

## 가중치를 준 특징벡터에 의한 2차원 패턴의 인식에 관한 연구

이태원\*\*·이규태\*·박청환\*

### A Study on the 2-Dimensional Pattern Recognition using the Weighted Feature Vector

Rhee Tae Won\*\*, Lee Kyu Tae\* and Park Cheong Hwan\*

#### Abstract

This paper describes a 2-dimensional pattern recognition algorithm which used critical points extracted from the concavity tree.

Conventional pattern recognition method using a concavity tree is likely to make an error when the convex features of the input pattern are to be extracted from the portions of a straight line or simple curve, which is because the most algorithms give much higher priority to the concave part of the pattern.

This paper proposes the algorithm which is designed to reduce the chance of errors by extracting the critical points in both convex and concave parts of the object pattern regardless of the object pattern's rotation, translation and scaling.

#### I. 서 론

특징에 의해서 패턴을 인식하는 과정은 먼저 입력 패턴에서 그 패턴을 잘 표현할 수 있는 중요한 특징들을 추출하고, 이들을 이용하여 그 패턴의 부류를 결정한다. 예를 들면, 의학진단의 경우에 입력 패턴은 병의 징후들이고 결과는 병의 종류이다. 문자인식의 경우에 입력 패턴은 알려지지 않은 문자이고 결과는 그 문자의 명칭이다. 음성인식의 경우 입력 패턴은 청각파형이고 결과는 그 단어의 명칭이다[1].

패턴의 특징이란 임의의 패턴이 다른 패턴과 분류될 수 있는 그 패턴의 고유한 양으로서 인간의 경우에는 시각, 청각, 촉각, 미각 등으로 지각된다. 이 중에서 시각을 통하여 얻은 데이터는 인간에 있어 가장 복잡하면서도 가장 유용한 특징이다[2]. 따라서, 이러한 시각기능을 가진 시스템을 개발하기 위한 연구가 활발히 진행되고 있으며 실제로 로봇공학, 공장 자동화 등에 응용되고 있다[3-6].

물체인식 시스템의 입력 패턴은 대부분이 3차원이나,

(1990. 1. 22 접수)

\*대학원 전자공학과

\*\*디지털시스템연구실, 교수

3차원 패턴을 처리하기 위해서는 엄청난 양의 데이터를 처리해야 하므로 3차원의 패턴을 카메라 등의 입력장치로 받아 들여 2차원 패턴으로 처리하여 인식할 수 있다. 예를 들면, Conveyer에 놓여 있는 공구나 부품들을 수직으로 설치된 Camera로 받아들일 경우 실제로는 3차원의 물체이나 입력된 2차원 패턴에서 물체의 외부 모양을 추출하여 인식할 수 있다. 따라서 2차원 패턴의 인식은 패턴 인식의 기본적인 중요한 분야로써 많은 알고리즘들이 발표되어 왔다. 이들은 크게 패턴의 전체를 탐색하여 인식하는 경우와 패턴을 특징을 가지고 있는 부분들로 나누어 인식하는 경우로 나눌 수 있는데, 전자의 경우로는 Fourier Descriptor, Template 등이 대표적인 방법인데 패턴이 겹쳐서 일부분이 가려진 경우, 회전된 경우, 크기의 변화가 있는 경우에는 그 효율성이 감소하고, 인식시간도 많이 걸린다[7, 8]. 후자의 경우로는 Curvature, Critical Point 등이 대표적인 방법으로 패턴이 가려진 경우, 회전된 경우 및 크기의 변화가 있는 경우에도 좋은 인식율을 보인다[9-11]. 특히, Critical point에 의한 방법은 인식시간과 메모리의 제약이 있는 경우에도 유용한데 잡음에 민감하다는 단점이 있다[12]. 이와 같이 살펴볼 때 시간과 메모리의 제약이 있는 상황에서 회전 및 크기의 변화가 있는 2차원 영상을 인식하

는 데는 잡음에 대한 민감도를 줄인 Critical point 에 의한 인식방법이 가장 효율적인 방법이다.

본 연구에서는 입력 패턴의 회전, 크기변화 및 요곡의 유/무에 관계없이 2차원 영상을 인식할 수 있는 특징점에 의한 패턴인식 알고리즘을 제안하고 실험을 통해서 그 효율성을 입증하였다.

## II. 특징점에 의한 패턴인식

### 2.1 Concavity Tree 에 관한 고찰 [13]

2차원의 패턴은 각각의 패턴에 따른 고유한 특징부분이 있는데 이를 특징점(Critical point)이라 하고 이러한 특징점을 추출하여 패턴을 인식하는 방법을 특징점에 의한 패턴인식이라 한다.

패턴의 특징점은 각 패턴에서 요곡부분만을 추출해 내고 추출된 요곡부분을 간략하면서도 정확하게 표현하는 점을 찾아냄으로써 정할 수 있다. 따라서, 특징점을 찾기 위해서는 먼저 패턴의 요곡부분을 추출하는 과정이 선행되어야 한다. 본 논문에서는 패턴의 요곡부분을 추출하기 위해서 Concavity Tree 라는 요곡 패턴의 기본적 구조를 정의한다. Concavity Tree 는 원패턴에서 그 패턴의 요곡 패턴을 추출하고 추출된 요곡 패턴에서 또다시 요곡 패턴을 추출하는 과정을 반복적으로 수행하여, 원패턴을 중요한 요곡정보를 갖는 부분으로 나누어 구성하는 과정을 나타내는 것으로 이를 도식적으로 나타내면 Fig. 1과 같다.

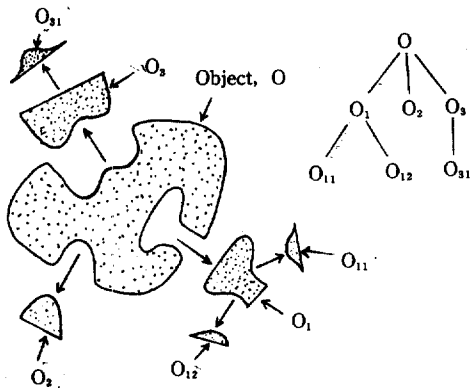


Fig. 1. The concavities and the concavity tree of patterns O.

### 2.2 특징점 추출

#### 2.2.1 특징점의 정의 및 추출방법

이상적인 경우로 대상체영상의 경계가 연속적 경계

(Continuous Boundary) B를 갖는 패턴에 대한 Concavity Tree 구조를 이용한 특징점 추출은 다음과 같다.

가) 단계 1의 특징점 추출

단계 1의 특징점은 단계 1의 요곡 패턴  $S_1$ 에서 추출된다. 각각의  $S_{i1}$ 에서  $(B \cap S_{i1})$ 을 계산하면 2차원의 Line Segment 가 얻어지는데 얻어진 Line Segment 의 양 끝점이 특징점이다. 이렇게 얻은 모든  $S_{i1}$ 의 양 끝점의 집합을  $C_1$  즉,  $C_1$ 은 단계 1에서 얻은 특징점의 집합이다.

$$C_1 = \bigcup_{i=1}^N [ \text{endpoints of line segments } (B \cap S_{i1}) ]$$

나) 후속 단계의 특징점 추출

후속 단계의 특징점은 각 단계의 요곡영상에서 추출되며 추출 방법은 단계 1과 같다. 이렇게 해서 특징점의 집합  $C_1, C_2, C_3 \dots C_N$ 이 얻어지는데 이러한  $C_i$ 는 패턴 S 에 대한 특징점의 총합을 형성하고 특징점의 총합을 C 라 하면

$$C = \bigcup_{i=1}^N C_i \quad (1)$$

이다. 여기서 N 은 단계의 개수이다.

#### 2.2.2 특징점의 저장

추출된 특징점은 인식하고자 하는 패턴 각각의 고유한 명칭과 함께 저장되어 데이터베이스를 구성하는데, 특징점을 저장할 때는 특징점이 추출된 위치 즉, 특징점의 (X, Y) 좌표값 및 추출된 단계를 함께 저장한다.

### 2.3 특징벡터 구성

특징벡터는 Sixtuple  $j_x$ 로 표시되며 j는 주어진 패턴의 j번째 특징벡터를 나타내고 이 벡터의 i번째 요소는 식 (2)로 표시된다.

$$j_x = \frac{(X_i - \bar{X})}{S_j} + \frac{(Y_i - \bar{Y})}{S_j} \quad (2)$$

$$S_j = \sum_{i=1}^q \sqrt{(X_i - \bar{X})^2 + (Y_i - \bar{Y})^2} \quad (3)$$

여기에서,  $\bar{X}, \bar{Y}$ 는 각 특징벡터를 구성하는 특징점의 중심좌표이며 식 (3)의  $S_j$ 는 크기변수로서 j번째 특징벡터의 에너지를 나타낸다. 식 (3)에서 q는 특징벡터 구성에 포함된 특징점의 수이다.

### 2.4 특징벡터에 의한 인식

#### 2.4.1 특징벡터 매칭과정

특징벡터의 매칭과정은 두 가지로 나눌 수 있다. 첫번째는 특징벡터를 구성하고 있는 특징점들의 추출된 단계

를 매칭시키는 'Level Code 벡터' 매칭이고, 두번째는 매칭된 'Level Code 벡터'의 실제 매칭정도를 판단하는 '실제 특징벡터 (Actual Feature Vector)' 매칭과정으로 나눌 수 있다.

가) Level Code 벡터 매칭

Level Code 벡터 매칭은 실수영역에서의 계산을 줄이기 위한 과정으로 특징벡터가 매칭되면 언제나 그 특징벡터를 구성하는 특징점들의 추출된 단계도 서로 동일해야 한다는 점에 근거를 둔 것이다. Level Code 벡터는 특징벡터에 포함된 특징점들의 추출된 단계로 구성되며 Level Code 벡터의 매칭함수는 식 (4)와 같고 매칭이 이루어지면  $D_i$  값이 0이다.

$$D_i(m, n) = \sum_{k=1}^p \{ \frac{d_k}{mL_i - nL_i} \} \quad (4)$$

이 때,  $d_k$ 는 데이터베이스를,  $p$ 는 입력된 패턴을 나타내며,  $m, n$ 은 각각의 벡터들의 순서를 나타낸다. 또,  $L_i$ 는 해당 특징점의 추출된 레벨값으로 정수치이다.

나) 실제 특징벡터 매칭

매칭된 Level Code Vector Set 에 한해서 식 (5)와 같은 실제 특징벡터 매칭함수를 적용하여 실제 특징벡터 매칭을 행한다.

$$d_i(m, n) = | \sum_{k=1}^p Z_i | - | \sum_{k=1}^p Z_j | \quad (5)$$

첨자들은 앞에서 정의했던 것과 같으며,  $\{d_i(m, n) < T_m\}$ 이 ( $i=1$ )에서 ( $i=q$ )까지 모두 성립할 때 해당벡터가 매칭된 것이다. 여기서  $T_m$ 은 인식해야할 영상들의 성질과 요구되는 시스템의 정확도에 따라 실험치로 정해 주어야 한다.

2.4.2 매칭률 계산

인식대상의 패턴들이 전체가 요곡으로 이루어진 패턴인 경우에 매칭률은 간단하게 발생한 특징점과 매칭된 특징점의 비로써 계산할 수 있다.

$$\text{매칭률} = (\text{매칭된 특징점의 수} / \text{추출된 특징점의 수}) \times 100(\%)$$

2.4.3 회전율, 크기변화를 및 위치변화를 계산

회전율:

$$\theta = (\sum \{ \arg (d_k^* Z) - \arg (e_n Z) \}) / \text{매칭된 벡터수} \quad (6)$$

크기변화율:

$$\rho = (\sum (e_n S) / (d_k^* S)) / \text{매칭된 벡터수} \quad (7)$$

위치변화율:

$$T_x = (G_m x - G_p x), \quad T_y = (G_m y - G_p y) \quad (8)$$

III. 제안된 특징점에 의한 패턴인식

제 2장에서 살펴본 Concavity Tree를 이용한 특징점에 의한 패턴인식은 인식대상 패턴의 요곡부분만을 중점적으로 분석하여 특징점을 추출하므로 인식대상 패턴은 Fig. 2와 같이 많은 비요곡부분을 갖는 패턴을 인식할 때는 그 정확도가 떨어진다. 또, 추출된 특징점과 매칭된 특징점의 비로써 매칭률을 계산하는 방법은 패턴의 전체가 요곡으로 이루어졌을 경우에는 매우 간단하고 좋은 방법이나 많은 비요곡부분을 갖는 패턴에 대해서는 부정확한 방법이다.

본 장에서는 위에서 제시된 문제점을 해결하기 위하여 비요곡부분도 고려하는 특징점 추출방법과 벡터에 가중치를 준 매칭률 계산방법을 제안했다.

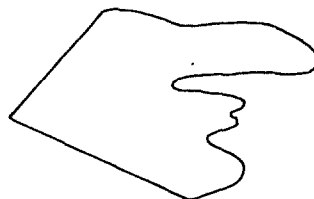


Fig. 2. A pattern having convex part

3.1 특징점 추출

입력된 패턴의 Contour Line의 좌표열에서 특징점을 추출하는데 특징점은 입력 패턴을 Concavity Tree로 나타내어 각각의 concave part의 소좌표열을 얻어 다음의 문턱치를 적용하여 정한다.

T1: 잡음에 의한 요곡에 대비한 문턱치

T2: 특징점을 정하기 위한 문턱치

T1을 이용하여 주어진 소좌표열이 요곡부분인지를 판단하여 요곡부분일 때는 소좌표열의 Convex hull로부터의 수직거리가 T2에 가장 가까운 점을 택하고 비요곡부분일 때는 소좌표열의 양 끝점을 특징점으로 택한다. 특징점을 추출하는 상세한 과정을 Fig. 3에 나타내었다.

3.2 추출된 특징점의 저장

위와 같은 과정을 거쳐 추출된 특징점은 특징벡터 구성 및 인식과정을 위해서 저장되어야 한다. 특징점은 패턴의 Contour Line의 좌표열과 같은 순서로 각각의 추출된 단계와 좌표값이 함께 저장되며 인식해야 하는 패턴의 각각마다 고유한 명칭을 붙여서 특징점의 X 좌표값, Y 좌표값, 그리고 추출된 단계의 순으로 저장된다.

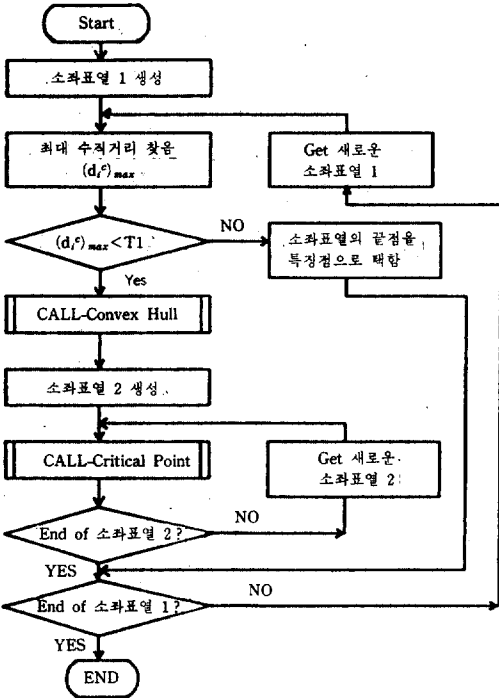


Fig. 3. Flowchart of critical point extraction

3.3 가중치를 준 벡터에 의한 매칭률 계산

일반적으로 인식해야 할 패턴의 전체가 요곡부분으로 구성되어 있을 때 매칭률은 입력 패턴에서 발생한 특징점의 수와 데이터베이스의 어느 특정 패턴과 매칭된 특징점의 비로써 매우 간략하고, 정확하게 계산될 수 있을 것이다. 그러나, 인식해야 할 패턴들이 Fig. 4와 같이 요곡부분은 같고 비요곡부분만이 다를 때 위와 같은 매칭률 계산방법은 부정확하다.

일반적으로, 비요곡부분이 포함된 패턴의 특징점은 비요곡부분보다 요곡부분에서 월등히 많은 특징점이 추출되므로 입력 패턴이 데이터베이스의 특정 패턴과 동일한 요곡부분을 갖는다면 입력 패턴에서 추출된 특징점들의 많은 부분이 데이터베이스 패턴과 매칭되므로 Fig. 4와 같은 패턴의 매칭률은 서로 상이한 패턴임에도 불구하고 매우 높은 매칭률을 보인다. 그러므로, 비요곡부분을 갖는 패턴에 대한 매칭률 계산은 매칭된 벡터에 가중치를 주어 크기가 큰 벡터가 매칭되었을 때는 그와 비례한 매칭률의 증가가 있어야 한다. 이러한 방법을 사용하면 Fig. 4의 패턴을 매칭시켰을 경우 비요곡부분에서 추출된 벡터는 매칭되지 않고, 매칭되지 않은 벡터는 상대적으로 큰 벡터들이기 때문에 두 패턴이 서로 유사하지 않음을 정확히 인식할 수 있다. 이러한 개념에 기본을 두어

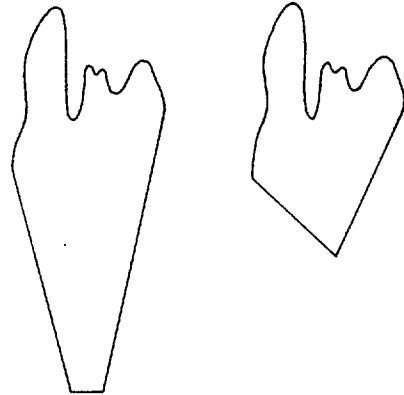


Fig. 4. Different patterns which have same concave

제안된 알고리즘에서는 특징벡터의 매칭을 시작하기 전에 입력 패턴의 좌표열과 같은 크기의 Array를 만들어 각각의 값을 "0"으로 주고 벡터가 매칭되면 매칭된 벡터에 포함된 좌표값들의 위치에 해당하는 Array 값들을 "1"로 두었다. 이에 대한 상세한 과정을 Fig. 5에 나타내었다.

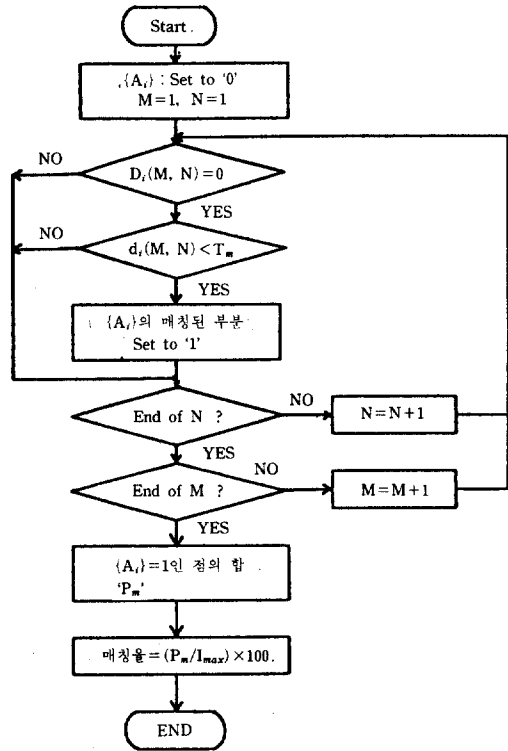


Fig. 5. Flowchart of matching procedure with weighted feature vector

### IV. 실험 및 검토

#### 4.1 모델 패턴의 선택

모델 1과 모델 2는 제안된 알고리즘이 특징벡터에 가중치를 두어 매칭을 하므로써 요곡부분은 같고 선형부분만이 서로 다른 패턴을 분류할 수 있음을 증명하기 위한 것이고, 모델 3과 모델 4는 제안된 알고리즘이 요곡의 유무에 관계없이 전체가 요곡으로 이루어진 패턴 및 요곡이 전혀 없는 패턴을 구분할 수 있음을 보이기 위한 것이다. 또, 모델 5는 두 개의 패턴이 서로 겹쳐 있을 때 각각의 부분적인 패턴이 모델의 어느 것에서 파생된 것인지를 분류할 수 있음을 보이기 위한 것이다.

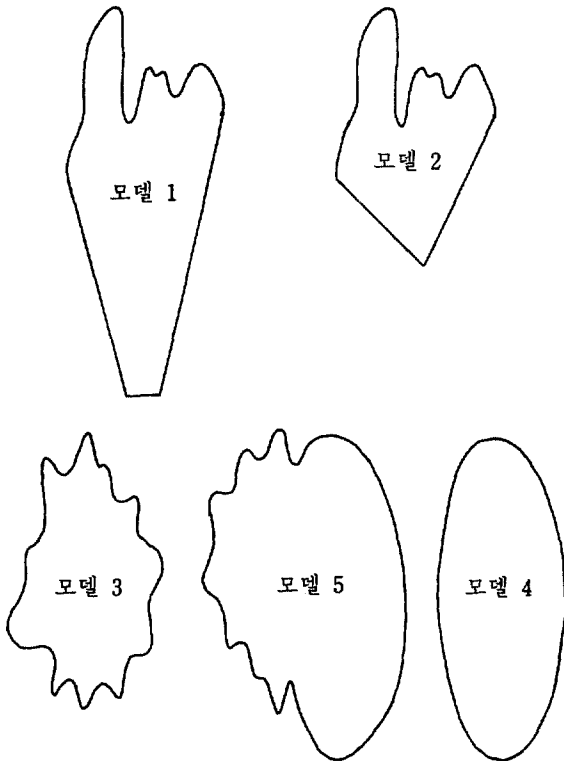


Fig. 6. The sample patterns

#### 4.2 제안된 알고리즘의 타당성 검증

##### 4.2.1 요곡의 유무 및 회전, 크기변화, 위치변화에 독립적임을 검증

제안된 알고리즘이 요곡의 유무 및 회전, 크기변화, 위치변화에 관계없이 패턴을 구분할 수 있음을 증명하기 위해서 위치, 크기의 변화 및 회전된 모델 1, 모델 3, 모델 4를 입력하였을 때 각각의 매칭률과 회전을 및 크기변화율, 위치변화율을 나타내었다. 매칭률 계산식은

식 (9)와 같다.

$$\text{매칭률} = (P_m / I_{max}) \times 100(\%) \quad (9)$$

회전을 등의 표기방법은 실제 변화량을 오른쪽에 나타내고 측정된 값을 왼쪽에 나타내었다. 예를 들면, 위치변화율에서 (48/50, 0/2)는 X 좌표축으로 50 이동되고, Y 좌표축으로 2만큼 이동된 패턴이 계산치로는 각각 48, 0만큼 이동된 것으로 계산되었음을 의미한다.

Table 1. Rotation, translation, scaling and matching rate for model 1, model 2, model 4

	특징점수	매칭율	회전각	크기변화	위치변화
모델 1	32	93	42/45	0.99/1	48/50, 0/2
모델 3	73	0	.	.	.
모델 4	28	0	.	.	.

입력패턴 : 모델 3

	특징점수	매칭율	회전율	크기변화	위치변화
모델 1	32	0	.	.	.
모델 3	73	95	46/45	0.82/0.8	1/0, -3/0
모델 4	28	5	.	.	.

입력패턴 : 모델 4

	특징점수	매칭율	회전율	크기변화	위치변화
모델 1	32	0	.	.	.
모델 3	73	8	.	.	.
모델 4	28	98	46/45	0.91/0.9	2/0, 21/25

##### 4.2.2 가중치의 타당성 검증

가중치의 타당성을 검증하기 위하여 모델 2 및 모델 3과 모델 5에 대해서 가중치를 주었을 때와 주지 않았을 때의 매칭률을 구했다. Table 2에서 볼 수 있는 것처럼 가중치를 주지 않았을 때의 매칭률이 훨씬 더 크게 산출됨을 알 수 있는데, 모델 1과 모델 2는 모델 1을 기준으로 볼 때 52%가 같고, 모델 3과 모델 5는 모델 3을 기준으로 볼 때 60%가 같은 패턴이다. 따라서, 가중치를 주었을 때의 매칭률이 실제 두 패턴의 유사도에 근접한 매칭률을 산출함을 알 수 있다.

Table 2. Measured and real matching rate with weighting factor

	가중치유	가중치무	가중치유	가중치무
모델 1	93(%)	95(%)	모델 3	99(%) 98(%)
모델 2	50(%)	84(%)	모델 5	66(%) 75(%)

##### 4.2.3 겹쳐진 패턴에 대한 매칭여부 검증

두 영상이 겹쳐 (Occluded) 있을 때 영상의 부분 부분

이 모델의 어느 영상에서 파생되었는지를 찾을 수 있다. 본 논문에서 제안한 알고리즘의 이러한 특성을 보이기 위해서 모델 3의 요곡부분과 모델 4의 일부분으로 혼합된 영상모델 5를 입력영상으로 하여, 모델 3의 부분과 매칭된 영역을 '사각형'으로 표시하였으며, 모델 4와 매칭된 영역을 '원'으로 표시하도록 하였을 때 Fig. 7과 같이 각각의 부분에 대해서 정확하게 매칭함을 알 수 있다.

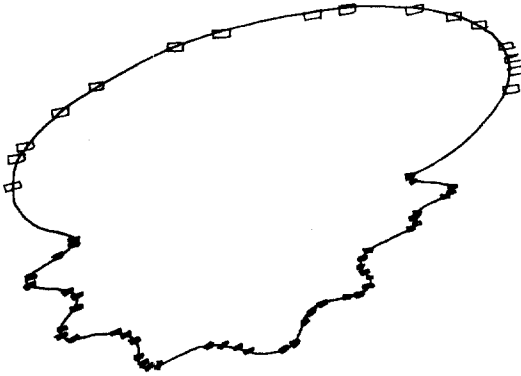


Fig. 7. Matching of the occluded pattern

## V. 결 론

본 연구에서는 회전, 크기변화 및 위치변화에 독립적이고 요곡의 유무에 관계없이 패턴 인식 알고리즘을 위한 특징점 추출 및 특징벡터 구성에 관한 알고리즘을 제안하고, 이를 실험을 통하여 검증하였다.

실험결과 회전, 크기변화 및 위치변화가 있는 각각의 입력 패턴들을 5% 오차 이내에서 정확하게 분류할 수 있었고, 비요곡 부분을 갖는 입력 패턴 및 요곡이 전혀 없는 패턴이 입력되었을 경우에도 95% 이상의 매칭률로 패턴을 정확히 인식할 수 있었다. 특히 요곡부분이 같은 두 개의 패턴에 대한 실험결과에서 실제로 52%가 동일한 두 패턴에 대한 매칭률이 50%로 계산됨을 확인하여 제안된 알고리즘이 크기, 위치, 회전 및 요곡의 유무에 관계없이 입력 패턴을 분류할 수 있음을 확인하였다.

## 참고문헌

1. J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*. Addison-Wesley (1974).
2. R. Nevatia, *Machine Perception*. Prentice-Hall, (1982).
3. A. Pugh, *Robot Vision*, U.K., (1983).
4. R.C. Gonzalez and R. Safabackhsh, "Computer Vision Techniques for Industrial Application and Robot Control", *IEEE Trans. Comput.*, pp. 16-32, Dec. (1982).
5. R.P. Kruger and W.B. Thompson, "A Technical and Economic Assessment of Computer Vision for Industrial Inspection and Robot Assembly", *Proc. of IEEE*, **69**, Dec. (1981).
6. D.H. Ballard and C.M. Brown, *Computer Vision*. Prentice-Hall, (1982).
7. Zahn, Charles, T. and Roskies, Ralph, Z., "Fourier Descriptor for Plane Closed Curves", *IEE Trans. on Computers*, Vol. C-21, No. 3, 269-281, March. (1972).
8. A. Rosenfeld and A. Kak, *Digital Picture Processing*. Academic Press (1972).
9. J.L. Turney, T.N. Mudge and R.A. Volz, "Recognition Partially Occluded Parts", *IEEE Trans. on Pattern Analysis and Machine Intell.*, Vol. PAMI-7, No. 4, pp. 410-421, July (1985).
10. H.C. Lee and K.S. Fu, "3-D Shape from Contour and Selective Confirmation", *IEEE Computer Society Proceedings of Work-shop on Computer Vision: Representation and Control*, pp. 162-170, Aug. (1982).
11. Y.J. Tejwani and R.A. Jones, "Machine Recognition of Partial Shapes Using Feature Vectors", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-15, No. 4, pp. 504-516, July/August. (1985).
12. B.K. Miller and R.A. Jones, "Reliable Formation of Feature Vector for 2-D Shape Representation", *Computer Vision for Robots*, **595**, 109-118, (1985).
13. J. Sklansky, "Measuring Concavity on a Rectangular Mosaic", *IEEE Trans. on Computer*, Vol. C-21, No. 12, pp. 1355-1364, Dec. (1972).