

論文 90-27-6-17

## SMT 자동화를 위한 시각 시스템의 실시간 구현

(A Real-Time Implementation of the Vision System for SMT Automation)

田炳煥,<sup>\*\*\*\*</sup> 尹逸東,\* 金勇煥,\* 黃臣煥,\* 李商郁,\*\* 崔宗秀\*\*\*(Byeong Hwan Jeon, Il Dong Yun, Yong Hwan Kim, Shin Hwan Hwang,  
Sang Uk Lee, and Jong Soo Choi)

## 要 約

본 논문에서는 SMT(surface mounting technology) 자동화를 위한 고정밀 실시간 시각 시스템의 개발에 대하여 기술하였다. 또한 SMD(surface mounted device)에 대하여 위치와 방향을 정밀하게 구한 뒤 이들 정보를 이용하여 ruling을 수행하는 부품 검사 알고리즘을 제시하였고 그 알고리즘의 병렬 처리 방법을 제시하였다. 영상 취득 및 처리의 실시간 구현을 위하여 다기능 A/D-D/A 보드, 프레임 메모리 보드, 그래픽 오버레이 및 윈도우 제어 보드, 밝기 분포 추출기 및 특징 추출기 보드등 여러 가지 하드웨어 모듈을 개발하였다. 특히, 부품 검사 알고리즘의 병렬 처리를 위하여 DSP56000 DS DSP(digital signal processor)를 사용한 PE(processing element)보드를 개발하였다. 개발한 각 하드웨어 모듈 및 소프트웨어를 MVME1147 컴퓨터에 통합함으로써 독립된(stand-alone)시각 시스템을 이루었다.

## Abstract

This paper describes design and implementation of a real-time high-precision vision system for an automation of SMT (surface mounting technology). Also, a part inspection algorithm which calculates the position and direction of SMD (surface mounted device) accurately and performs the ruling using those information are presented, and a parallel processing technique for implementing those algorithms is also described. For a real-time implementation of image acquisition and processing, several hardware modules, namely, multi-functional A/D-D/A board, frame memory board, graphic overlay and window control board, histogram extractor board, and feature extractor board are developed. Particularly, a PE (processing element) board which employs the DSP56001 DSP (digital signal processor) is developed for the purpose of concurrent processing of part inspection algorithms. A stand-alone vision system is built by integration of the developed hardware modules and related softwares.

\*準會員, \*\*正會員, 서울대학교 制御計測工學科  
(Dept. of. Control & Inst. Eng., Seoul Nat'l Univ.)

\*\*\*正會員, 中央대학교 電子工學科  
(Dept. of Elec. Eng., Joongang Univ.)

\*\*\*\*準會員, 三星電子 家電部門 生産技術本部 로보  
트 開發팀  
(Robotics R&D Team corporate production Eng.  
& Tech. dev.)

接受日字: 1990年 2月 16日

## I. 서 론

SMT(surface mounting technology)는 PCB(printed circuit board)의 표면에 각종 소형 전자 부품을 납착하고 납땜하는 일련의 자동화 기술을 의미하는 것으로, 전자 기기 및 제품의 소형화, 고신뢰성, 다기능에 대한 기존의 기술은 PCB의 PTH(plate through hole)내에 부품의 리드(lead)를 삽입하여 납땜하는 방식으로서, PCB의 윗면에만 모든 부품이 배치되

는데 비해 SMT는 PCB의 표면에 부품을 정착하여 납땀하므로, 부품을 PCB의 윗면과 아랫면 모두에 배치할 수 있어서 이론적으로 기존보다 약 2배의 실장 효과가 기대된다. 그러나 부품의 소형화와 미세한 배선으로 인하여 SMT에서는 기존의 실장 기술보다 더욱 정교한 정착 기술이 필요하며, 대량 생산을 위해서는 고속 동작이 요구된다. 특히 부품 정착시에 발생하는 미세한 위치 오차도 불량률의 원인이 되므로 정착기에 공급 부품의 체크 기능을 갖게 하여 정착 후 위치의 어긋남과 불량품을 영상 처리에 의해 자동적으로 검출하는 기능이 필요하다. 따라서 최근 소량 다품종 생산을 위하여 시각 시스템(vision system)에 의해 제어되는 고정밀 조립용 로봇과 CAD/CAM을 결합한 시스템이 개발되고 있는 추세를 감안할 때 정착 정밀도 및 고속동작을 만족시킬 수 있는 시각 시스템의 개발이 필요함을 알 수 있다.

시각 시스템이 고속 수행 능력을 갖도록 하기 위해서는 고속 범용 프로세서에 의한 프로세서 제어와 수치 연산 전용 보조 프로세서의 이용이 아울러 요구되며 다수의 프로세서를 이용한 병렬 처리가 더욱 효과적이다. 그리고 시각 시스템이 높은 정밀도를 갖추기 위해서는 높은 공간 해상도와 명암도(gray level)를 가진 영상 데이터를 취득하여 처리할 수 있어야 한다. 따라서 최근 개발되어 발표된 시각 시스템들<sup>[1,2,3]</sup>은 모두 고해상도 영상 처리와 다수의 프로세서를 사용 병렬처리 기법을 이용하고 있음을 알 수 있다. 본 논문에서는 이상과 같은 하드웨어 설계 개념에 입각하여 SMT 자동화를 위한 고속 시각 시스템의 개발에 대해 기술하였다.

본 시스템은 프로세스 및 전체 시스템의 제어를 위해 MC68030 32bit CPU를 사용하며, 수치연산 전용으로 Motorola DSP 56001을 사용하였다. 또한 다수의 DSP 보드를 사용할 경우 어드레스 변환 look up table을 이용하여 영상 데이터의 신속하고 효율적인 분배가 이루어지도록 하여 병렬처리에 따른 손실 요소를 극소화 하였다. 그리고 구현된 시스템에 부품 검사를 위한 기존의 알고리즘을 적용하여 시스템의 실제동작 확인 및 성능평가를 수행하였다.

## II. 부품 검사 알고리즘

SMT에 있어서 부품 검사(part inspection)는 필수적으로 요구되는 시각검사 작업이다. 부품 검별에서는 장착 로봇의 end-effector에 위치한 SMD(surface mounted device)의 정확한 위치와 방향을 추출하고, 리드(lead)를 가진 SMD의 경우 그 소자에 대

한 ruling을 수행한다. 여기서 ruling이라 함은 자로 길이를 측정하듯이, 리드를 가진 SMD에서 리드의 길이와 간격을 측정하는 것을 말한다. SMD에 대한 검사를 수행한 후, 장착 로봇의 end-effector는 그 소자를 인쇄회로기판 위에 갖다 놓게 된다.

일반적으로 로봇 비전에서 물체의 면적, 위치, 방향과 같은 비교적 간단한 기하학적인 특징을 추출하고자 할 때에는 계산시간 및 알고리즘의 복잡성을 고려할 때 2진 영상(binary image)을 사용하는 것이 효과적이다.<sup>[4]</sup> 따라서, 본 부품검사 알고리즘에서는 간단한 기하학적인 형태를 만족하는 SMD를 대상으로 하며, 그 소자의 위치와 방향을 추출하는 것이 목적이므로 2진 영상을 이용하였다. 카메라의 시야에 하나의 물체만이 비추어진다면 물체외의 화소(pixel)는 배경으로 간주할 수 있으며, 물체와 배경을 뚜렷이 구분할 수 있다면 물체에 대한 특성 함수(characteristic function)를 간단히 정의할 수 있다.<sup>[5]</sup> 즉 물체에 대한 특성 함수는 물체에 해당하는 화소에 1을 할당하고, 배경에 해당하는 화소에는 0을 할당함으로써 얻을 수 있다. 그런데 2진 영상은 명암도 영상을 2치화함으로써 얻을 수 있으므로, 문턱값(threshold value)보다 밝은 화소는 0을 할당하고, 그렇지 않으면 1을 할당함으로써 특성 함수를 얻을 수 있다.<sup>[1]</sup>

화면내에 존재하는 물체의 위치를 추출하기 위해서 먼저 물체의 면적을 정의할 필요가 있다. 즉  $n \times m$  화면내에 한 개의 물체만이 존재한다고 가정하면 물체의 면적  $A$ 는 주어진 2진 영상의 특성 함수  $b(x, y)$ 를 이용하여 다음과 같이 구할 수 있다.

$$A = \sum_{y=1}^n \sum_{x=1}^m b_{xy} \quad (1)$$

물체는 여러 개의 화소에 구성되기 때문에 물체의 위치를 무엇으로 정의할 것인가는 중요하다. 물체내의 어떤 특징점을 기준점으로 하여 물체의 위치를 정의하는 방법도 있으나 보통의 경우는 물체의 면적 중심(center of area)을 대표점으로 한다. 단위 면적당 질량이 상수일 때 그 물체의 면적중심은 질량중심(center of mass)과 일치한다. 따라서 질량중심은 그 물체의 질량을 대표하는 하나의 점이라 할 수 있으므로 본 논문에서는 카메라에 비추어진 물체의 2진 영상에서 물체의 면적중심을 찾아 그 점을 물체의 위치로 정의하였다. 2차원인 경우 면적중심  $(\bar{x}, \bar{y})$ 는 다음과 같은 식을 만족한다.

$$\bar{x} = \frac{\sum_{y=1}^n \sum_{x=1}^m x b_{xy}}{A}, \quad A \neq 0 \quad (2a)$$

$$\bar{y} = \frac{\sum_{y=1}^n \sum_{x=1}^m y b_{xy}}{A}, \quad A \neq 0 \quad (2b)$$

여기에서 면적 A는 물체에 대한 0계 모멘트이고  $\bar{x}, \bar{y}$ 는 1계 모멘트이다. 물체의 특성 함수가 주어졌을 때, 이 물체의 방향을 추정하는 방법에는 물체내의 모든 점에서 거리의 제곱합이 최소화 되는 직선을 추정함으로써 물체의 방향을 구하는 방법이 일반적으로 이용된다.<sup>[2]</sup> 이 방법에 의하면 물체에 대한 방향 성분  $\theta$ 는 결국 물체에 대한 2계 모멘트로 구할 수 있다. 그 2계 모멘트는 3가지가 존재하는데 그들을 각각  $\alpha, \beta, \gamma$ 로 표현하면 식(3)과 같다.

$$\alpha = \sum_{y=1}^n \sum_{x=1}^m (x')^2 b_{xy} \quad (3a)$$

$$\beta = \sum_{y=1}^n \sum_{x=1}^m (x' y') b_{xy} \quad (3b)$$

$$\gamma = \sum_{y=1}^n \sum_{x=1}^m (y')^2 b_{xy} \quad (3c)$$

여기서  $x' = x - \bar{x}$ 이고  $y' = y - \bar{y}$ 이다. 식(3)을 이용하여 물체의 방향 성분  $\theta$ 를 구하면 다음과 같다.<sup>[2]</sup>

$$\theta = \frac{1}{2} \tan^{-1} \frac{\beta}{\alpha - \gamma} \quad (4)$$

Ruling은 의미 그대로 자로 측정하듯이 검사하는 방법으로써, 물체의 방향을 정밀하게 구한 뒤 물체의 방향과 직각의 방향으로 ruler를 평행이동시켜 가면서 run-length 법으로 리드의 길이와 간격을 측정한다. 여기서 ruler라 함은 물체의 방향 성분에 해당하는 기울기를 가지면서 물체 내부의 점을 지나는 직선을 말한다. 이 ruling 알고리즘을 상술하면 아래와 같다.

먼저, 방향 추출 알고리즘에서 구한 물체의 방향  $\theta$ 로 부터 물체의 기울기 a는 식(5)를 이용하여 구할 수 있다.

$$a = \tan(\theta) \quad (5)$$

따라서, 물체의 중심점  $P(x_c, y_c)$ 를 지나고 기울기 a를 가진 직선의 방정식은 식(6)과 같다.

$$y = ax - ax_c + y_c \quad (6)$$

이때, 기울기 a의 절대값이 1보다 같거나 작으면 ruler는 y축을 기준으로 이동하고 x좌표값을 순차적으로 증가시키면서 ruling을 수행한다. 그러나, 기울기 a의 절대값이 1보다 크면 ruler는 x축을 기준으로 이동하고 y좌표값을 순차적으로 증가시키면서

ruling을 수행하는데 이것을 도시하면 그림1과 같다. 그리고, 기울기 a의 절대값이 1보다 작을 때의 ruling 알고리즘을 그림 2에 순서도로써 나타내었다.

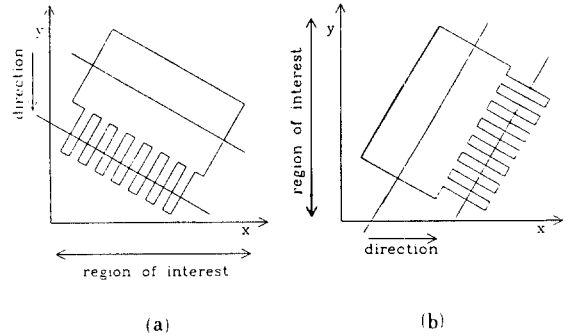


그림 1. 기울기의 크기에 따른 ruler의 이동 방향과 ruling 방향  
(a)  $\|a\| \leq 1$ 일 때  
(b)  $\|a\| > 1$ 일 때

Fig. 1. The direction of ruler and ruling with respect to the magnitude of slope.  
(a) when  $\|a\| \leq 1$ ,  
(b) when  $\|a\| > 1$ .

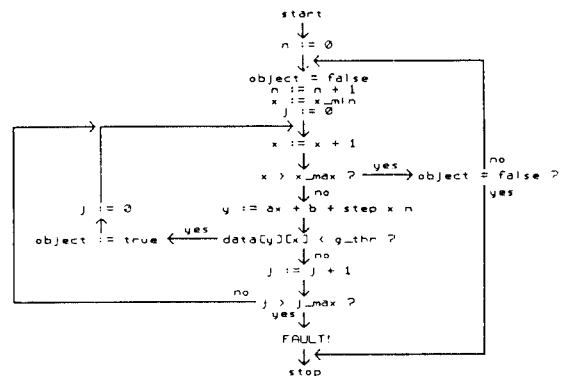


그림 2. Ruling 알고리즘에 대한 순서도  
Fig. 2. Flow chart for ruling algorithm.

Ruling에서는 ruler의 방향을 물체의 방향과 정확하게 일치시키는 것이 중요하다. 물체의 방향을 잘못 측정하여 ruler의 방향이 물체의 방향과 일치하지 않을 경우 검사 결과는 실제와 매우 다를 가능성이 있기 때문에 ruling을 적용시키기 위해서는 부품에

대한 정밀한 방향 추정이 중요한 과제가 된다. 그런데, 위치와 방향 추출 알고리즘은 각 화소 값과 영상 공간에서의 좌표값만이 필요하고 화소간의 상관관계나 화소의 분포등은 전혀 고려하지 않는다. 따라서 이러한 경우 한 프로세서가 전체 영상을 처리하기 보다는 한 영상을 여러 부분으로 분할하여 다수의 프로세서가 동시에 처리하게 함으로써 처리 시간을 대폭 감소시킬 수 있다. 즉 전체 512×512영상을 그림 3 과 같이 128×256 크기의 8부분으로 분할하였을 때 부분 영상 S<sub>i,j</sub>에서는 다음과 같은 식을 계산하게 된다.

|                 |                 |
|-----------------|-----------------|
| S <sub>00</sub> | S <sub>01</sub> |
| S <sub>10</sub> | S <sub>11</sub> |
| S <sub>20</sub> | S <sub>21</sub> |
| S <sub>30</sub> | S <sub>31</sub> |

그림 3. 512×512 영상을 8개의 프로세서가 병렬 처리하기 위하여 영상을 분할한 예  
 Fig. 3. The example of seperating 512×512 image for the parallel processing of the algorithm by 8 processor.

$$A_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} b_{xy} \quad (7a)$$

$$X_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} x b_{xy} \quad (7b)$$

$$Y_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} y b_{xy} \quad (7c)$$

$$\alpha_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} (x')^2 b_{xy} \quad (7d)$$

$$\beta_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} (x' y') b_{xy} \quad (7e)$$

$$\gamma_{i,j} = \sum_{y=j \times 128 + 1}^{(j+1) \times 128} \sum_{x=1 \times 256 + 1}^{(i+1) \times 256} (y')^2 b_{xy} \quad (7f)$$

따라서, 식(2)의 A,  $\bar{x}$ ,  $\bar{y}$ 와 식(3)의  $\alpha$ ,  $\beta$ ,  $\gamma$ 는 각각 위 식을 이용하여 다음과 같이 구할 수 있다.

$$A = \sum_{j=0}^3 \sum_{i=0}^1 A_{i,j} \quad (8a)$$

$$\bar{x} = (\sum_{j=0}^3 \sum_{i=0}^1 X_{i,j}) / A \quad (8b)$$

$$\bar{y} = (\sum_{j=0}^3 \sum_{i=0}^1 Y_{i,j}) / A \quad (8c)$$

$$\alpha = \sum_{j=0}^3 \sum_{i=0}^1 \alpha_{i,j} \quad (8d)$$

$$\beta = \sum_{j=0}^3 \sum_{i=0}^1 \beta_{i,j} \quad (8e)$$

$$\gamma = \sum_{j=0}^3 \sum_{i=0}^1 \gamma_{i,j} \quad (8f)$$

본 논문에서 구현한 시각 시스템은 DSP processor와 CPU를 이용하여 식7(a-c)와 식8(a-c)로 부터 물체의 위치를 계산한 후, 식 7(d-f)와 식 8(d-f)를 이용하여 물체의 방향을 계산하도록 하고 있다.

일반적으로 알고리즘을 병렬 처리하는데는 항상 job scheduling<sup>[6,7,8,9,10]</sup>과 처리된 데이터의 재합성(re-organization) 방법<sup>[11,12,13]</sup> 그리고 병렬 처리에 따른

손실 요소등을 고려하여야 한다. 그러나 물체의 위치와 방향을 구하는 위의 알고리즘에서는 각 프로세서간의 데이터 교환이 전혀 없으므로 병렬 처리에서 일반적으로 고려되는 프로세서간 통신 손실(inter processor communication cost)을 0으로 간주할 수 있다. 다만, 주 프로세서는 식(8)을 계산하기 위하여 2번의 나눗셈과 48번의 덧셈을 계산하기 위한 데이터의 교환이 이루어지는데 이것은 전체 계산량에 비해 거의 무시할 수 있는 값이다. 이상의 고찰에서 볼 때 부품 검사 알고리즘의 병렬 처리에 따른 손실 요소는 거의 없으므로 프로세서의 수의 증가에 따른 시스템의 성능 증대가 매우 효율적으로 이루어짐을 알 수 있다.

### Ⅲ. 하아드웨어의 구성

본 논문에서 SMT 자동화를 위한 시각 시스템을 하아드웨어로 구현할 때 다음과 같은 목표들을 실행하고자 하였다.

- 고속 처리 능력
- 고해상도 영상의 취득 및 처리
- 독립된 시각 시스템
- 기능의 모듈화
- 비전 알고리즘의 병렬 처리
- 통합된 하아드웨어 및 소프트웨어 개발 환경
- 범용 시각 시스템으로의 활용

SMT 자동화를 위해서는 시각 시스템에서의 검사와 로봇에 의한 장착 시간이 172핀 소자의 경우, 일반적으로 3.5초 이내에 수행될 것을 요구하고 있다. 따라서 검색 작업의 고속수행은 시각 시스템에 있어서 가장 중요한 필요 조건이다. 일반적으로 간단한 영상 처리에서는 정수 덧셈과 곱셈, 그리고 비트 조작이 대부분을 이루며 영상 영역 전체에 걸쳐 수행되는 경우가 많다. 따라서 하나의 범용 프로세서 각 전체 영상에 대해 연산을 수행시키기 보다는 덧

셈, 곱셈등을 고속으로 수행하는 보조 프로세서를 다수 채용하여 영상을 분할, 담당케 함으로써 처리 시간을 대폭 감축시킬 수 있다. 본 논문에서는 범용 프로세서인 MC68030으로 하여금 전체 시스템을 제어하도록 하고, 대량의 연산은 DSP56001로 하여금 담당하게 하였다. 또한 다수의 DSP 보드를 사용할 경우 영상 데이터의 전송 및 분배가 효율적으로 이루어지도록 하여 비전 알고리즘의 병렬 처리화와 함께 고속 처리가 가능하도록 설계하였다. 또한 기능의 모듈화를 통해 시스템의 신뢰도를 향상시키고, 소프트웨어로 처리할 수 있는 기능 중 단순 반복적이면서 처리 시간이 많이 걸리는 모듈은 하드웨어로 제작하여 영상 취득과 동시에 연산 결과를 출력할 수 있도록 설계하였다. 본 논문에서 하드웨어로 제작된 모듈은 밝기 분포 취득기(histogram extractor), 특징 추출기(feature extractor), 그래픽 오버레이 제어기(graphic overlay controller), 윈도우 제어기(window controller)로서, 소프트웨어에 의해 수행될 수 있는 연산을 하드웨어가 담당하도록 하여 전체 시스템의 소프트웨어를 단순화하고 처리 시간상의 이득을 얻을 수 있도록 하였다.

자동화된 전체 공정을 가정할 때 자동 표면 실장기는 여러 종류의 자동화된 기계 중 한 부분에 해당되고 주제어기의 제어하에 작업이 이루어진다고 볼 수 있다. 따라서 SMT에 필요한 시각검사 작업을 주 제어기에 의존하지 않고, 시각 시스템으로 하여금 독립적으로 수행시킴으로써 주제어기의 부하를 줄일 수 있으며, 시각 시스템의 이동성(portability) 및 범용성을 제고시킬 수 있다. 또한 개발된 시스템을 다양한 응용 분야에 적용시키기 위해서는 하드웨어 및 소프트웨어의 개발 환경이 우수하여야 한다. 본 논문에서는 UNIX 운영 체제하에서 고급 언어인 C로 전체 시각 시스템을 제어, 운용할 수 있도록 하였으며 VME 버스 시스템하에서 하드웨어를 개발하기 위하여 MVME1147 컴퓨터 시스템을 이용하였다. MVME1147 컴퓨터는 MC68030-based 컴퓨터로서 4 Mbyte의 주기억 장치와 85 Mbyte 자기 디스크 기억 장치, 150 Mbyte의 테이프 백업 장치, 그리고 UNIX 운영 체제등으로 구성된다.

본 논문에서 개발한 시각 시스템은 그림 4와 같은 전체 구조를 가지며 프로세스 제어와 데이터 전송의 효율을 향상시키기 위하여 VME 버스와 VIDEO 버스의 두가지 계층적 버스구조를 이용하였다. VME 버스는 CPU가 시스템을 구성하는 각 모듈에 대해서 데이터 전송 및 통제를 위하여 사용되며 VIDEO 버스는 A/D 변환기로 부터의 디지털 비디오 데이터를

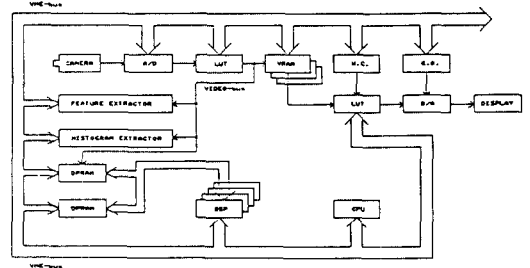


그림 4. 전체 시스템의 블럭 다이어그램  
Fig. 4. Block diagram of overall system.

필요로하는 각 모듈에 96Mbit/s의 전송률로 전송하기 위해 이용된다.

A/D-D/A 보드는 그림5와 같은 구조로 되어 있는데 A/D 변환기는 카메라로 부터 출력되는 아날로그 비디오 신호를 디지털 신호로 변환하며, D/A 변환기는 디지털 비디오 신호를 모니터로 출력하기 위하여 아날로그 신호로 변환한다. 여기서 A/D변환기는 최대 8개의 카메라로 부터의 신호를 선택하여 받아들일 수 있으며 512×512×8bit의 해상도를 가진다. 그리고, 프레임 메모리 전후에 있는 LUT(look-up table)에 의해 실시간 thresholding, 영상 반전, 밝기분포 변환(histogram modification), 각 bit plane의 추출등의 기능을 수행할 수 있도록 되어 있다.

한편 컴퓨터 비전 알고리즘의 효율적인 수행을 위해서는 프레임 메모리에 저장된 영상 데이터의 자유로운 액세스가 보장되어야 한다. 이를위해 본 시스템에서는 TMS4461 VRAM을 사용하여 영상표시 구간에 상관없이 메모리 액세스가 가능하도록 하였다.

그래픽 오버레이 보드는 문자 및 도형을 영상 데이터와 겹쳐서 화면에 표시하기 위해서 사용하며, 그래픽 출력 제어기인 GDC 82720과 512×512×4 bit의 그래픽 메모리로 구성되어 있다. 그래픽 오버레이 보드는 A/D 보드내에 있는 GDC와 동기되어 작동하며 MC68030주프로세스의 제어를 받는다. GDC 82720

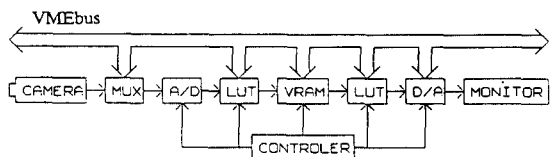


그림 5. A/D-D/A 변환부의 블럭 다이어그램  
Fig. 5. Block diagram of A/D-D/A converter.

은 직선, 원, 문자등과 같은 간단한 도형의 묘화 기능이 있기 때문에 주프로세서의 부하를 줄일 수 있다. 이러한 그래픽 오버레이 기능을 이용함으로써 원래의 영상 데이터에 대한 변경없이 각종 문자 및 도형을 화면에 표시할 수 있으며, 또한 입력 영상을 처리하여 얻은 영상 영역화, 에지 검출, 특징 추출 등의 결과를 본래 영상과 겹쳐서 표시함으로써 처리 결과를 용이하게 확인할 수 있다.

윈도우 제어 기능은 화면의 특정 부분을 선택적으로 지정하여 표시하는 것으로서, 화면에 최대 8 개까지의 윈도우 지정이 가능하도록 설계하였다. 윈도우는 화면의 표시에 영향을 줄 뿐만 아니라, 제반 하드웨어 및 소프트웨어의 동작에 영향을 끼친다. 윈도우 신호에 의한 하드웨어 동작을 살펴보면 프레임 버퍼 메모리를 선택하는데 있어서 윈도우 신호에 의해 각각 다른 프레임 버퍼를 저장할 수 있으며, A/D 보드에 있는 입력, 출력 look-up table을 윈도우별로 다르게 지정할 수 있으므로, 한 화면에 여러 종류의 윈도우를 표시할 수 있다. 윈도우 발생기는 3 개의 look-up table(LUT) RAM으로 이루어진다. 9bit의 수평 어드레스와 9 bit의 수직 어드레스는 첫번째 LUT를 통과함으로써 각각 4 bit의 중간 윈도우 신호를 만든 다음 이를 이용하여 최종적으로 8 bit의 윈도우 제어 신호를 출력한다. 이상에서 설명한 윈도우 신호의 발생 원리를 그림 6에 나타내었다.

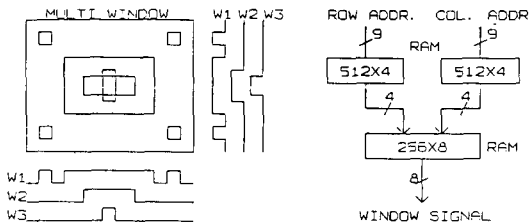


그림 6. RAM을 이용한 윈도우 제어 신호의 발생  
Fig. 6. Generation of window control signal using RAM.

밝기 분포 추출기에서 얻고자하는 밝기분포도(histogram)란 화소들의 명암도를 0부터 255로 나눈다고 할 때, 한 화면에서 각각의 명암도를 갖는 화소의 빈도를 의미한다. 그러므로 밝기 분포도를 실시간으로 얻는 방법의 하나로서 그림 7과 같은 구조를 생각할 수 있다. 이 방식은 256개의 기억 장소를 갖는 RAM을 사용한 것으로서 영상 데이터가 RAM의 입

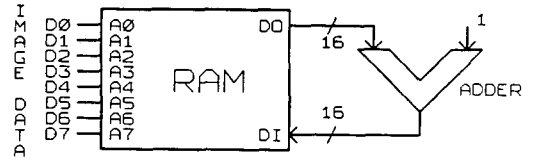


그림 7. 256 개의 밝기분포도를 구하기 위한 하드웨어 구성도  
Fig. 7. The circuit diagram for extracting histogram from an image.

력에 연결되어 각각의 명암도가 각각의 기억 장소를 가지며, 명암도를 갖는 데이터가 들어오면 그 명암도에 대응하는 기억 장소의 데이터를 읽어서, 가산기를 이용하여 읽어낸 데이터에 1을 더하여 다시 저장하는 방식이다.

특징 추출기의 기능은 특정 밝기 구간에 해당하는 화소를 물체에 해당되는 영역으로 간주하여 그 화소들의 수직 방향 어드레스와 수평방향의 어드레스를 각각 더한 후에 그 영역의 전체 화소 갯수로 나눔으로써 물체의 중심 위치에 해당하는 어드레스를 얻는 것이다. 단 물체에 해당되는 화소 갯수는 밝기분포도로 부터 얻기 때문에 특징추출기는 단지 수직방향 어드레스와 수평 방향의 어드레스를 가산만 한다. 이러한 중심위치(centroid)는 표면 실장 기술에서 식(2)를 이용하여 물체의 실제 위치를 파악하는데 중요한 정보로 이용된다. 그림 8에 이러한 기능을 위한 블록 다이어그램을 나타내었다. 이와 같은 구조를 갖는 특징 추출기는 수평 방향과 수직 방향에 대해 각각 1개씩 사용되며, 각각 독립적으로 연산을 수행한다. 여기서 수평 방향 어드레스와 수직 방향 어드레스는 각각 GDC에서 나오는 18 bit의 어드레스 값에서 하위 9 bit와 상위 9 bit에 해당된다.

일반적으로 DSP는 곱셈기(multiplier)와 고기능 누산기(accumulator)를 내장하고 있어서 일반 범용 프로세서와는 달리 곱셈 연산을 1개의 명령 사이클에 수행하므로 곱셈량이 많은 알고리즘의 처리에 적당하다. 부품 검사 알고리즘은 II장에서 고찰한 바와 같이 계산량이 많고 병렬처리가 용이하므로 다수의 DSP를 PE(processing element)로 이용하여 병렬처리가 가능하도록 하였다. DSP를 사용한 PE들은 VME 버스에 연결되어 MC68030주 프로세서의 제어를 받게 되며, 연산에 필요한 영상 데이터는 A/D-D/A 보드로 부터 VIDEO 버스를 통해 직접 전송 받게 설계하였다. DSP 보드는 DSP56001과 비디오 데이터 RAM부, 어드레스변환부(address modifier),

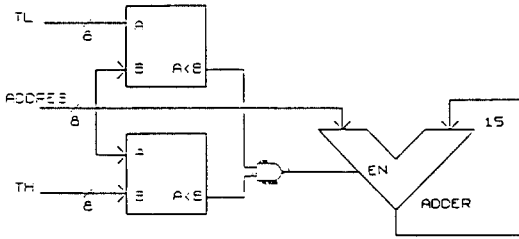


그림 8. 특징 추출기의 블럭 다이어그램  
Fig. 8. Block diagram of feature extractor.

프로그램 RAM, Y 데이터 RAM부, X 데이터 RAM부, dual-port RAM부, 인터럽트 제어부로 구성된다. 이들 기능 모듈의 구성 형태를 그림9에 블럭 다이어그램으로 나타내었다. 이 그림에서 보면, dual-port RAM만이 VME 버스에 연결되어 있고 다른 기능모듈은 VME 버스와 독립적으로 존재하고 있음을 알 수 있다. 따라서 MC68030 CPU 측에서 보면 DSP 보드는 하나의 메모리 보드로 간주하게 된다. 비디오 데이터와 어드레스는 A/D 보드로 부터 공급받는데 비디오 어드레스는 18 bit로써 256Kbyte의 정보를 전송한다. 그러나 DSP 보드에 할당된 입력 데이터 RAM 영역은 32 Kbyte로 제한되어 있기 때문에 256 Kbyte의 전체 영상 데이터 중에서 특정부분을 DSP 보드의 32 Kbyte로 대응시키기 위한 비디오 어드레스의 변환이 필요하게 된다. 이러한 작용은 비디오 어드레스를 입력으로 갖고 DSP RAM 어드레스를 출력으로 하는 어드레스 변환 look-up table에 의해 이루어지며, 각 DSP 보드는 한 화면 내에서 자신이 처리해야할 부분의 비디오 어드레스가 입력될 때에만 DSP RAM 어드레스가 출력되도록 look-up table을 프로그램함으로써 VIDEO 버스를 통해 특정 영역의 비디오 데이터만을 취득할 수 있다. 이와 같이 함으로써 각 DSP 보드는 한 화면의 데이터가 입력되

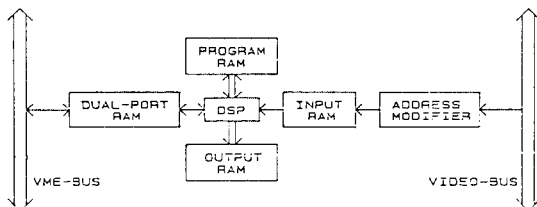


그림 9. DSP56001 보드의 블럭 다이어그램  
Fig. 9. Block diagram of DSP 56000 board.

는 동안, 512×512의 영상 공간 내에서 임의의 형상을 가진 윈도우를 취하여 최대 32 Kbytes의 영상을 취득할 수 있다. 따라서, 512×512의 영상을 1개의 PE 보드로 처리하는 경우 8개의 부분 영상을 순차적으로 처리하여야 하며, 8개의 PE 보드를 사용하면 8개의 부분 영상을 각 PE 보드에 할당하여 병렬 처리할 수 있다.

IV. 실험 결과 및 검토

본 논문에서 설계-제작한 각 보드를 MVME1147 컴퓨터 시스템에 통합하여 시각 시스템을 구성하고 시각 시스템으로서의 제반 기능과 성능에 대하여 실험을 수행하였다. 그림10은 제작한 각 하아드웨어 모듈을 MVME1147 컴퓨터에 통합하여 시각 시스템을 이룬 실물 사진이고 그림11은 리드를 가진 소자

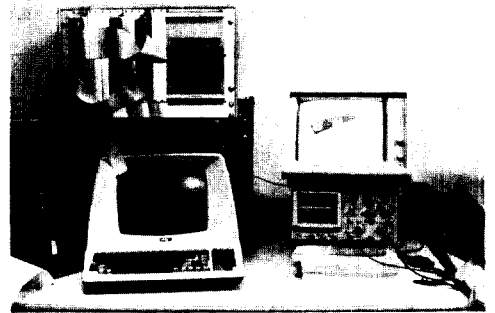


그림10. 완성된 시각 시스템  
Fig. 10. Completed vision system.

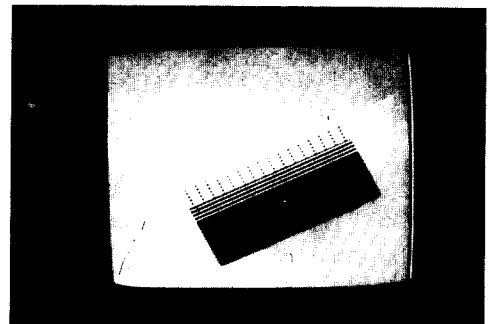


그림11. 물체의 위치와 방향을 구하고 ruling을 수행한 모습  
Fig. 11. The picture which shows the result of ruling from the extracted position and direction of an object in image.

의 위치와 방향을 구한 뒤 ruling을 수행한 모습이다.

본 논문에서 개발한 시각 시스템에서 위치 추출, 방향 추출, ruling 등의 부품 검사 알고리즘을 수행하는데 이용되는 연산 처리 모듈은 MVME1147 컴퓨터, DSP 보드, 특정 추출기 보드이다. 부품 검사 알고리즘의 수행 시간은 위의 세가지 연산 처리 모듈 중 어느 것을 이용하는가에 따라 다르게 나타난다. 실험에서는 아래의 2가지 방법으로 실험하여 부품 검사 알고리즘의 수행 시간을 측정하였다.

1. MVME1147 컴퓨터만을 이용하는 경우
2. MVME1147 컴퓨터와 1개의 DSP 보드, 그리고 특정 추출기 보드를 이용하는 경우

1,2 모두 영상 데이터 취득과 표시에는 A/D-D/A 보드, 프레임 메모리 보드, 윈도우 제어기, 그리고 그래픽 오버레이 보드를 이용하였다. 또한 2진 영상을 얻기 위하여 밝기분포도에 의한 thresholding 기법을 이용하였으며, 밝기분포도는 실시간 밝기 분포 추출기를 사용하여 얻었다. 1)은 부품 검사 알고리즘을 MC68030 프로세서로 수행시키는 것이며 프로그램은 C언어로 작성하였다. 1)의 경우 MC68030 프로세서가 프레임 메모리 보드로 부터 영상 데이터를 읽어 들이는 시간이 약 600ms 정도로서 전체 처리 시간에서 상당한 비중을 차지한다. 2)는 1개의 DSP 보드를 이용하여 부품 검사 알고리즘을 수행하는 것이다. DSP 보드는 영상 데이터를 32 Kbyte 단위로 받아들여 처리하므로 512×512전체 영상을 처리하기 위해서는 8번의 데이터 전송이 필요하다. 8번의 데이터 전송에 소요되는 시간은 약 400ms로서 역시 전체 처리 시간에서 큰 비중을 차지한다. 그런데, 부품 검사 알고리즘의 수행 시간은 영상 내에서 물체에 해당하는 화소의 수에 비례하여 증가한다. 표1의 (a) (b)는 영상에서 물체에 해당하는 화소의 수가 약15%를 점유할 때 위의 1), 2)의 경우 수행시간을 나타낸 것이다. 이상의 각 실험 결과는 SMT에서 요구되는 검사 및 장착 시간의 한계 3.5초를 충분히 만족함을 알 수 있다.

그런데, 전술한 것처럼 본 논문에서 개발한 시각 시스템은 다수의 DSP 보드를 사용하여 부품검사 알고리즘의 병렬처리가 가능하도록 되어 있다. 따라서, II 장에서의 병렬처리 알고리즘에서 고찰한 바와 같이 부품검사 알고리즘은 PE간의 프로세서간 통신 손실이 거의 0이고 job scheduling에 소요되는 시간이 0이므로 사용하는 DSP 보드의 수에 반비례하여 처리시간이 감소하게 된다. 표 2는 DSP 보드를 8개 사용하여 그림3와 같이 영상을 분할하여 처리하는

표 1. 연산처리 장치의 선택에 따른 부품 검사 알고리즘의 수행시간

Table 1. The execution times for part inspection employing various processing devices.

|        | 계산 시간       | 전송 시간 |
|--------|-------------|-------|
| 위치 추출  | 800ms       | 666ms |
| 방향 추출  | 1177ms      | -     |
| ruling | 17ms/1 line | -     |

(a) (1)의 경우

|        | 계산 시간       | 전송 시간 |
|--------|-------------|-------|
| 위치 추출  | 33ms        | 50ms  |
| 방향 추출  | 366ms       | 400ms |
| ruling | 17ms/1 line | 400ms |

(b) (2)의 경우

표 2. 8개의 DSP 보드로 부품 검사 알고리즘을 수행할때의 예상 소요 시간

Table 2. The expected execution time for conducting part inspection using 8 DSP boards.

|        | 계산 시간       | 전송 시간 |
|--------|-------------|-------|
| 위치 추출  | 44ms        | 50ms  |
| 방향 추출  | 46ms        | -     |
| ruling | 17ms/1 line | -     |

경우의 부품 검사 알고리즘의 예상 수행 시간이다. 실험에서는 구하려는 물체의 위치와 방향의 정밀도를 높이기 위하여, II 장에서 기술한 바와 같은 계산량이 많은 알고리즘을 이용하였으나 부품의 형상에 대하여 제약을 가하여 간단한 알고리즘을 이용한다면 위치 및 방향 추출에서의 수행시간을 감소시킬 수 있을 것이다.

이상의 실험 결과에서, 본 논문에서 개발한 시각 시스템은 SMT 자동화에 요구되는 정밀도와 고속수행 능력을 만족하고 있음을 확인하였으며, 독립된 시각 시스템으로서의 요건을 갖추고 있음을 알 수 있다. 본 논문에서 개발한 시각 시스템은 범용 시각 시스템으로서의 요건을 충족하고 있으므로 SMT 자동화외의 다양한 분야에서의 산업용 시각 시스템으로 이용될 수 있을 것이다.

### V. 결 론

본 논문에서는 SMT 자동화를 위한 실시간 시각



시스템을 개발에 대해 기술하였다. SMT 자동화를 위해서는 정밀한 부품 검사를 수행하는 알고리즘과 이 알고리즘을 고속으로 처리할 수 있는 하드웨어 장치가 필요하다. 본 논문에서는 III장에서 제시한 바와 같은 고속 연산처리 기능을 갖는 하드웨어의 설계 개념에 입각하여 고속 시각 시스템을 위한 영상 취득 및 처리 장치를 개발하였다.

첫째, SMD 검사 및 장착시 요구되는 정밀도를 고려하여 512×512의 해상도를 갖는 A/D-D/A 보드와 프레임 메모리 보드를 개발하였다. 또한 프레임 메모리 보드에서 dual port 메모리 기술을 사용하여 영상 데이터 취득 및 전송에 요구되는 시간을 최소화함으로써 영상처리 알고리즘이 실시간으로 동작할 수 있는 기반을 제공하였다.

둘째, 영상처리 결과를 문자 및 도형으로 나타내기 위한 그래픽 오버레이와 알고리즘에 사용되는 데이터의 사용 범위를 한정하기 위한 윈도우 제어 보드를 개발하였으며 입력 영상의 밝기 분포를 실시간으로 계산할 수 있는 밝기 분포 추출기와 물체의 면적 중심을 실시간으로 계산할 수 있는 특징 추출기를 개발하였다.

셋째, 수치 연산 능력을 향상시키기 위하여 DSP 56001을 이용한 보조 연산 처리 보드를 개발하였으며, 이 DSP 보드를 여러개 사용할 경우 병렬 처리가 가능하도록 설계하였다.

넷째, 이상에서 열거한 각 하드웨어 모듈을 MVME1147 컴퓨터 시스템에 통합함으로써 독립된 시각 시스템을 이루었다.

시각 시스템은 로보트 시스템과 결합하여 실제 작업을 담당하는 로보트에게 시각 정보를 제공함으로써 로보트의 지능화에 기여하며 궁극적으로 자동화에 기여한다. 따라서 보다 효율적이고 유연한 자동화 시스템의 구축을 위해서는 시각 시스템을 이용하는 지능 로보트 시스템에 대한 연구가 계속되어야 할 것이다.

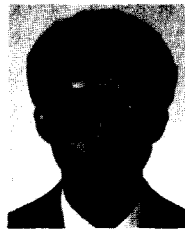
#### 參 考 文 獻

- [1] Lorenz A. Schmitt and Stephen S. Wilson, "The AIS-5000 parallel processor," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 320-330, May 1988.
- [2] D. Paul, W. Hattich, W. Nill, S. Tatari, and G. Winkler, "VISTA: Visual interpretation system for technical applications - architecture and use," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, no. 3, pp. 399-407, May 1988.
- [3] Richard S. Wallace and Michael D. Howard, "HBA vision architecture: built and benchmarked," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 33, pp. 227-232, Mar. 1989.
- [4] K.S. Fu, R.C. Gonzalez, C.S.G. Lee, Robotics, control, sensing, vision, and intelligence, McGraw-Hill, 1988.
- [5] Berthold Klaus Paul Horn, Robot vision, *The MIT Press*, 1986.
- [6] Harold S. Stone, "Multiprocessor scheduling with the aid of network flow algorithms," *IEEE Trans. Software Engineering*, vol. SE-3, no. 1, pp. 85-93, Jan. 1977.
- [7] Gururaj S. Rao, Harold S. Stone, and T.C. Hu, "Assignment of Tasks in a distributed processor system with limited memory," *IEEE Trans. Computer*, vol. C-28, no. 4, pp. 291-299, Apr. 1979.
- [8] Yuan-Chieh Chow and Walter H. Kohler, "Model for dynamic load balancing in a heterogeneous multiple processor system," *IEEE Trans. Computer.*, vol. C-28, no. 5, pp. 354-361, May 1979.
- [9] Ashok K. Agrawala, Satish K. Tripathi, and Glenn Ricart, "Adaptive routing using a virtual waiting time technique," *IEEE Trans. Software Engineering*, vol. SE-8, no. 1, pp. 76-81, Jan. 1982.
- [10] Philip Heidelberger and Kishor S. Trivedi, "Queueing network models for parallel processing with asynchronous tasks," *IEEE Trans. Computers*, vol. C-31, no. 11, pp. 1099-1109, Nov. 1982.
- [11] Alok N. Choudhary and Janak H. Patel, "A parallel processing architecture for an Integrated vision system," *Proceedings of 1988 International Conference on Parallel Processing*, vol. I Architecture, pp. 383-387, 1988.
- [12] M.H. Sunwoo and J.K. Aggarwal, "Flexibly coupled multiprocessors for image processing," *Proceedings of the 1988 International Conference on Parallel Processing*, vol. I Architecture, pp. 452-461, 1988.
- [13] James J. Little, Guy E. Blelloch, and Todd A. Cass, "Algorithmic techniques for computer vision on a fine-grained parallel machine," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 3, pp. 244-257, Mar. 1989.

著 者 紹 介



**田 炳 煥(正會員)**  
 1965年 12月 9日生. 1988年 2月  
 서울대학교 제어계측공학과 졸업.  
 1990年 2月 동대학원 석사과정졸  
 업, 현재 삼성전자 가전부문 생  
 산기술본부 로봇개발팀 연구원  
 주관심분야는 영상처리 및 로보  
 트 비전 분야 등임.



**黄 臣 煥(正會員)**  
 1964年 3月 25日生. 1986年 서울  
 대학교 제어계측공학과 졸업. 19  
 88年 서울대학교 대학원 제어계  
 측공학과 석사학위 취득. 1988年  
 ~현재 동대학원 박사과정 재학중  
 주관심분야는 영상처리및 컴퓨터  
 비전 등임.



**尹 逸 東(準會員)**  
 1965年 7月 21日生. 1989年 2月  
 서울대학교 제어계측공학과 졸업  
 공학사 학위취득. 현재 동대학원  
 석사과정 재학중. 주관심분야는  
 영상처리 및 VLSI 신호처리 분  
 야 등임.



**李 商 郁 (正會員)** 第25卷 第7號 參照  
 현재 서울대학교 제어계측공  
 학과 부교수



**金 勇 煥(正會員)**  
 1966年 3月 12日生. 1989年 2月  
 서울대학교 제어계측공학과 졸업  
 현재 동대학원 석사과정 재학중  
 주관심분야는 영상처리 및 예러  
 정정 부호화 기법분야 등임.



**崔 宗 秀 (正會員)** 第26卷 第9號 參照  
 현재 중앙대학교 전자공학과  
 부교수