

〈論 文〉

# 퍼지 클러스터링과 스트링 매칭을 통합한 형상 인식법

남원우\* · 이상조\*\*

(1993년 2월 20일 접수)

## Pattern Recognition Method Using Fuzzy Clustering and String Matching

Weon-Woo Nam and Sang-Jo Lee

**Key Words :** Image Processing(화상 처리), Fuzzy Clustering(퍼지 집단화), String Matching(스트링 매칭)

### Abstract

Most of the current 2-D object recognition systems are model-based. In such systems, the representation of each of a known set of objects are precompiled and stored in a database of models. Later, they are used to recognize the image of an object in each instance. In this thesis, the approach method for the 2-D object recognition is treating an object boundary as a string of structural units and utilizing string matching to analyze the scenes. To reduce string matching time, models are rebuilt by means of fuzzy c-means clustering algorithm. In this experiments, the image of objects were taken at initial position of a robot from the CCD camera, and the models are constructed by the proposed algorithm. After that the image of an unknown object is taken by the camera at a random position, and then the unknown object is identified by a comparison between the unknown object and models. Finally, the amount of translation and rotation of object from the initial position is computed.

### 1. 서 론

공장자동화에 대한 관심이 확대됨에 따라 이를 실현하기 위한 각 분야의 연구가 활발히 진행되고 있다. 그 중에서 인간의 눈을 대신할 수 있는 로봇트 비전에 대한 연구도 활발히 진행되고 있는 분야로써 지금은 2차원 평면인식을 넘어서 3차원 인식까지 연구되고 있다. 그러나 3차원 인식은 아직 초기 단계로써 좀더 깊은 연구가 필요하고 현재는 2차원 인식으로써 물체를 판단하여 로봇트를 제어한다.

로봇트 비전에 있어서 주된 문제는 첫째, 주어진 문제를 풀기위하여 카메라로부터 받은 화상을 심벌화된 데이터로 만든다. 둘째, 데이터를 해석하여 데이터베이스를 구축한다. 셋째, 데이터 베이스에 따라 일을 수행한다. 즉, 첫 번째 문제는 디지털 화상 해석의 문제이고 두 번째 문제는 화상 인식을 표현하는 문제이고 세번째는 제어전략에 대한 문제이다.

자동 생산에 대한 기대가 증가함에 따라 2차원 물체의 해석에 점점 흥미를 갖게 되었고, 2차원 물체의 해석에 있어서 기본이 되고 또 그 응용분야가 넓어 현재 활발히 연구가 진행되고 있는 주제는 모델을 기반으로 한 물체 인식(model based object recognition)이다. Model based 시스템에는 학습

\*정회원, 연세대학교 기계공학과 대학원

\*\*정회원, 연세대학교 기계공학과

단계 (training phase)와 인식 단계 (recognition phase)의 두가지 단계가 있다. 학습단계란 주어진 화상을 해석하여 인식전략을 개발하는 단계이고, 인식단계란 주어진 인식 전략에 따라 미지의 화상이 어떤 모델에 맞는가를 인식하는 단계이다. 그러므로 화상인식에 있어서 주된 과제는 인식전략을 세우는 것이다.

'62년도에 Moment Invariant법에 의한 패턴 인식법이 개발되었고, 그 후 Complex Moment법을 이용하여 화상의 정규화를 시도했다. 그러나, 이와 같이 화상의 모멘트 또는 고유값으로 화상을 인식한다든가 정규화하기 위해서는 전체 화상의 데이터가 필요하기 때문에 계산 시간이 너무 많이 소요된다. 그래서 '80년대 초반에 화상의 윤곽선만을 뽑아서 이것을 모델로 선정한 후에 패턴 인식을 시도하여 가리워진 물체의 인식을 연구하였다. 그러나, 이것은 모델과의 매칭은 좋게 나왔지만 물체의 위치를 판정하기는 불가능하였다. Tsai와 Yu<sup>(4)</sup>는 string matching을 이용하여 화상인식을 했으나 스트링의 선정이 미흡하여 계산시간이 너무 길었을 뿐만 아니라 위치도 판정하지 못하였다. Grosky<sup>(5)</sup>는 Yu<sup>(4)</sup>가 제안한 방법을 보완하여 Icon Indexing Mechanism을 연구하였다.

본 연구에서는 블럭 처리 기법에 의해서 화상을 분할함으로써 그림자와 물체의 분할을 가능하게 하였고, Grosky가 제안한 방법을 이용하여 매칭에 필요한 스트링을 선정하여 모델을 구성하였다. 또한, 학습단계의 인식전략을 세우는데 있어서 Bezdek<sup>(6)</sup>이 발표한 퍼지 클러스터링 방법을 이용하여 적절한 클러스터를 선정하여 계산시간을 줄였다.

본 연구에서 제안한 방법은 모델들을 계층적으로 찾아나가지 않고, 먼저 클러스터의 중심을 찾은 후에 각 클러스터의 중심과 미지 물체와의 거리를 계산하여 거리가 가장 짧은 클러스터를 구하고, 그 클러스터 내의 멤버들과의 거리를 다시 계산하여 정확한 모델을 찾아 나가는 순서를 밟는다.

## 2. 화상의 전처리 과정

### 2.1 블럭 처리 기법에 의한 화상의 분할

Image processor에서 받아 들이는 화상은 8bit의 명암차이를 나타낼 수 있으므로 0부터 255까지 256개의 gray level로 보여진다. 이 256개의 gray

level을 갖는 화상을 0 또는 1만의 값으로 이루어진 화상으로 만든다. 이렇게 만들어진 화상을 이진 화상이라고 한다.

화상의 밝기가 고르지 않거나 그림자가 있으면 화상을 정확하게 물체와 배경으로 분할하기가 쉽지 않다. 그래서 본 논문에서는 블럭처리에 의한 화상의 분할방법을 사용하여 화상을 적절히 배경과 물체로 분리하였다.<sup>(9)</sup> 여기에서 A, B의 값은 물체의 선명도를 결정하는 요인으로 조도가 세어지면 A, B의 차이를 크게하고 약해지면 작게한다. 또한, C의 값은 그림자의 선명도를 결정하는 인자이므로 보통 B의 값과 비슷하게 결정한다. 그리고, D, E의 값은 1보다 조금 작은 값을 택한다.

단계 ① : 전체화상의 평균을 이용한 화상의 분할  
 $\text{화소값} > \text{전체화상의 평균} \times A$ 이면 배경  
 $\text{화소값} < \text{전체화상의 평균} \times B$ 이면 물체

단계 ② :  $(\text{블럭의 평균} - \text{그 주변의 평균}) <$   
 $(\text{블럭 평균의 최대값} - \text{최소값}) \times C$   
 $\text{화소값} > \text{주변블럭의 평균}$  이면 배경  
 아니면 물체

단계 ③ :  $\text{주변블럭의 평균} > \text{블럭의 평균}$   
 $\text{화소값} > \text{블럭의 평균} + (\text{주변 블럭의 평균} - \text{블럭의 평균}) \times D$  이면 배경  
 아니면 물체

단계 ④ :  $\text{주변블럭의 평균} < \text{블럭의 평균}$   
 $\text{화소값} > \text{블럭의 평균} - (\text{주변의 평균} - \text{블럭의 평균}) \times E$  이면 배경  
 아니면 물체

이 알고리즘에서 단계 ①의 과정을 이용해 배경 또는 물체의 선명도를 높이고, 단계 ②의 과정은 처리중인 블럭이 주변의 블럭과 밝기가 비슷하다는 것을 의미한다. 단계 ③의 과정에서 처리된다는 것은 주변의 밝기가 밝으며 처리중인 블럭의 밝기는 어둡다는 것을 의미한다. 그러므로 이런 경우는 처리중인 블럭이 물체일 확률이 높으며 그에 따라 임계값을 높여 물체가 될 가능성을 높여야 한다. 마찬가지로 단계 ④의 과정은 임계값을 낮추어 배경이 될 가능성을 높이는 과정이다.

이런 과정에서 잡음을 최소화하도록 많은 실험을 거쳐 A, B, C, D, E의 값을 각각 0.6, 0.2, 0.2, 0.95, 0.95로 선정하여 그림자 부분의 분할이 가능

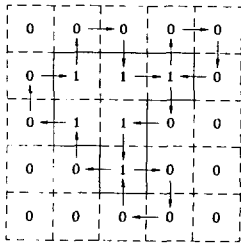


Fig. 1 Example of contour following

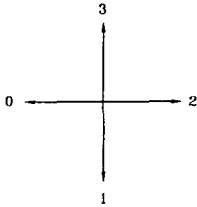


Fig. 2 Direction code

하도록 하였다.

2.2 윤곽선 추출 및 정점 추출

물체의 특징을 결정하는 변수로써 거리와 각을 이용한다. 이 변수들을 구하기 위해서는 정점을 지정해야하는데 이 정점을 찾기위해서 먼저 윤곽선을 찾는다. 본 논문에서는 Chang<sup>(3)</sup>에 있는 방향코드(direction code)를 이용하는 알고리즘으로 윤곽선을 찾는다.

또, 다각형 근사화법을 이용하여 물체의 정점을 구한다. 이 과정에서 일정한 오차 범위를 허용하면서 원래의 윤곽선과 흡사한 다각형으로 부터 다각형의 정점을 얻는다. 본 논문에서는 면적이탈에 기초를 둔 다각형 근사화법을 이용한다.

Fig. 1은 윤곽선을 찾는 예로서, 여기에서 1은 화상을, 0은 배경을 나타내고, Fig. 2의 방향코드를 이용하여 구할 수 있다.

2.3 Weighted Levenshtein Distance (WLD)

동적 스트링 매칭의 일종인 레벤슈타인 거리에 가중치를 준 Weighted Levenshtein Distance (WLD)를 이용하여 모델간의 거리를 계산한다.<sup>(4,5)</sup> 모델 A, B의 거리를 계산하는 WLD의 정의는 식 (1)과 같다.

$$WLD(A, B) = \min(pa(i) + qb(i) + rc(i)) \tag{1}$$

여기서,  $a(i)$  = replacements of changings  
 $b(i)$  = insertions  
 $c(i)$  = deletions  
 $p, q, r$  = weighting factors

이다.

본 논문에서는 weighting factor  $pa(i)$ ,  $qb(i)$ ,  $rc(i)$  중에서  $pa(i)$ 의 값만 weighting을 주고,<sup>(5)</sup> 그 정의는 식(2)와 같다.

(Def) Cost of changing corner of  $\alpha$  to the corner of  $\beta$

$$P \frac{|\theta_\alpha - \theta_\beta|}{\theta_{max}} + Q \frac{|d_\alpha - d_\beta|}{d_{max}} \tag{2}$$

여기서,  $\theta_\alpha$  = 정점  $\alpha$ 의 각  
 $\theta_\beta$  = 정점  $\beta$ 의 각  
 $d_\alpha$  = 정점  $\alpha$ 에서 다음 정점까지의 거리  
 $d_\beta$  = 정점  $\beta$ 에서 다음 정점까지의 거리  
 $P = 0.7$   
 $Q = 0.3$   
 $\theta_{max}$  = 최대 각 차이 =  $20^\circ$   
 $d_{max}$  = 최대 거리 차이 =  $0.4 d_\beta$

3. 퍼지 c-means 알고리즘

퍼지 c-means 알고리즘은 일반화된 그룹내의 여러 제곱 합을 나타내는 목적함수를 최소화 할 수 있도록 데이터 집합을 분할하는 알고리즘으로써, Bezdek<sup>(6)</sup>에 의해 소개된 뒤 순수 이론적 측면뿐만 아니라 패턴 인식의 응용 분야에서 많이 이용되고 있다.

3.1 목적함수

퍼지 c-means 알고리즘을 위한 목적함수는 식 (3)과 같다.

$$J_m(U, V) = \sum_{i=1}^n \sum_{j=1}^c (u_{ij})^m d^2(x_j, v_i), 1 \leq m < \infty \tag{3}$$

여기서,  $U = \{u_{ij}\}$ 는 소속 함수(membership function)

$V = \{v_i\}$ 는 클러스터의 중심

$X = \{x_j\}$ 는 Feature Vector

$n$  = 데이터의 갯수

$c$  = 클러스터의 수

$m = 2$

$$d^2(x_j, v_i) = \text{inner product matrix} \\ = x_j \text{와 } v_i \text{ 사이의 거리의 제곱}$$

을 나타낸다.

$m$ 이 1보다 큰 경우에 모든  $i, j$ 에 대하여  $x_j \neq v_i$ 를 만족한다고 가정하면 다음 조건식들을 만족할 때만  $(U, V)$ 가  $J_m$ 의 최소화를 가능하게 한다.

모든  $i, j$ 에 대하여,

$$u_{ij} = \left[ \sum_{k=1}^c \left\{ \frac{d^2(x_j, v_i)}{d^2(x_j, v_k)} \right\}^{1/(m-1)} \right]^{-1} \quad (4)$$

$$v_i = \frac{\sum_{j=1}^n (u_{ij})^m x_j}{\sum_{j=1}^n (u_{ij})^m} \quad (5)$$

이 알고리즘은 조건을 나타내는 식(4)와 식(5)를 포함하는 과정을 반복함으로써  $J_m$ 은 어떤 정해진 값에 수렴하게 된다.

### 3.2 데이터의 클러스터 수 결정 방법

에러 제곱합의 평가함수  $J_e$ 는  $c$ 가 증가함에 따라 단조적으로 감소한다. 그러나, 만약  $n$ 개의 데이터가 실제로  $c$ 개의 잘 분할된 클러스터로 구성되어 있다면 클러스터의 갯수가  $c$ 개일 때까지  $J_e$ 는 빠르게 감소하다가 클러스터의 갯수가  $c$ 개 이후부터는 매우 천천히 감소하다가 클러스터의 갯수가  $n$ 이 될 때  $J_e$ 의 값은 0이 된다.

식(6)은 퍼지 클러스터링 알고리즘을 위한 에러 제곱합의 평가함수이다.

$$J_e = \sum_{i=1}^n \sum_{j=1}^c (u_{ij})^m \times d^2(x_j, v_i) \quad (6)$$

## 4. String Matching 이론

임의의 물체의 형상과 이동량을 계산하기 위한 방법으로 여러가지 이론이 있지만, Grosky<sup>(4)</sup>가 제안한 스트링 매칭 이론이 어떤 이론보다 정확하고 빠르게 본 논문에서는 Grosky의 스트링 매칭 이론을 기초로 매칭을 시도하였다.

### 4.1 Model Building

스트링 매칭을 하기 위한 모델을 만들기 위하여 먼저 화상처리 기법을 이용하여 윤곽선을 추출한 후 윤곽선의 일부를 특권 스트링(privileged string)으로 만든다. 이 때, 특권 스트링을 선정하는 방법은 다음과 같다.

- (1) 256 gray level의 화상을 얻는다.
  - (2) 블러 처리 기법을 이용한 이진 화상을 얻는다.
  - (3) 윤곽선을 추적하여 그 좌표값을 구한다.
  - (4) 정점 추출 알고리즘을 이용하여 정점을 구한다.
  - (5) (4)에서 구한 정점을 이용하여 다각형 근사화(polygonal approximation)를 한다.
- 각 꼭지점들은  $V_1, \dots, V_n$ 의 정점으로 나타내어진다.
- (6) Feature Vector,  $F = F_1, \dots, F_n$ 을 구한다.

$$F_i = [A_v, d_v, X_v, Y_v]^T \quad i=1, \dots, n$$

여기서,  $A_v$ 는 정점  $V$ 의 각,

$D_v$ 는 다음 정점까지의 거리,

$X_v, Y_v$ 는 정점의 좌표,

$n$ 은 정점의 갯수

이다.

모델과 미지 물체의 매칭을 할 때 걸리는 계산 시간을 줄이기 위하여, 물체의 전체 윤곽선에서 특권 스트링(privileged string)을 택하지 않고 일부 윤곽선에서 특권 스트링을 만든다.

전체 윤곽선이 아니라 일부 윤곽선을 사용하면 가려진 물체도 인식을 할 수 있다. 여기서 선택한 특권 스트링의 갯수는 계의 가동에 큰 영향을 주는 변수이다. 특권 스트링의 갯수가 많아지면 모델 형상이 실물에 더 가까울 수가 있으나 계산 시간이 그 만큼 많아지고, 이와 반대로 작아지면 계산시간은 빨라지나 인식능력이 떨어진다. 본 연구에서는 서로 인접하는 5개의 스트링을 특권 스트링으로 선정한다.

특권 스트링을 결정했으면 이를 이용하여 다음의 순서에 따라 모델을 만든다.

- (1) 모델로 삼고자 하는 물체에서 구한 정점 중에서 각의 크기가 가장 작은 정점을 찾은 후, 그 정점으로 부터 연속되는 5개의 스트링을 특권 스트링으로 선택한다. 이는 날카로운 코너가 노이즈에 덜 민감하기 때문이다.

- (2) 그 다음 각이 작은 정점으로부터 (1)의 과정을 반복하여, 하나의 모델에 대하여 최고 5개까지의 특권 스트링을 구한 후, 이를 모델 스트링으로 만든다.

$$S = \{s_1, \dots, s_n\}$$

여기서,  $s_i$ =특권 스트링,  $i=1 \sim n$ , ( $n=5$ )

(3) 각 모델에 대하여 (1), (2)의 과정을 수행하여 모델 스트링을 구성 한다. 이 때,

$$\text{모델 스트링의 수} = \text{모델의 수} \times 5$$

가 된다.

(4) 두개의 스트링  $s_i$ 와  $s_j$ 로부터 WLD로 계산한  $s_i, s_j$ 의 거리  $\text{Dist}(s_i, s_j), \text{Dist}(s_j, s_i)$ 를 계산한다.

여기서,  $s_i, s_j \in S, s_i \neq s_j$

$\text{Dist}(s_i, s_j)$ 함수는 비대칭 함수 (nonsymmetric function) 이므로 시스템의 matching 능력을 향상시키기 위하여 대칭 함수 (symmetric cost function) 인  $\text{Sim}(s_i, s_j)$ 를 정의할 필요가 있다.

$$\text{Sim}(s_i, s_j) = \{\text{Dist}(s_i, s_j) + \text{Dist}(s_j, s_i)\} / 2 \quad (7)$$

#### 4.2 Fuzzy Clustering을 이용한 모델의 재구성

계산 시간을 줄이기 위하여 퍼지 클러스터링 이론에 의하여 모델을 재구성할 필요가 있다. 이를 위하여 먼저 feature vector를 결정한 후에 두 개의 feature vector를 이용하여 모델을 재구성한다.

(1) Feature Vector의 결정

$\max(\{\text{Sim}(s, s_i) | s_i \in S\})$  값이 최소인 모델 스트링과  $\min(\{\text{Sim}(s, s_i) | s_i \in S\})$  값이 최대인 모델 스트링을 feature vector의 토대로 결정한다. 즉, 유사성이 가장 먼 모델과 유사성이 가장 가까운 모델로 feature vector를 만든다.

앞에서 구한 두개의 string  $s_1$ 과  $s_2$ 로 구성된 feature vector( $X_i$ )는 다음과 같다.

$$\begin{aligned} X_i &= (X_{i1}, X_{i2}) \\ X_{i1} &= \{\text{Sim}(s_1, s_i) | s_i \in S\} \\ X_{i2} &= \{\text{Sim}(s_2, s_i) | s_i \in S\} \end{aligned} \quad (8)$$

(2) 모델의 재구성

식(9)와 같이 클러스터와 멤버로 이루어진 형태로 모델을 재구성한다.

$$\begin{aligned} &C_1\{m_{11} \ m_{12} \ \dots\} \\ &C_2\{m_{21} \ m_{22} \ \dots\} \\ &C_3\{m_{31} \ m_{32} \ \dots\} \\ &\dots \\ &\dots \\ &C_n\{m_{n1} \ m_{n2} \ \dots\} \end{aligned} \quad (9)$$

여기에서  $C_i$ 는 클러스터를,  $m_{ij}$ 는 멤버를 나타

낸다. (단,  $i=1 \sim n, j=1 \sim n$ )

#### 4.3 미지 물체와 모델간의 매칭

매칭모델이 결정되면 미지의 물체와 모델과의 정점을 이용하여 미지물체의 변환 벡터를 계산할 수 있다.

$$\begin{Bmatrix} X_s \\ Y_s \end{Bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{Bmatrix} X_m \\ Y_m \end{Bmatrix} + \begin{Bmatrix} T_x \\ T_y \end{Bmatrix}$$

여기서,  $X_m, Y_m$ 은 모델의 정점

$X_s, Y_s$ 는 미지물체의 정점

$T_x, T_y$ 는  $x, y$ 축 이동량

$\theta$ 는 회전량

을 나타낸다.

그러나, 위의 식은 하나의 정점에 대하여 변환을 계산한 값이므로 정확한 변환 값이라고 볼 수가 없다. 그래서 두 개 또는 그 이상의 정점에 대한 변환을 계산하여 변환 값의 오차를 보정해 나간다.

모델의 정점벡터를

$$\{(X_{m1}, Y_{m1}), \dots, (X_{mn}, Y_{mn})\}, \quad n \geq 5$$

이라 놓고,

미지 물체의 정점벡터를

$$\{(X_{s1}, X_{s2}), \dots, (X_{sn}, Y_{sn})\}, \quad n \geq 5$$

이라 놓으면 변환벡터

$$\Omega = (k \cdot \cos \theta, k \cdot \sin \theta, T_x, T_y)^T \quad (10)$$

는 다음 식과 같이 계산된다. 단,  $k$ 는 확대·축소율을 나타낸다.

$$M = \begin{bmatrix} X_{m1} - Y_{m1} & 1 & 0 \\ Y_{m1} & X_{m1} & 0 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X_{mn} - Y_{mn} & 1 & 0 \\ Y_{mn} & X_{mn} & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} X_{s1} \\ Y_{s1} \\ \cdot \\ \cdot \\ \cdot \\ X_{sn} \\ Y_{sn} \end{bmatrix}$$

오차는

$$E = M\Omega - B \quad (11)$$

가 된다.

이 때 오차의 제곱을 최소화 하는 변환벡터  $\Omega$ 를 계산하기 위하여 pseudo-inverse 방법을 사용한다.

$$\therefore \Omega = (M^T M)^{-1} M^T B \quad (12)$$

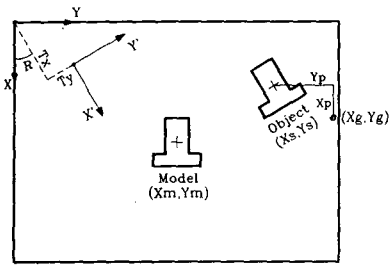


Fig. 3 Transformation of image on monitor

Fig. 3은 모니터 상에서 화상의 이동을 보여주는 것으로써, 화상의 중심이 앞에서 구한 변환 행렬에 의해 변환된 좌표값은 다음과 같다.

$$\begin{Bmatrix} X_s \\ Y_s \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} X_m \\ Y_m \end{Bmatrix} + \begin{Bmatrix} T_x \\ T_y \end{Bmatrix}$$

그리퍼의 중심과 물체의 중심간의 상대위치를 구하면 다음과 같다.

$$\begin{aligned} X_p &= X_g - X_s \\ Y_p &= Y_g - Y_s \end{aligned} \tag{13}$$

위의 좌표값에 화소와 실제거리 간의 보정계수를 곱하면 다음과 같은 실제 이동거리를 얻는다.

$$\begin{aligned} X_{mm} &= X_p \times CalX \\ Y_{mm} &= Y_p \times CalY \end{aligned} \tag{14}$$

모델을 구성한 초기 위치와 물체를 찍은 위치에서 카메라가 이루는 각의 차이  $\beta$ 를 구하면 다음과

같다.

$$\beta = (J_1 + J_2) - (J_{01} + J_{02}) \tag{15}$$

여기서,  $J_1, J_2$ 는 물체를 찍은 위치의 로봇트 조인트 각이고,  $J_{01}, J_{02}$ 는 초기 위치의 로봇트 조인트 각이다.

이 때, 실제로 물체가 회전한 각은 다음과 같이 계산된다.

$$Z = R_i - \beta + \theta \tag{16}$$

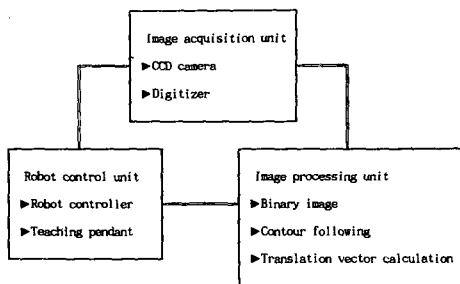
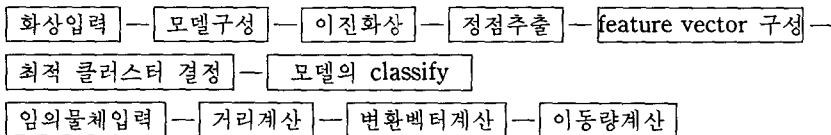
### 5. 실험 및 결과

#### 5.1 실험 장치 및 방법

패턴 인식 실험에 사용한 장치는 크게 로봇트 구동부, 화상 입력부, 화상 처리부로 나누어져 있다. Fig. 4(a)는 실험 장치의 개략을 블록 다이어그램으로 나타낸 것이고, Fig. 4(b)는 실험 장치를 나타낸 사진이다.

시편으로는 모양과 크기가 다른 M10×136 육각보울트, M8×150 육각보울트, M16, M12 육각너트, M8 사각너트를 사용하였다. 또한, 로봇트는 로봇트 콘트롤러 시스템(Robot Controller System)에 의해 작동되는 4축 로봇트를 사용하였고 모델명이 FARA SCARA ROBOT SM2이고 AC 서보 모터를 채용한 수평 다관절 로봇트이다.

로봇트 비전으로 패턴 인식을 하기위한 실험 방법의 순서는 다음과 같다.



(a)



(b)

Fig. 4 (a) Block diagram of experimental setup  
(b) Experimental setup

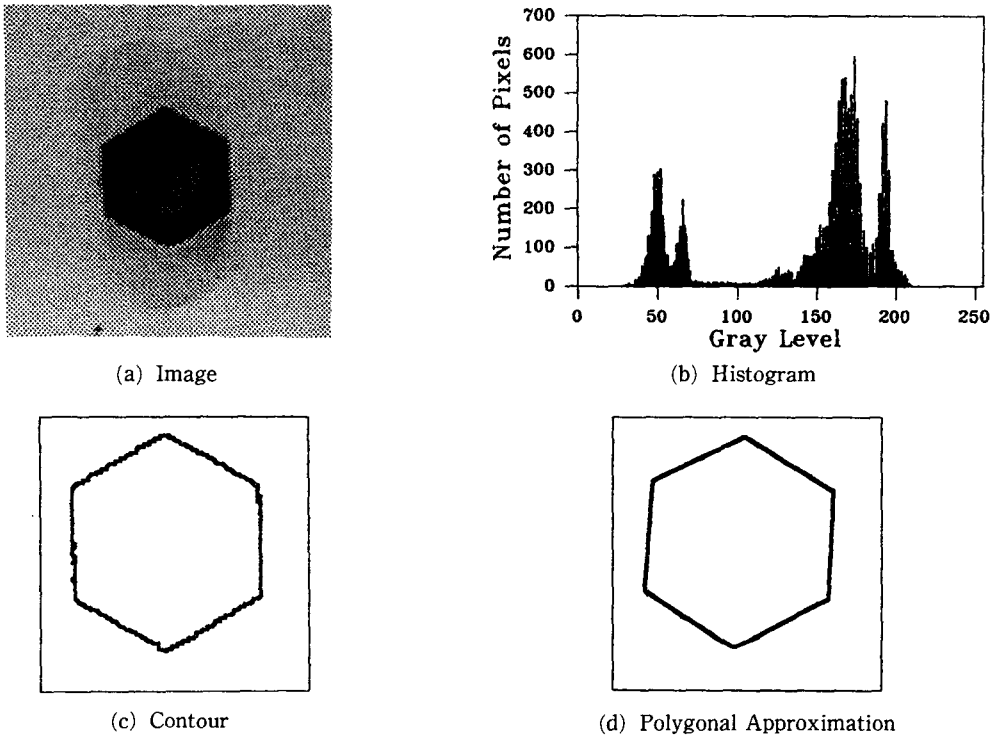


Fig. 5 Image processing results of model 1

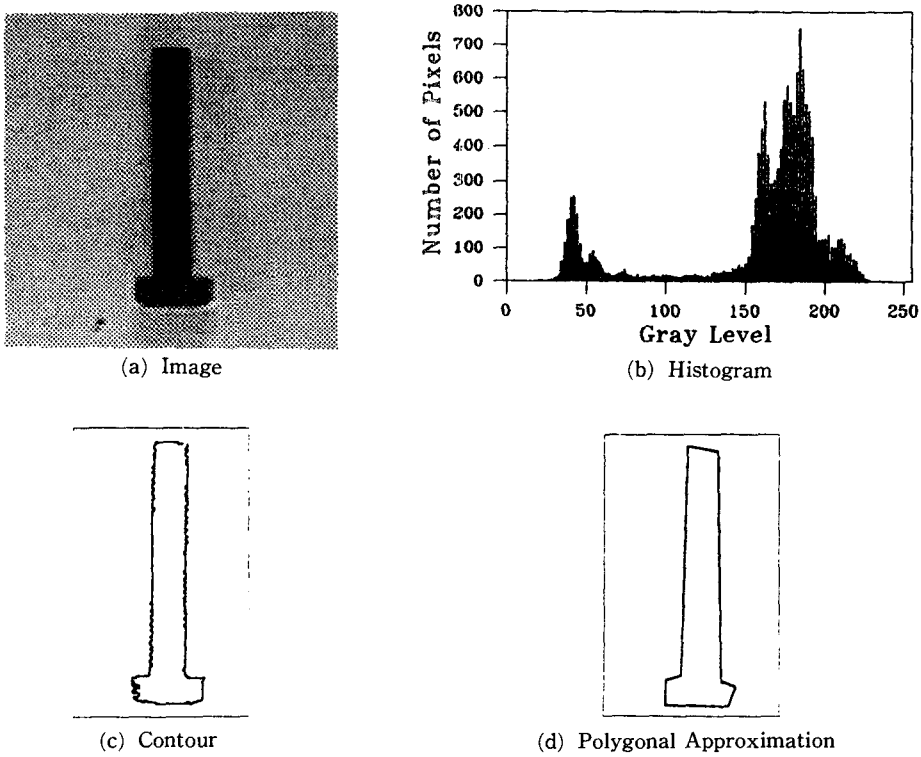
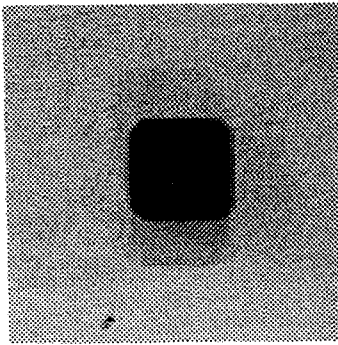
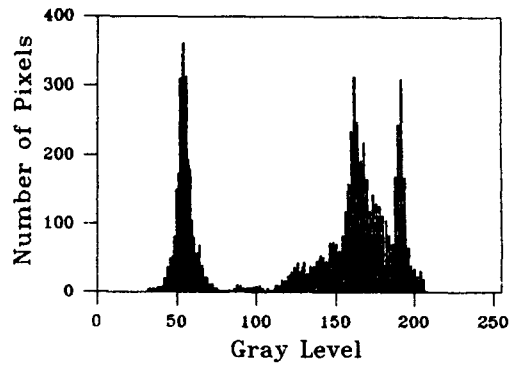


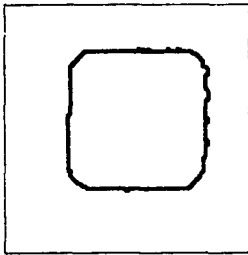
Fig. 6 Image processing results of model 2



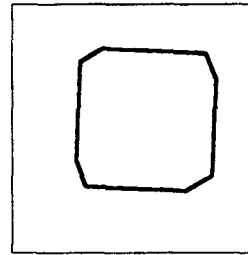
(a) Image



(b) Histogram

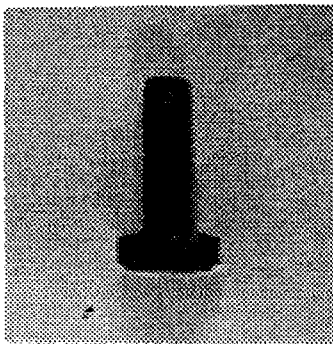


(c) Contour

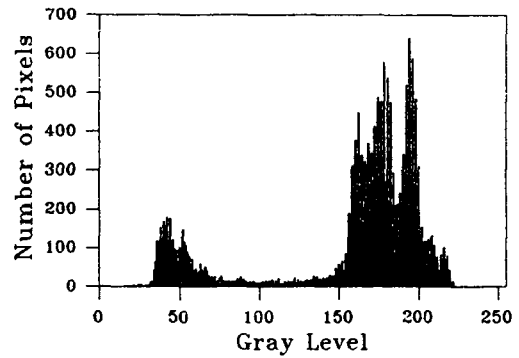


(d) Polygonal Approximation

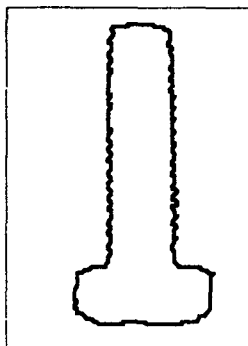
Fig. 7 Image processing results of model 3



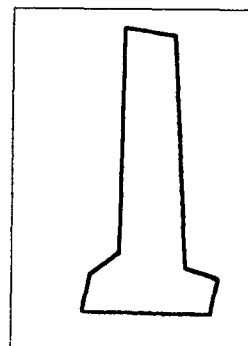
(a) Image



(b) Histogram



(c) Contour



(d) Polygonal Approximation

Fig. 8 Image processing results of model 4



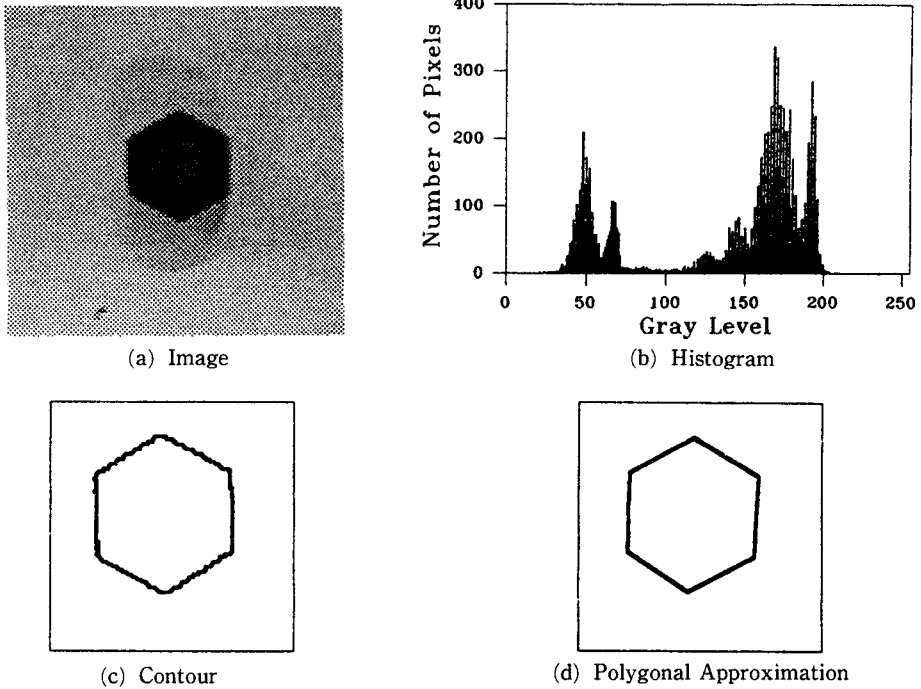


Fig. 9 Image processing results of model 5

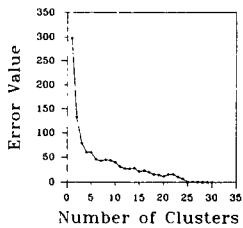


Fig. 10 Error function

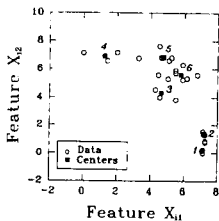


Fig. 11 Data points and centers

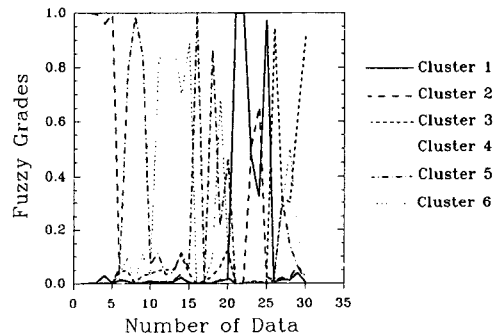


Fig. 12 Membership function

화상 모니터의 XY 상에서의 실제 길이는 서로 다르므로 각 축에 대하여 calibration을 해 주어야 한다.

계산된 calibration factor는 X축이 100/295이고 Y축의 값은 100/303이다.

### 5.2 실험 결과 및 분석

각각의 모델에 대한 히스토그램과 이진화상에 대한 윤곽선, 다각형 근사화의 결과는 다음의 Fig. 5~Fig. 9에 나타냈다.

Fig. 10은 식(6)의 평가함수에 따라 구한 에러함수로써 이를 보면 6개가 최적 클러스터의 갯수임을 알 수 있다. 또, Fig. 11은 6개 클러스터의 중심을 나타낸 그림이다. Fig. 12는 각각의 모델에 대한 소속함수를 나타낸 그림이다.

#### • 모델 화상의 결과에 대한 분석

Table 1은 Fuzzy c-means 알고리즘을 이용하여

**Table 1** Classifications of models using optimal fuzzy C-means algorithm

Class	$C_x$	$C_y$	Members	Center
1	7.131	0.207	21 22 25	21
2	7.196	1.317	1 2 3 4 5	1
3	1.245	6.909	6 10 17	17
4	4.744	6.843	7 8 9 16 18	16
5	5.801	5.612	11 12 13 14 15	15
6	5.905	6.317	19	19
7	3.266	6.787	20	20
8	7.266	0.770	23	23
9	7.265	0.855	24	24

모델을 재구성한 결과이다. 이 결과, 거리를 1~14번 계산하면 매칭을 시킬 수가 있다.

(1) 실험 1. 정적 상태에서의 패턴 매칭

모델을 촬영한 위치에서 로봇트를 움직이지 않고 카메라가 볼 수 있는 한계내에 미지의 물체를 놓고 이 상태에서의 스트링 매칭을 통하여 미지의 물체가 원점에서부터 어느 정도 병진이동과 회전이동을 했는가를 분석한다.

실험 1에 대한 히스토그램과 이진화상에 대한 윤곽선, 다각형 근사화의 결과는 Fig. 13에 나타냈다.

Table 2는 실험 1에 대한 결과를 나타낸 것으로 초기의 로봇트 좌표는 다음과 같다.

$$X=267.69 \text{ mm}, Y=101.33 \text{ mm}$$

$$J_{01}=-34.43^\circ, J_{02}=128.44^\circ,$$

$$\alpha=95.66^\circ$$

(2) 실험 2. 동적 상태에서의 패턴 매칭

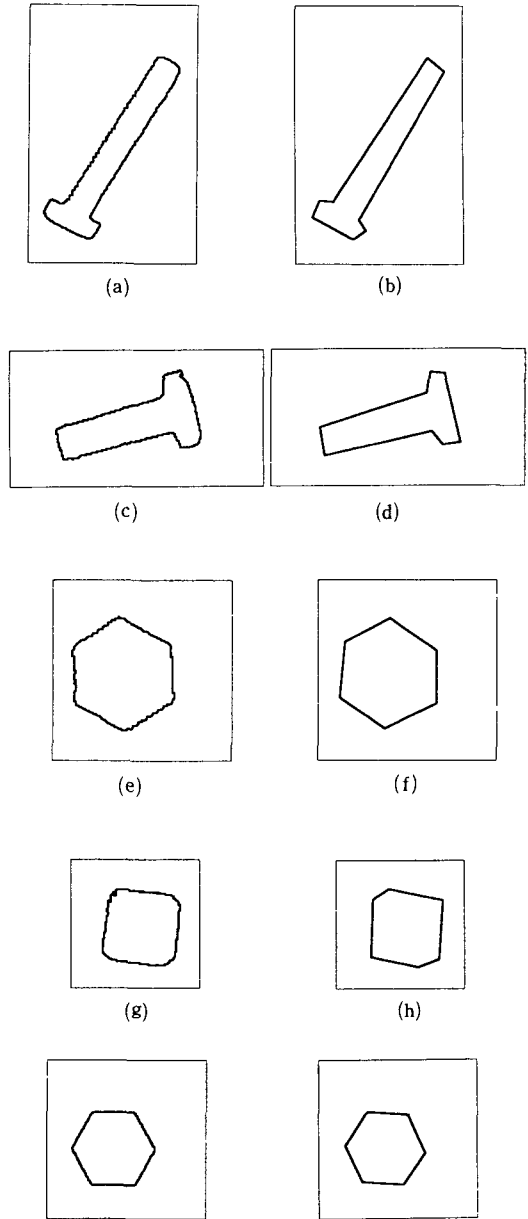
실험 1에서는 로봇트를 움직이지 않은 상태에서 화상 처리를 하고나서 로봇트를 움직였으나, 실험 2에서는 원점과 멀리 떨어져 있는 물체를 인식하기 위하여 먼저 로봇트를 구동시킨 후에 화상처리를 했다.

Fig. 14는 실험 2에 대한 히스토그램과 이진화상에 대한 윤곽선, 다각형 근사화의 결과이다.

Table 3은 실험 2에 대한 결과를 나타낸 것으로 초기의 로봇트 좌표는 다음과 같다.

$$X=324.28 \text{ mm}, Y=-11.55 \text{ mm}$$

$$J_1=-54.65^\circ, J_2=120.69^\circ, \alpha=179.14^\circ$$



(a) Contour of scene 1 (b) Polygonal approximation of scene 1  
 (c) Contour of scene 2 (d) Polygonal approximation of scene 2  
 (e) Contour of scene 3 (f) Polygonal approximation of scene 3  
 (g) Contour of scene 4 (d) Polygonal approximation of scene 4  
 (i) Contour of scene 5 (j) Polygonal approximation of scene 5

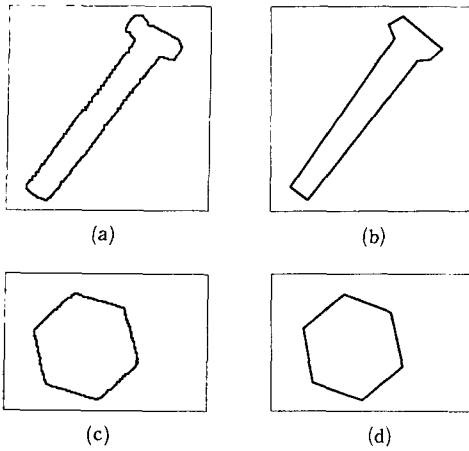
**Fig. 13** Image processing results of experiment 1

**Table 2** Result of experiment 1      단위 : X, Y(mm)  $\theta(^{\circ})$ 

SCENE		Robot position	Calculated position	Error	Matching model
SCENE1	X	227.04	279.79	2.75	6
	Y	224.96	223.09	1.87	
	$\theta$	-100.95	-101.84	0.88	
SCENE2	X	288.84	284.62	4.62	16
	Y	144.04	145.57	1.53	
	$\theta$	-146.13	31.71	177.84	
SCENE3	X	250.59	252.32	1.73	2
	Y	172.44	171.44	1.00	
	$\theta$	-132.75	-73.50	59.25	
SCENE4	X	221.09	216.22	4.87	11
	Y	133.66	129.57	4.09	
	$\theta$	-169.45	-168.34	1.11	
SCENE5	X	209.15	206.03	3.12	21
	Y	211.50	222.18	10.68	
	$\theta$	19.98	18.53	1.45	

**Table 3** Result of experiment 2      단위 : X, Y(mm)  $\theta(^{\circ})$ 

SCENE		Robot position	Calculated position	Error	Matching model
SCENE6	X	361.84	367.97	6.13	1
	Y	33.64	39.35	5.71	
	$\theta$	-24.37	-86.42	62.05	
SCENE7	X	370.65	364.97	5.88	6
	Y	106.07	112.08	6.01	
	$\theta$	43.25	45.18	1.93	
SCENE8	X	295.99	301.10	5.11	22
	Y	57.12	63.10	6.98	
	$\theta$	-48.26	-225.58	177.32	



(a) Contour of scene 6 (b) Polygonal approximation of scene 6  
 (c) Contour of scene 7 (d) Polygonal approximation of scene 7

Fig. 14 Image processing results of experiment 2

## 6. 결 론

본 논문에서는 스트링 매칭과 퍼지 클러스터링을 통합한 패턴 인식 알고리즘을 이용하여 로봇트 비전 실험을 수행하였다. 모양과 크기가 다른 시편들(M10×136 육각보울트, M8×150 육각보울트, M16, M12 육각너트, M8 사각너트)을 대상으로 패턴 인식 실험을 수행한 결과에서 다음과 같은 결론을 얻었다.

(1) 퍼지 클러스터링을 이용하여 모델들을 재구성하는 제안된 알고리즘은 미지 물체와 모델과의 매칭을 위한 거리 계산의 횟수가 모델을 계층적으로 찾는 기존의 스트링 매칭 알고리즘보다 감소된다.

(2) 본 논문에서 선정한 시편들을 모델로 실험한 결과, 로봇트를 움직이지 않은 상태와 로봇트를 움직인 상태에서 미지 물체에 대한 패턴 인식 결과를

비교하면, 전자의 경우가 직선 및 회전 이동량에 대한 오차가 작다.

## 참고문헌

- (1) Rafael, C. Gonzalez and Paul Wintz, 1987, "Digital Image Processing," Addison-Wesley Publishing Company
- (2) Fu, K. S. and Gonzalez, R. C. and C. S. G. Lee, 1987, "Robotics : Control, Sensing, Vision and Intelligence," McGraw-Hill Book Company
- (3) Shi-Kuo Chang, 1989, "Principles of Pictorial Information Systems Design," Prentice-Hall International, Inc.
- (4) Rajiv Mehrotra and William I. Grosky, 1989, "Shape Matching Utilizing Indexed Hypotheses Generation and Testing," IEEE Transactions on Robotics and Automation. Vol 5, No 1, February, pp. 70~77.
- (5) Wen-Hsiang Tsai and Shiao-Shian Yu, "Attributed String Matching with Merging for Shape Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, NO 4, July, pp. 453~462, 1985.
- (6) James, C. Bezdek, 1981, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York.
- (7) Li, P. G. and Wu, S. M. 1988, "Monitoring Drilling Wear States by a Fuzzy Pattern Recognition Technique," Journal of Engineering for Industry, Vol 110, August, pp. 297~300.
- (8) 이태원 외, 1991, "확장된 Fuzzy Clustering 알고리즘을 이용한 자동 목표물 검출," 전자공학 회논문지, 제28권 B편 제 10 호, pp. 96~105.
- (9) 이명구, 1991, "피부 반응 검사의 자동화를 위한 영상 처리 알고리즘에 관한 연구," 석사학위논문, 연세대학교 대학원, 서울