴의 종류를 나타내며 라인 또는 공백
 - c. 주변 이미지와의 위치 관계를 알려주며 수평 또는 수직
 - d. RUN의 홀 이미지 여부
 - e. 해당 RUN의 스캔 라인에서의 마지막 패턴 여부
- 위의 정보를 스캔 영역내 존재하는 모든 라인에 대해 이를 반복하여 생성한다.

4.4 MRLC 조정

PCB 검사 작업에 소요되는 시간은 검사 대상 후보의 수, 즉 결함이 있을 것이라 예상되는 경우의 수에 비례할 것이다. 이것은 MRLC내 라인 패턴과 수평 space 패턴의 수에 직접적인 영향을 받게 되어 있다. 이러한 이유로 검사 시 효율적인 MRLC 정보를 사용할 수 있도록 검사 정보 생성 시스템은 MRLC 생성 시 적용되는 다음 네 가지 요인들을 작업자가 조정해 가면서 MRLC 결과를 비교해 볼 수 있다. 이들 요인은 실험에 의하여 밝혀진 것으로 여러 요인 중에서 검사에 영향을 줄 수 있고 이들에 의하여 실시간의 검사 및 효율성의 증대를 기대할 수 있는 요인을 추출한 것이다.

(1) SPACE 패턴 폭

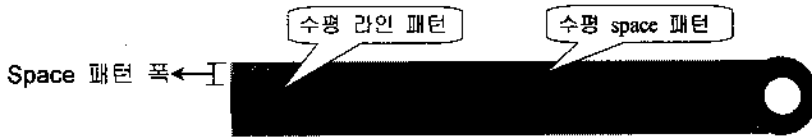
수평 space 패턴을 결정하는 기준폭으로 〈그림 11〉과 같이 수평 라인 패턴으로 부터의 거리를 mm 단위로 정해준다.

(2) 폭 결정시 검사 횟수

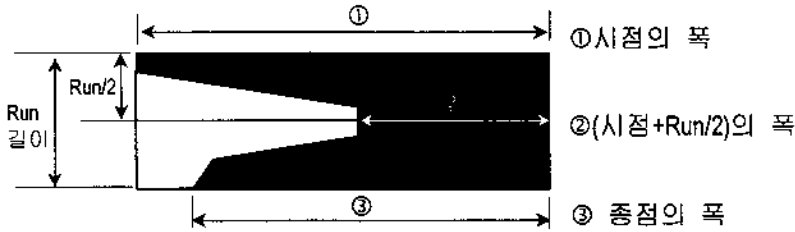
RUN의 수평 여부 판단은 패턴의 폭과 RUN 길이의 비율을 비교해 보아야 한다. 이 때 사용하는 폭은 해당 RUN의 최소 폭이며, 이를 구하기 위한 계산 횟수를 의미한다. 기본값은 2 회이며 RUN의 시작 위치와 종료 위치에서의 폭 중에서 작은 값을 사용하도록 되어 있다. 2회를 초과하는 경우에는 시점과 종점 사이를 등분한 위치에서의 폭을 이용한다. 테스트에 의하면 검사 횟수와 MRLC 생성시 소요되는 시간이 비례하며, 5회 이상인 경우는 결과에 큰 영향을 주지 않는다. 이상적인 계산 횟수는 3회이다. 예를 들어 〈그림 12〉와 같은 이미지에서 폭 계산횟수가 3회인 경우 패턴의 폭은 최소폭인 ②이다.

(3) 패턴의 폭과 RUN 길이 비율

패턴의 폭이 결정되면 폭과 RUN의 길이를 비교하여 수평 패턴인가 아닌가를 결정하여야 한다. 이 비율은 해당 패턴의 폭을 1로 보았을 때, 상대적인 RUN의 길이를 나타내는 값이다. 이 값이 작을수록 짧은 RUN도 수평 패턴으로 분류될 가능성이 높아진다.



〈그림 11〉 수평 space 패턴



〈그림 12〉 패턴의 폭 결정

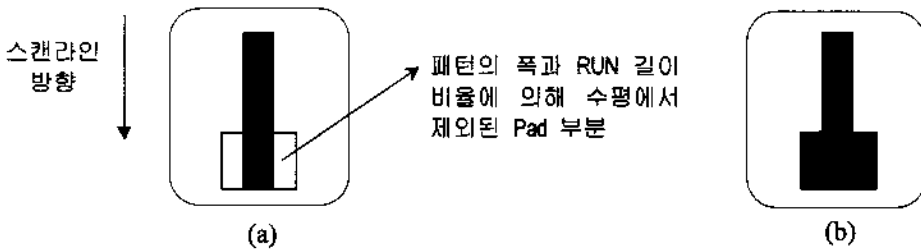
(4) MRLC 보정 작업

MRLC 생성 후 이미지를 재검토하여 RUN의 방향, 특히 수평 RUN을 보정해줄 것인지의 여부를 지정한다. 보정작업의 목적은 라인패턴의 양 끝부분에 존재하는 패턴들이 라인패턴과 동일한 방향을 갖는 연속 패턴으로써 인식되도록 하기 위한 것이다. 따라서 보정작업은 수평 라인 패턴을 초과하지 않으면서 시점 또는 종점 부분에서 좌/우 방향으로 연속되는 라인 패턴들의 방향을 수평으로 조정해주는 것을 의미한다. 이 작업은 패턴의 폭과 RUN 길이 비율이 큰 경우에는 수행하는 것이 좋다. 보정 작업을 통해서 〈그림13〉의 (a)와 같이 수평 라인 패턴에서 제외되는 부분이 (b)와 같이 수정된다.

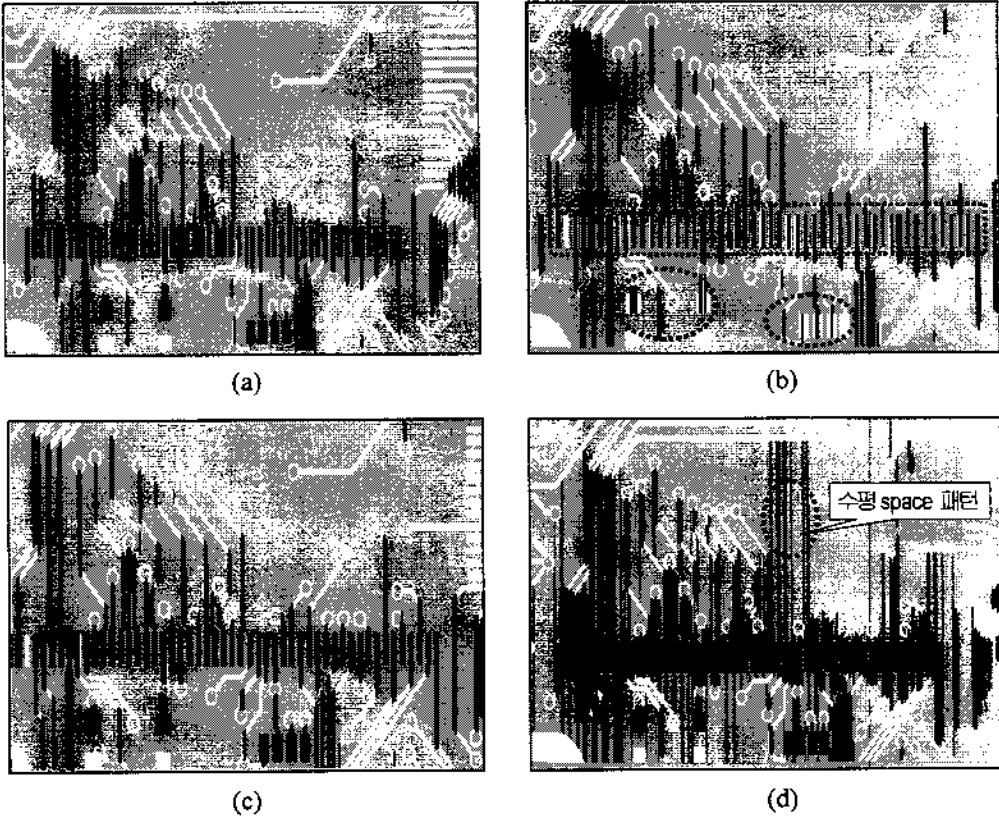
〈그림 14〉는 언급한 MRLC 관련 변수들의 조정에 따른 MRLC 결과를 보여준다. RUN의 방향성은 스캔라인 방향을 기준으로 하기 때문에 여기서는 세로방향의 수평이 된다. 그

림에서 흰색 부분은 라인 패턴, 검정색 부분은 수평라인 그리고 옅은 회색으로 표현된 부분은 수평 space 패턴을 나타낸다.

〈그림 14〉의 (a)와 같이 패턴의 폭과 RUN 길이 비율이 1인 경우에는 육안으로 수평이라고 생각되는 이미지는 거의 수평으로 분류된다. 그러나 패턴의 폭과 RUN 길이 비율을 크게 하면 (a)에서 수평이었던 이미지 중에서 Pad 부분에 있는 짧은 RUN들은 수직으로 바뀐다. (b)에서 점선영역 내에서 그 예를 볼 수 있다. 그러나 Pad 부분은 검사의 용이성을 위해서 이와 연결된 라인 이미지와 동일한 방향을 갖도록 해야 하기 때문에 이러한 RUN들은 유효하지 않은 정보가 된다. 따라서 이를 보완하기 위해서 별도의 작업이 필요하다. 'MRLC 보정작업'이 이러한 기능을 수행한다. (c)는 MRLC 보정작업을 수행한 결과로써 (b)의 점선영역 내 존재하는 pad부분이 수정되었음을 알 수 있다. SPACE 패턴 폭을 정해주면



〈그림 13〉 MRLC 보정



〈그림 14〉 MRLC 결과 비교

수평 라인패턴을 기준으로 일정 영역만큼의 공백을 수평 space 패턴으로 인식한다고 하였다. (d)에서 원안의 옅은 회색 부분이 수평 space 패턴을 나타낸다.

5. 적용 사례

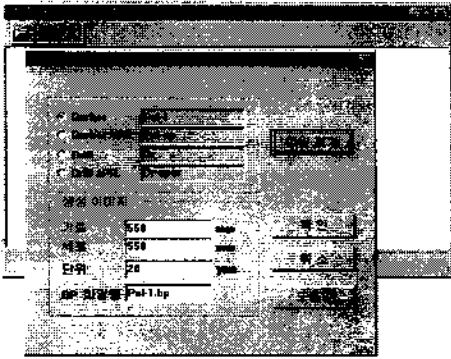
본 논문은 L사의 PCB 검사시스템에 포함된 검사정보 생성 시스템에 대한 내용이다. 본 시스템은 Windows 95를 OS로 하는 시스템에서 사용될 수 있도록 개발되었다.

〈그림 15〉는 검사정보 생성 시스템의 주요 기능인 비트맵 이미지 생성 부분과 MRLC정보 생성 부분에 대한 실행 과정을 보여준다. (a)는 비트맵 이미지로 변환할 Gerber정보와 이미지의 정밀도 등의 요구 조건들을 지정해주는 단계이다. (b), (c),(d)는 완성된 비트맵 이미지를 확인하는 과정으로써 (b)는 가이드궤쇠를 포함한 전체 판넬 이미지이고, (c)와 (d)는 이것

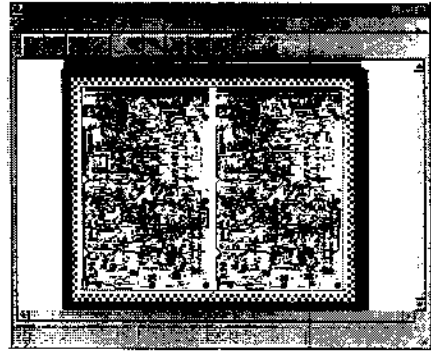
을 부분 확대한 결과이다. (c)에서 회색으로 표시된 원은 홀 이미지이다. (e)와 (f)는 MRLC정보를 만드는 방법을 결정하는 단계로써 검사에 유용한 최적의 MRLC를 얻기 위한 조건들을 선택해 주어야 한다. 〈그림 15〉의 (g)는 한 개의 카메라가 검사할 영역내 이미지에 대한 MRLC정보이고, 이를 부분 확대한 결과가 (h)이다. 화면상에 표현된 색의 구별로 패턴, 방향 등의 MRLC 정보를 쉽게 확인할 수 있다.

6. 결론

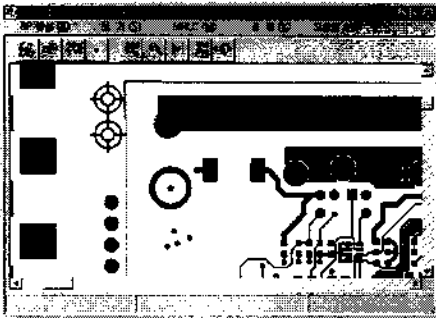
본 논문은 PCB 검사 공정에서 기준정보로 사용되는 MRLC 형식의 검사정보를 생성해주는 시스템에 대해서 설명하였다. CAD에서 설계된 판넬 이미지로부터 중간 확인용 비트맵 이미지와 검사용 MRLC 정보를 생성하는 일련의 과정에서 요구되는 조건선택과 기능수행 과정에 관한 부분을 구현하기 위한



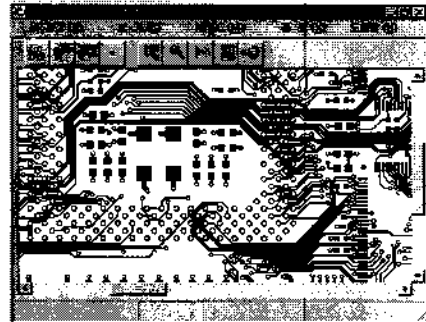
(a)



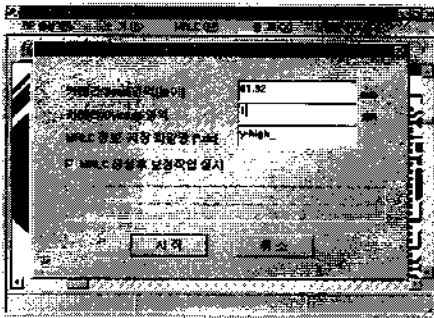
(b)



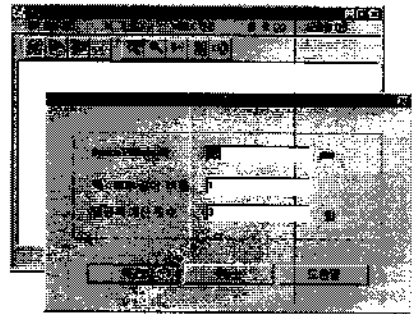
(c)



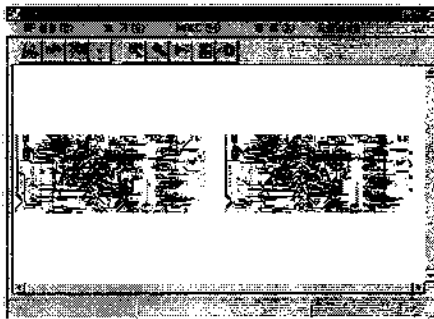
(d)



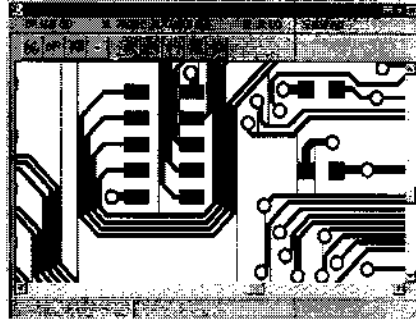
(e)



(f)



(g)



(h)

〈그림 15〉 검사정보 생성 시스템 수행 과정

아이디어를 설명하였다. 또한 Gerber 파일을 비트맵으로 변환하기 위한 아이디어를 자세히 소개하였고, 검사에 쓰이는 MRLC 정보의 생성방법을 설명하였다. 특히 주어진 조건에 따라서 MRLC가 어떻게 변화하는가에 대하여도 실험적으로 규명하도록 노력하였다.

[참 고 문 헌]

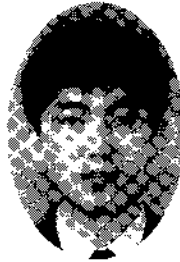
[1] 월간 전자기술, Transistor Special 2.3, pp. 33-44, 1995년 12월호.

[2] Katsumi Wasaki, Yasushi Fuwa, Masayoshi Eguchi, and Yatsuka Nakamura, "Logical Coloured Petri Net Expanded to be Suitable for Making the Control System Model", ICARCV : INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION'96, Vol.2 of 3, pp. 708-713, 1996.11.

[3] K.Wasaki, Y.Fuwa, M.Eguchi, and Y.Nakamura : "Extended Petri Nets for Control System Software", Technical Report of IEICE, COMP93-12, SS93-6, pp. 37-44, 1993.

[4] P.N.Kawamoto, Y.Fuwa, and Y.Nakamura : "Basic Concepts for Petri Nets with Boolean Markings", Journal of Formalized Mathematics, Vol.4, No.1, 1993.

[5] <http://www.minnitron.co.uk/data.html>, Data formats explained.



이철수

한양대학교 산업공학과를 졸업하고 한국과학기술원 산업공학과에서 석사와 박사학위를 받았다. 현재 전남대학교 산업공학과 부교수로 재직 중이다. 관심분야는 CAD/CAM과 CNC 콘트롤러이다.



고은희

전남대학교 산업공학과에서 학사와 석사 학위를 받았고, 현재 전남대학교 자동차 연구소에 재직중이다.