비젼 시스템에서 신경 회로망을 이용한 검사 영역에 관한 연구

A Study on Inspection Area using Neural Network for Vision Systems

오 제 휘, 차 영 엽
(Je-Hui Oh and Young-Youp Cha)

Abstract : A FOV, that stands for "Field Of View", refers to the maximum area where a camera could be wholly seen. If a FOV of CCD camera cannot the cover overall inspection area, the overall inspection area should be divided into sub-areas of size FOV. In this paper, we propose a new neural network-based FOV generation method by using a newly modified self-organizing map(SOM) which has multiple structure based on a self-organizing map, and uses new training rule that is composed of the movement, creation and deletion terms. Then, experimental results using real PCB indicate the superiority of the method developed in this study to the existing sequential method.

Keywords : vision system, neural network, solder joint inspection, self-organizing map

I. 서론

오늘날 전자 제품의 소형화와 다기능화, 그리고 고성능화가 요구됨에 따라, 전자 제품 제작에 이용하는 SMD (Surface Mount Device) 실장기술은 수요자의 요구를 충족시키기 위해 소형화 그리고 고밀도화 되어 가고 있다. 그런데 이러한 부품제작과 실장기술의 급속한 발전에 반하여 검사기술은 기존의 인간에 의한 목시검사에 의존하는 실정이다. 기존의 목시검사는 소형화, 고밀도화 된 SMD 검사시 작업자 눈의 피로 가중, 집중력 저하, 작업자에 따른 판정기준 상이, 작업자의 숙련도와 컨디션에 따른 품질의 산포 다양, 검사 결과의 빠른 feedback 불가, 품질의 신뢰도 하락, 인건비 상승 등의 문제점을 가지게 되었다. 따라서 이러한 단점을 극복하기 위하여 부품의 유무와 납땜 상태를 자동으로 검사하는 시스템들이 등장하기 시작했으며, 그 중에서 컴퓨터 비젼 시스템(computer vision system)을 이용한 검사장비는 좋은 결과를 나타내고 있다[1].

PCB의 납땜 상태 검사를 위한 일반적인 비젼 시스템의 구성은 Host PC부, 영상 처리부, 구동부로 이루어지며, Host PC부는 주변 장치를 통제하고 검사 알고리즘을 구현하는 부분이며, 영상 처리부는 화상처리 보드를 사용하여 CCD 카메라를 통해 들어오는 영상을 검사 알고리즘에 이용할 수 있도록 처리해주는 부분이다, 마지막으로 구동부는 검사 시스템이 자동으로 동작하기 위한 영역이동, 카메라 조정 등의 작업을 하는 부분으로서, 이러한 시스템을 이용한 간략한 검사 흐름도는 그림 1과 같이 나타낼 수 있다. 이와 같은 컴퓨터 비젼 시스템에서 하드웨어(구동부, 영상 처리부)와 함께 가장 중요한 부분은 검사 알고리즘이라고 할 수가 있다. 따라서 시스템 구성과 검사 알고리즘에 관한 연구는 폭넓게 진행되어져 왔으며[1], 최근에는 신경 회로망을 이용한 PCB의 납땜 상태 검사 알고리즘도 발표되었다[2]. 하지만 그림 1에서 보듯이 검사 알고리즘 만큼 중요한 FOV(Field of View)에 관한 연구는 거의 전무하다.

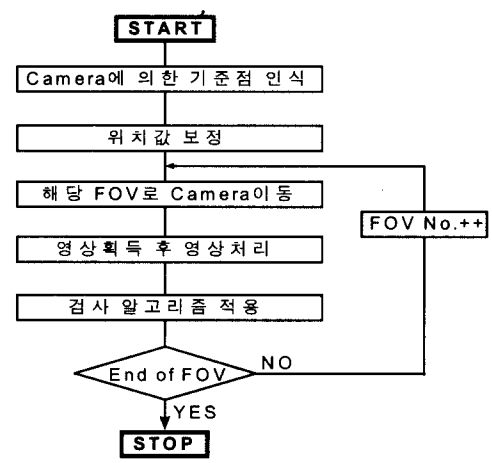


그림 1. 컴퓨터 비젼 시스템의 흐름도.
Fig. 1. Flowchart of computer vision system.

FOV란 임의의 구동부 좌표에서 카메라가 볼 수 있는 최대한의 범위를 나타낸다. FOV 생성과 이용을 위하여 선행되어야 할 사항으로는 FOV의 크기 결정과 생성된 FOV의 검사 순서 결정이 있다. FOV의 크기 결정을 위해서는 카메라의 해상도(resolution)에 따른 실제 거리값과 영상 거리값에 대한 보정(calibration)작업[3]으로 결정되며, 생성된 FOV의 검사 순서 결정은 최소 거리의 결정을 위해 신경 회로망의 Hopfield 모델을 이용한 알고리즘[4]을 사용하여 결정할 수 있다.

본 연구는 FOV의 중요성을 인식하여 위의 선행 조건들이 행해졌을 경우, CCD 카메라의 FOV를 최소 개수로

접수일자 : 1997. 11. 19., 수정완료 : 1998. 5. 2.
오제휘, 차영엽 : 원광대학교 기계공학부

생성하는 알고리즘 개발에 목적을 두고 있다. 실제 검사 시스템에서는 특정 PCB의 FOV의 크기는 일정하게 정해진다. 그러므로, 그 PCB에서 FOV의 중첩을 최소화하고 FOV의 개수와 그 사이의 거리를 최적화 한다면, 총 부품 검사 시간이 줄어들게 할 수 있을 것이며, 이러한 효과는 검사 시스템의 검사 시간을 단축시켜 생산성 향상을 가능하게 한다. 이러한 최적의 FOV 생성 알고리즘 개발을 위하여 신경 회로망 중에서 패턴 분류에 많이 사용되는 self-organizing map을 기본으로 한 네트워크를 제안한다. 제안된 네트워크는 다중 구조의 형태를 가졌으며, 이동·생성·삭제항 등의 부차적 항이 있는 새로운 트레이닝 법칙을 사용한다

신경 회로망을 이용한 새로운 FOV 생성 알고리즘의 설명을 위하여, 먼저 II장에서는 기존의 시스템에 대부분 사용되고있는 순차적인 FOV 생성 알고리즘과 이 알고리즘의 취약점을 설명한다. III장에서는 IV장에서 제안되는 알고리즘의 이론적 기초가 되는 Kohonen의 self-organizing map을 간단히 설명하고 IV장에서는 본 연구에서 새로이 제안되는 FOV 생성 알고리즘에 대한 이론과 새로운 트레이닝 법칙을 설명한다. 그리고, V장에서는 실제 PCB에 적용하여 제안된 알고리즘의 유용성을 증명한다.

II. 순차적인 FOV 생성 알고리즘

기존의 FOV 생성 알고리즘의 대표적인 예는 순차적인 방법을 통한 FOV 생성이다[5]. 이 알고리즘은 이전에 생성된 FOV에 포함되지 않는 부품들 중 가장 왼쪽 부품의 x축의 좌표를 새로이 생성하는 FOV의 x축 시작 좌표로 하여, FOV x축의 길이만큼 위치를 설정하고, y축에서도 마찬가지로 이전에 생성된 FOV에 포함되지 않은 부품들 중 가장 아래쪽 부품의 y축의 좌표를 새로이 생성하는 FOV의 y축 시작 좌표로 하여, FOV y축의 길이만큼 위치를 설정하여 하나의 FOV를 만든다. 그리고 이 FOV에 포함된 부품을 다음 FOV 생성에는 삭제시킨다. 그리고 삭제된 부품을 제외한 나머지 부품을 이용하여 위의 방법과 같은 작업을 순차적으로 수행한다(아래->위->아래). 이러한 작업은 남는 부품이 없을 때까지 행함으로서 FOV 생성은 종료된다. 그림 2는 순차적인 FOV 생성 순서도를 나타낸다.

이러한 방법은 비록 모든 부품을 FOV에 포함시킬 수는 있겠지만, 검사 부품이 많을수록 야기되는 생성시간의 과중성과 FOV들간의 중첩에 따른 FOV의 과다성 등의 단점을 가지고 있다. 특히 FOV 수의 과다성은 그림 3에서와 같이 각 FOV간의 많은 중첩과 공간의 비효율적 사용에 의해 발생되며, 결국 PCB검사의 총 검사 시간이 길어지는 결과를 낳게 한다. 그림 3(a)는 기존의 순차적인 FOV 생성 알고리즘을 사용했을 경우의 결과를 예시하여 나타낸 것이며, 그림 3(b)는 (a)와 같은 부품의 위치를 가질 때 얻을 수 있는 최적화된 FOV를 나타낸다. 여기서 검게 칠한 작은 사각형은 부품을 나타내며, 커다란 사각형은 CCD 카메라의 FOV를 나타낸다. 그림에서 (a)는 (b)보다 FOV간의 중첩과 개수가 많아서 검사시간과 구동부의 이동횟수 관점에서 안 좋은 상태이다.

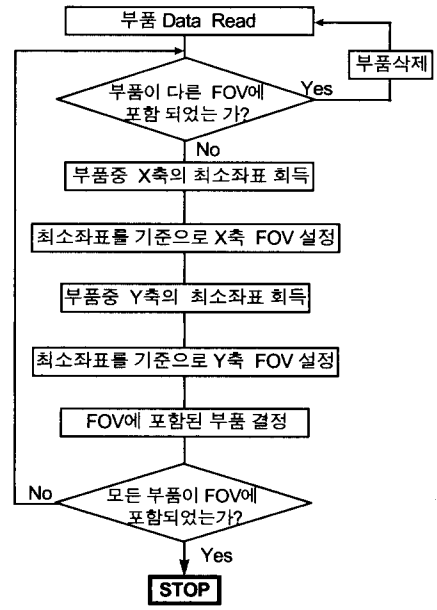
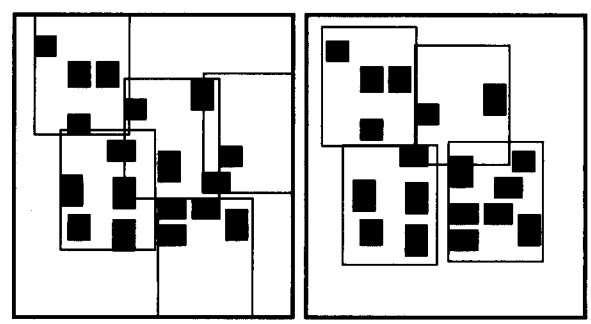


그림 2. 순차적인 FOV 생성 알고리즘.
Fig. 2. Algorithm of sequential FOV generation.



(a) (b)

그림 3. 순차적 FOV(a) 와 최적화 된 FOV(b)의 비교.
Fig. 3. Comparison of sequential FOV(a) with optimal FOV(b).

III. Kohenen's self-organizing map

네트워크 구조 및 신호 처리 방법에 의해 신경 회로망을 크게 3가지로 나눌 수 가 있다. 첫 번째는 입력 신호들을 임의의 출력 신호로 변형시키는 feedforward network이며, 두 번째는 초기의 입력값이 feedback 시스템에 의하여 점근적으로 최종 상태에 접근하면서 출력값을 찾아내는 feedback network이다. 그리고, 마지막으로 측면의 영향력에 의해 스스로 작업을 완료하거나, 서로 다른 신호들의 분류가 가능한 self-organizing map이 있다[6]. 본 연구는 FOV와 부품이 거리 단위로 표시되고, 부품의 좌표를 입력값으로 하여 이를 분류하고, 그 출력을

FOV의 중심좌표로 얻는 작업이 필요하므로 self-organizing map의 네트워크를 사용한다. 그 중에서도 튜보 코호넨(Tuevo Kohonen)에 의해 개발된 Kohonen의 Self-Organizing Map(이하 KSOM)을 기본으로 하였다[6].

self-organizing map 네트워크에서는 기대된 출력이 없다. 대신에 신경 회로망은 자기 조직화 특성에 의해 임의의 추상적인 관계를 추론할 수 있으며, 더 많은 입력이 인가 될수록 네트워크는 그 학습을 개선하고 변화된 입력들에 적응하여 출력을 내보낸다. 이러한 구조의 한 이점으로는 변화하는 상태와 입력에 대해 대처할 수 있다는 것이다. 그러므로, 이 네트워크는 입력을 다른 카테고리(category)로 분류할 때와 음성 인식, 로봇 모터 제어 등에 사용된다.

KSOM의 목적은 N-차원의 입력 공간을 의미있는 지형학적인 순서로 1차원 또는 2차원의 출력 공간(출력 뉴런)에 매핑할 수 있게 하는 것이다. 이러한 목적을 성취하기 위해 경쟁학습(competitive learning)에 의한 승자독점(winner-take-all)원리와 측면 제어(lateral inhibition)가 이용된다[7].

$$\text{입력 벡터 } X\text{를 } X=[x_1, x_2, \ldots x_p]^T \qquad (1)$$

로 하고, j번째 출력층 뉴런과 상응하는 가중치 벡터 W_j를

$$W_j=[\omega_{j1}, \omega_{j2}, \ldots, \omega_{jp}]^T, \quad j=1,2,..,N \qquad (2)$$

로 표시한다면, 입력 벡터 X와 가중치 벡터 W_j를 이용한 트레이닝 법칙은 다음과 같다.

$$W_j^{new} = W_j^{old} + \eta(X_i - W_j^{old}) \qquad (3)$$

여기서 η는 학습율(learning rate)을 나타내고, i는 1에서 p까지로 입력 뉴런(neuron)의 수를 나타내며, j는 1에서 N까지로 출력 뉴런의 수이다. 그리고 w_{ji}는 i번째 입력 뉴런과 j번째 출력 뉴런을 연결하는 가중치를 나타낸다. 입력 벡터의 분류를 위한 출력 층의 승자 뉴런의 결정은 입력값 X와 가장 비슷한 가중치 W_j를 갖는 출력 뉴런을 선택하는 것과 같다. 이러한 출력 뉴런을 선택하는 방법은 두 가지가 있다. 첫째는

$$I_{j\,\max} = \sum_{i=1}^{p} \omega_{ji}\, x_i \qquad (4)$$

과 같은 I_j를 선택하는 것이고, 두 번째는 입력 벡터와 최소의 Euclidean norm을 갖는 가중치를 선택하는 것이다. 즉, $i(X)$가 승자 뉴런이라 한다면 이 식은 다음과 같다.

$$i(X)=k, \quad where \ ||W_k - X|| < ||W_j - X|| \qquad (5)$$

위와 같이 출력 뉴런을 선택하여 승자가 된 뉴런만이 "1"이 되고 나머지는 "0"이 되는데 이러한 방법이 경쟁학습에 의한 승자독점 원리이다. 보통 가중치 벡터와 입력 벡터는 정규화(normalization)-가중치 벡터와 입력 벡터의 최대 크기가 "1"로 설정됨-시키는데 그 이유는 트레이닝 규칙이 입력 벡터로부터 가중치 벡터를 뺀 값을 사용하기 때문이다. 그리고, 좀더 효율적인 패턴 분류를 위하여 측면 제어를 이용한다. 측면 제어의 대표적인 방법은 이웃관계 함수(neighborhood function) $\Lambda_{i(x)}(n)$ 의 사용으로, 승자 뉴런이 선택되면 승자 뉴런의 이웃하는 거리에 따라 연결강도를 달리하는 방법으로 연결 강도는 거리에 반비례한다. 이웃관계 함수를 이용한 트레이닝 법칙은 다음과 같다.

$$W_j(n+1)=\begin{cases} W_j(n)+\eta(n)[X-W_j(n)], & j\in\Lambda_{i(X)}(n) \\ W_j(n), & otherwise \end{cases} \qquad (6)$$

여기서 $\eta(n)$은 n 시간에서의 학습율이다.

IV. 신경 회로망을 이용한 FOV 생성 알고리즘

본 연구에서는 위의 KSOM을 기본으로 하여 새로운 FOV 생성 알고리즘을 제안한다. 우선, FOV 생성을 위해 변수들에 대한 정의를 내린 후, 새로운 FOV 생성 알고리즘에 대하여 설명한다.

1. FOV의 변수

FOV와 부품과의 관계를 이해하기 위하여 그림 4는 변수들을 보여주고 있다. 부품에 관한 변수는 ($C_{x,\min}$, $C_{y,\min}$), ($C_{x,\max}$, $C_{y,\max}$)로서 각각 최소점의 좌표와 최대점의 좌표를 나타낸다. 그리고 FOV에 관한 변수는($F_{x,\min}$, $F_{y,\min}$), ($F_{x,\max}$, $F_{y,\max}$), ($F_{x,cen}$, $F_{y,cen}$), Fov_x, Fov_y로서, 각각 FOV의 최소점 좌표, 최대점 좌표, 중심점 좌표, X방향의 길이, Y방향의 길이들을 나타내며 이 FOV 변수들의 상호 관계는 다음과 같다.

$$\begin{aligned} F_{x,\min} &= F_{x,cen} - \frac{Fov_x}{2} \\ F_{y,\min} &= F_{y,cen} - \frac{Fov_y}{2} \\ F_{x,\max} &= F_{x,cen} + \frac{Fov_x}{2} \\ F_{y,\max} &= F_{y,cen} + \frac{Fov_y}{2} \end{aligned} \qquad (7)$$

2. FOV 생성 알고리즘

위의 FOV의 변수에서와 같이 FOV의 크기는 이미 정해진다. 그러나 고유의 KSOM에는 거리의 제약을 줄 수 없으며, 출력 뉴런의 수가 변할 수 없는 커다란 단점을 갖는다. 그래서 본 연구에서는 고유의 KSOM을 변형하고, 새로운 항들을 만들어 사용하였다.

실제 이용하는 입력 벡터를 부품의 변수 $C_{x,\min}$, $C_{y,\min}$, $C_{x,\max}$, $C_{y,\max}$ 4개를 사용하였고, 가중치 벡터(출력 뉴런과 동일)로는 FOV의 중심 좌표 ($F_{x,cen}$, $F_{y,cen}$)를 사용하여 유동적인 출력 뉴런 수를 갖는 2차원의 Map으로 하였다.

입력 벡터가 2차원의 공간을 가지고, 부품에 따라 크기가 다르기 때문에 한 개의 KSOM만을 사용한다면, 원하는 출력값을 얻기는 힘들 것이다. 그러므로 KSOM의 수를 2개로 하는 다중의 네트워크를 사용하였다.

즉, 입력벡터는 $C_{x,\min}$, $C_{y,\min}$으로 하고 가중치 벡터

는 $F_{x,cen}$, $F_{y,cen}$인 SOM1과 입력 벡터는 $C_{x,\max}$, $C_{y,\max}$으로 하고 가중치 벡터는 $F_{x,cen}$, $F_{y,cen}$인 SOM2로 하였다.

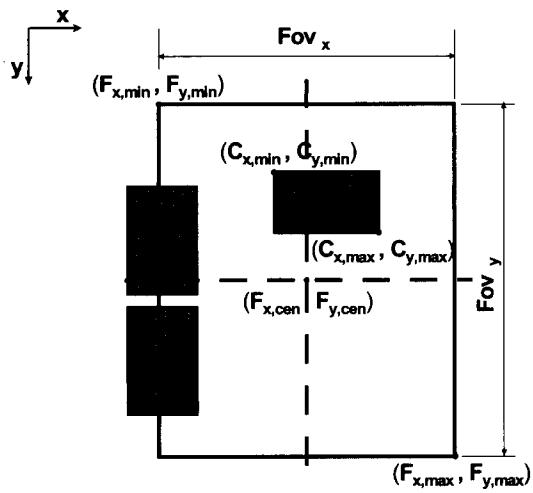


그림 4. FOV의 변수.
Fig. 4. Parameters of FOV.

(5)에 의하여 얻어지는 SOM1과 SOM2의 승자 뉴런을 각각 $i(X)_1$, $i(X)_2$로 한다면, 트레이닝 법칙을 다중 네트워크에 맞추어서 사용해야 한다. SOM1과 SOM2의 승자 뉴런이 같다면 이 부품은 같은 가중치(즉, FOV)에 속하므로 아무런 문제가 없다. 하지만 SOM1과 SOM2의 승자 뉴런이 다르다면 이 부품은 서로 다른 가중치 사이에 걸쳐있다고 볼 수가 있다. 여기서, 거리의 관계에 대한 고려가 없다는 KSOM의 단점을 보완하기 위하여 가중치가 일정 방향성을 갖도록 설계한다. 즉, 승자뉴런이 다른 경우 SOM2의 승자뉴런을 부품의 최소좌표 방향으로 이동하는 것이다. 이러한 설명을 수학적으로 표현하면 다음 식과 같다.

$$W_{j,2}(n+1)=\begin{cases} W_{j,2}(n)+\eta_1(n)[X_2-W_{j,2}(n)], & i(X)_1=i(X)_2 \\ W_{j,2}(n)+\eta_2(n)[X_1-W_{j,2}(n)], & otherwise \end{cases} \quad (8)$$

여기서, 하첨자 1, 2는 SOM1, SOM2를 나타내는 것이고, $\eta_1(n)$, $\eta_2(n)$ 는 학습율을 나타내는 것으로 SOM1과 SOM2의 승자 뉴런이 다른 경우가 같은 경우보다 크게 설정 해야한다($\eta_1(n)<\eta_2(n)$). 그리고 시간이 경과할수록 그 값들이 줄어들어야 한다. 위 식만 사용하면, 일정 방향성에 의한 FOV간의 필요 없는 중첩과 FOV의 정해진 크기에 의한 자신의 FOV가 없는 부품이 생길 수 있다. 이를 위하여 다음과 같은 가중치 벡터의 이동과 삭제, 그리고 생성이 가능한 부수적인 항을 만들어 새로 제안한 (8)의 트레이닝 법칙을 보완하겠다.

이동(Movement) 항 : 이전에 생성된 어떠한 FOV에도 포함되지 않은 부품 l 이 가장 인접한 임의의 FOV m 에 포함되기 위한 FOV의 이동식은

$$W_{m,2}(n+1)=\begin{cases} W_{m,2}(n)+\eta_3(n)[X_{l,1}-W_{m,2}(n)], & \delta \geq Gap_m \\ W_{m,2}(n), & otherwise \end{cases} \quad (9)$$

이고, 여기서 δ 는 이동 한계 변수로 FOV의 이동이 가능한 영역을 나타낸다. 그리고 Gap_m 은 최소 부품 거리항으로 부품 l 과 가장 가까운 임의의 FOV m 의 거리를 나타내며, 식은 다음과 같다.

$$Gap_m=\min||W_m(n)-X_{l,1}|| \quad (10)$$

생성(Creation) 항 : 전에 생성된 어떠한 FOV에도 포함되지 않은 부품이며, 이동 한계 변수 δ 를 초과하여 이동항을 적용할 수 없는 부품 l 이 있는 경우, 이 부품을 포함시키는 FOV를 생성 해야하며, 그 식은 다음과 같다.

$$W_{new}(n+1)=\alpha\cdot X_{l,1}, \quad where\ \delta<Gap_m \quad (11)$$

여기서, α는 FOV 생성 거리 변수로써 $0<\alpha\leq 1$ 부품 l을 포함시킬 FOV의 위치를 결정해주며, 사이에 있어야 한다.

삭제(Deletion) 항 : FOV간의 중첩을 최소화하기 위해, 임의의 면적이상 중첩이 된다면 그 FOV r 를 삭제하기 위한 식으로 다음과 같다.

$$W_r(n+1)=\begin{cases} W_r(n), & \phi\leq Dist_r \\ \infty, & otherwise \end{cases} \quad (12)$$

여기서 ϕ 는 삭제 경계 변수로 삭제를 위한 경계를 나타낸다. 그리고 $Dist_r$ 은 FOV의 최소 주변 거리항으로 임의의 FOV r 이 주변 FOV와의 거리를 나타내며, 식은 다음과 같다.

$$Dist_r=\min||W_r(n)-W_i(n)|| \quad (13)$$

위의 세 가지 항들을 추가하여 모든 부품이 FOV에 포함될 때까지 알고리즘을 실행시키면 된다. 이러한 과정을 정리하면 다음과 같다.

Step 0 : 입력 값 $C_{x,\min}$, $C_{y,\min}$, $C_{x,\max}$, $C_{y,\max}$을 읽고, 초기 FOV 개수를 설정한다. 그리고, 각각의 변수인 $\eta_1(n)$, $\eta_2(n)$, $\eta_3(n)$, δ, α, ϕ를 정한다.

Step 1 : 초기 가중치 벡터(FOV의 중심좌표)의 배열을 위한 최소 부품 위치를 찾은 후, FOV를 개수만큼 2차원의 그물형으로 배열한다.

Step 2 : $C_{x,\min}$, $C_{y,\min}$를 입력 값으로 하는 SOM1과 $C_{x,\max}$, $C_{y,\max}$를 입력 값으로 하는 SOM2에서 승자뉴런 $i(X)_1$, $i(X)_2$ 을 찾기 위해 (5)를 이용한다.

Step 3 : 승자뉴런 $i(X)_1$, $i(X)_2$ 을 이용하여, 트레이닝식-(8)에 적용한다.

Step 4 : Step 4.1과 4.2 그리고 4.3을 적절한 시간 간격을 주고 실행한다. 실제로는 Step 4.1이 Step 4.2보다,

Step 4.2는 Step 4.3보다 실행 빈도가 많아야 한다.

- Step 4.1 : 학습율 $\eta_1(n)$, $\eta_2(n)$를 감소시킨다.
- Step 4.2 : 이동항(9)과 생성항(11)을 실행시킨다.
- Step 4.3 : 삭제항(12)을 실행시킨다.

Step 5 : 모든 부품이 자신의 FOV를 갖는 경우 정지하고, 그렇지 않은 경우 Step 2로 간다.

Step 6 : 부품이 존재하는 FOV만 남기고 나머지 FOV는 삭제한다.

V. 실험 결과

실제 실험을 위해, FOV 생성 알고리즘을 적용한 컴퓨터 비젼 시스템은 그림 5와 같고, FOV 생성에 이용된 시편과 그 사양은 그림 6과 표 1에서와 같다. 이 시편을 이용하여, 순차적인 FOV 생성 알고리즘과 본 논문에서 제안한 신경 회로망을 이용한 FOV 생성 알고리즘을 각각 실행하였다.

그림 5. 컴퓨터 비젼 시스템.
Fig. 5. Computer vision system.

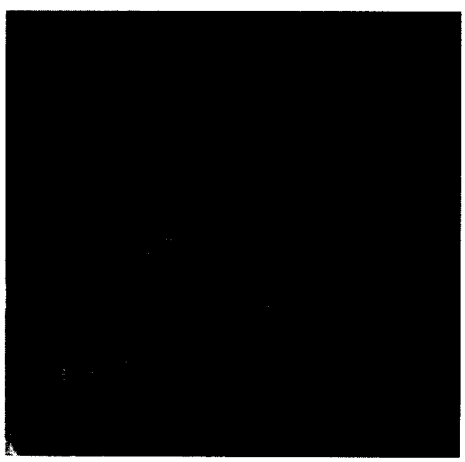


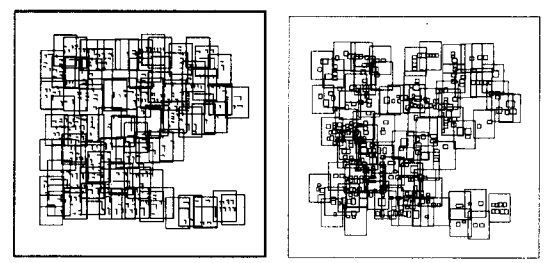
그림 6. 실험 시편.
Fig. 6. Experimental sample.

제안된 신경 회로망을 이용한 FOV 생성 알고리즘을 이용하기 위하여, 입력 값은 랜덤하게 하여 실행했으며, 각 변수의 값은 $\eta_1=0.4$, $\eta_2=0.9$ 로 하여 300회마다 0.9씩 감소시켰다. $\eta_3=0.8$ 로 하였고, $\delta=5(mm)$, $\alpha = FOV/2$, $\phi = FOV/2$ 또는 $FOV/3$ 으로 하였다. 그리고 이동항과 생성항은 900회마다, 삭제항은 1200회마다 실행하였다.

표 1. 검사 시편 사양.
Table 1. The specifications of inspection sample.

사 양	규 격
PCB 규격	124×120 (mm)
카메라 FOV 크기	10.6×14.0 (mm)
PCB 장착 부품 수	394 (개)

위와 같이 설정했을 때, 기존의 순차적인 FOV 생성 알고리즘을 이용한 경우의 FOV 생성 개수는 88개이었으며, 신경 회로망을 이용한 FOV 생성 알고리즘을 이용한 경우에는 85-79개 사이로 생성되었으며 평균 82개였다. 그림 7은 순차적 방법과 수정된 SOM을 이용한 FOV 생성 알고리즘 실행 결과의 한 예이다.



(a) (b)

그림 7. (a) 순차적 방법과 (b) 수정된 SOM을 이용한 방법의 FOV 생성 결과.
Fig. 7. Experimental FOV generation results using (a)sequential and (b) modified SOM method.

VI. 결론

본 연구는 CCD 카메라의 FOV를 자동으로 생성하기 위 하여 기존의 순차적인 방법과는 다른, 신경 회로망에서도 Self-Organizing Map을 기본으로 하여 다중 구조를 가지고, 이동·생성·삭제항 등의 부차적 항이 있는 새로운 트레이닝 법칙을 사용하는 알고리즘을 제안하였다. 본 연구에서 제안된 알고리즘은 기존 알고리즘의 결과값이 언제나 고정적인데 반하여, 유동적이다. 하지만, 전체 부품을 포함하는 FOV 개수는 기존의 알고리즘보다 약 10% 줄일 수 있음을 보여주었다. 이러한 결과값은 FOV의 중첩을 최소화시키고, 실제 PCB 검사 시에 총 검사 시간을 줄임으로서 생산성 향상을 가능하게 한다.

참고문헌

[1] T. S. Newman and A. K. Jain, "A survey of automated visual inspection," *Computer Vision and Image Understanding,* vol. 61, no. 2, pp. 231-261, 1995.

[2] Y. K. Ryu and H. S. Cho, "A neural network approach to extended gaussian image based solder joint inspection," *Mechatronics,* vol. 7, no. 2, pp. 159-184, 1997.

[3] Y. Y. Cha and D. G. Gweon, "A calibration and

range-data extraction algorithm of an omni-directional laser range finder for free ranging mobile robot," *Mechatronics,* vol. 6, no. 6, pp. 665-689, 1996.

[4] 오제휘. 차영엽, "부품 조립 공정에서 경로의 최적화 알고리즘," 한국정밀공학회지, 제14권, 제8호, pp. 122-129, 1997.

[5] 김철우, 비젼을 이용한 PCB 납땜 검사장치의 개발, 한국과학기술원, 1996.

[6] T. Kohonen, "The self-organizing map," *Proc. of the IEEE,* vol. 78, no. 9, pp. 1464-1480, 1990.

[7] A. S. Pandya and R. B. Macy, *Pattern recognition with neural networks in C++,* CRS Press, 1995.

오 제 휘

1996년 원광대 기계공학과 졸업. 1998년 동 대학원 졸업(석사), 관심분야는 영상처리, 지능제어.

차 영 엽

1984년 부산대 기계공학과 졸업. 1987년 한국과학기술원 생산공학과 졸업(석사). 1995년 한국과학기술원 정밀공학과 졸업 (박사), 1995년 3월~현재 원광대학교 기계공학부 조교수. 관심분야는 이동로봇, 영상처리, 지능제어.