

축 벡터 (pivot vector) 와 적응 수렴 계수 (cacf) 를 사용한 2차원 형상 인식

○ * ** *
하 동 수, 이 상 욱, 성 공 모

* 서울 대학교 전자 공학과 ** 서울 대학교 제어 계측 공학과

2-D Shape Matching using pivot vector and componentwise adaptive convergence factor

* ** *
Dongsoo Har, Sanguk Lee, Koengmo Sung

* Dept. of Electronics Eng. ** Dept. of Control & Instrumentation Eng.
Seoul National University

서울 대학교 전자 공학과

Abstract

2-D shape matching algorithm is proposed which uses pivot vector and componentwise adaptive convergence factor. It is proved that proposed algorithm has better result than any other algorithm. It is to be expected that this algorithm will work effectively in 3-D shape matching.

I. 서 론

2차원 형상 인식은 물체 인식 (object recognition)의 한 가지 기법으로 많은 연구가 이루어져 왔다. 형상인식에서는 영역분할 (segmentation)과 다각형 근사화 (polygonal approximation)를 거쳐 얻은 꼭지점들의 연계성을 이용하거나, 영역분할 후에 바로 경계선의 구조를 사용하여 물체들의 모양을 비교한다. 형상인식 알고리즘은 여러가지가 나뉘며, 이들은 크게 국부적 (local) 특성과 전역적 (global) 특성을 이용하는 두가지로 나뉘 수 있다. 이 중 Bir Bhanu[1]의 순위적 (hierarchical) 형상인식은 인식가능한 물체 변형의 범위가 넓을 뿐 아니라, 갑진 물체의 경우에 다른 알고리즘에 비하여 보다 체계적이고 효과적인 처리를 할 수 있다. 그러나 [1]에서 다음과 같은 문제점이 제기된다.

- (1) 전체 목적 함수의 기울기에 물체의 각 선분이 갖는 영향은 극부적인 2 - 3개의 선분에만 있을 뿐이다.
- (2) 수렴 계수를 1회의 확률 배정 (probability assignment)에 전체 선분에 대하여 균일하게 취하여 올바른 기울기의 효과가 확률 배정에 나타나지 못한다.
- (3) 잡음이 있을때의 인식 결과의 차이가 작지만 인식 결과에 대해 신뢰를 가질 수 없다.
- (4) 위험 함수 (penalty function)를 사용한 경우에도 정확한 인식 결과를 갑진 물체의 구성 물체 개별 분리에서 얻을 수 없다.
- (5) 순위법 형상인식은 선험적 지식을 필요로 하여 실용에 어려움을 준다.

[1]의 (1) - (4)의 보완을 위해 제안된 알고리즘은 다음에 유의하였다.

- (1) 확률 수렴이 가장 빠른 2개의 꼭지점을 물체에서 뽑아내어 2개의 꼭지점이 만드는 축 벡터를 중심으로 다른 선분들의 인식을 한다.

- (2) 수렴 계수를 각 선분이 갖는 확률 벡터의 성분예 따라 달리 하여 정확한 확률 배정을 한다.
- (3) 축 벡터를 중심으로 한 선분인식에 ripple을 사용하여 영역분할과 다각형 근사화에서 발생한 예러블 보상 (compensation) 한다.
- (4) 갑진 물체의 개별 분리에서는 축 비교물체를 중심으로 물체비교를 한다. 축 비교물체는 축 벡터가 먼저 생긴 물체를 가리키는 것이다.
- (5) 1번 순위와 2번 순위의 효과를 하나로 묶는다.

본 논문에서는 II. 전체 함수의 형성, III. 기울기 추정 알고리즘과 확률 수렴, IV. 축 벡터를 중심으로 한 확률 수렴 V. 실험 및 결과, VI. 결론의 순서로 제안된 알고리즘의 (1) - (5)의 구현을 살펴보기로 한다.

II. 전체 목적 함수의 구조

1. 확률 벡터
다각형 근사화된 비교물체를 T라 하고 역시 다각형 근사화된 비교되는 물체를 O라 하면

$$T = (T_1, T_2, \dots, T_n) \quad (2.1)$$

$$O = (O_1, O_2, \dots, O_m) \quad (2.2)$$

와 같이 T는 N개의 선분으로 O는 M개의 선분으로 된다. 다각형 근사화된 비교 물체의 선분을 비교되는 물체의 해당 선분에 배정하는 것이 인식의 과정이 된다. 이 때 비교물체의 각 선분은 비교되는 물체의 각 선분에 대하여 초기 인식 확률을 갖는다. 예로 비교물체의 1번 선분이 비교되는 물체의 각 선분에 대하여 그림 (2.1) 과 같은 관계를 갖는다

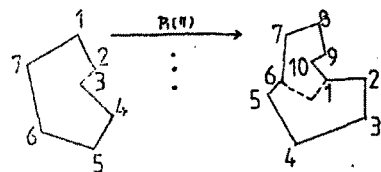


그림 (2.1) 선분 사이의 인식확률

$$P1 = [P1(1), P1(2), \dots, P1(M)] \quad (2.3)$$

의 확률벡터가 생긴다. 이 때

$$\sum_{m=1}^M P1(m) = 1, P1(m) \geq 0$$

의 제한 조건이 따른다. (2.3)과 그에 따른 제한 조건을 만족하는 확률 벡터가 비교물체의 N개의 선분에 대하여 생긴다. 초기 확률 계산은 (2.4) (2.5) 을 거쳐 생긴다 [1]. 여기서 M은 비교물체의 i번째 선분과 비교되는 물체의 k번째 선분의 각 특징 f를 가중치를 곱하여 더한 것이다. (2.4)에서 p=1일 때 내각의 비교를 하고 p=2일 때 외각 비교를 하였다. 여기서 외각은 다음과 같이 보일 수 있다 [1].

$$M(T_i, O_k) = \sum_{p=1}^p f_{ip} - f_{op} \cdot W_p \quad (2.4)$$

$$P(k) \propto 1 / (1 + M(T_i, O_k)) \quad (2.5)$$

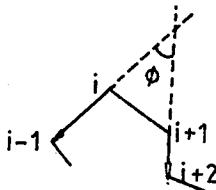


그림 (2.2) 의 각

2. 양립 함수 (compatibility function) 와 양립 벡터 (compatibility vector).

비교물체의 한 성분과 그 주위 선분 2개와 비교되는 물체의 한 선분과 역시 그 주위 선분 2개의 기하학적 복성이 어느 정도 유사한가를 나타내는 함수를 양립함수 C2라 하면, C2는 다음과 같다.

$$C2: S3 \times O3 \rightarrow [0, 1] \quad (2.6)$$

S3, O3는 다음과 같다.

$$S3 = \{(T_{i-1}, T_i, T_{i+1})\} \quad (2.7)$$

$$O3 = \{(O_{j-1}, O_j, O_{j+1})\} \quad (2.8)$$

S3의 순서쌍의 갯수는 N개이며 O3의 순서쌍의 갯수는 M개이다. C2의 계산을 위해 $T_{i-1} \leftrightarrow O_{j-1}$, $T_i \leftrightarrow O_j$, $T_{i+1} \leftrightarrow O_{j+1}$ 의 평균 회전도, 평균 길이비, 평균 내각을 계산하여 각각 변환과의 차를 구하여 더한다. 이 때 C2 (i, j, i-1, j-1, i+1, j+1)은

$$C2(i, j, i-1, j-1, i+1, j+1) = 1 / (1 + \text{총오차}) \quad (2.9)$$

와 같다 [1]. C2를 사용하여 비교물체의 i번째 선분이 비교되는 물체의 j번째 선분에 대하여 갖는 양립성 (compatibility)은 다음과 같은 양립벡터의 성분으로 보일 수 있다.

$$Q_i(k) = C2(i, j, i-1, j-1, i+1, j+1) * (P_{T_{i-1}}(k) + P_{T_{i+1}}(k)) \quad (2.10)$$

$$q_i(k) = Q_i(k) / \sum_{m=1}^M Q_i(k) \quad (2.11)$$

3. 전체 목적 함수
전체 목적 함수 J는 다음과 같다.

$$J = \sum_{i=1}^N \bar{P}_i \cdot \bar{Q}_i \quad (2.12)$$

(2.12)는 중심선분과 주위선분과의 불일치를 수식에서 잘 나타내고 기울기값을 구하기 쉬운 형태이다. J의 값이 최대가 될 때 확률수렴을 끝낸다.

III. 기울기 추정 알고리즘과 확률수렴

Pi(k)가 J의 표면에서 갖는 기울기는 다음과 같다.

$$\nabla_i(k) = \frac{\partial J}{\partial P_i(k)} = q_i(k) + \sum_j \frac{1}{D_j} * C2(j, 1, j+2, 1+2, i, k) * (P_i(1) - \bar{P}_i \cdot \bar{Q}_i) \quad (3.1)$$

이 때 $K = 1, \dots, M$ 이고 $j = i-1, i+1$, $D_j = \frac{\partial J}{\partial P_j(m)}$ 이다. n번째 기울기와 (n-1)번째 기울기의 합을 $\bar{V}_i^{(n)}$ 라 하면

$$\bar{V}_i^{(n)}(k) = \bar{V}_i^{(n-1)}(k) + \bar{V}_i^{(n)}(k) \quad (3.2)$$

$$\bar{V}_i^{(n)}(k) = \max_k \bar{V}_i^{(n-1)}(k) \quad (3.3)$$

이 된다. cfshoulder를 10으로 고정하면

$$\bar{P}_i^{(n)}(k) = \tan^{-1}(\text{cfshoulder} * \bar{V}_i^{(n)}(k) / \bar{V}_i^{(n-1)}(k)) \quad (3.4)$$

$$\bar{P}_i^{(n)}(k) = \begin{cases} \beta_i^{(n)}(k) & \beta_i^{(n)}(k) > 0 \\ -\beta_i^{(n)}(k) & \beta_i^{(n)}(k) \leq 0 \end{cases} \quad (3.5)$$

와 같이 cacf가 얻어진다. cacf를 구하면 n번째 과정의 확률 수렴은 다음과 같다.

$$P_i^{(n)}(k) = P_i^{(n-1)}(k) + \bar{P}_i^{(n)}(k) - \bar{V}_i^{(n-1)}(k) \quad (3.6)$$

IV. 축벡터를 중심으로 한 선분인식

축벡터란 가장 먼저 인식된 두 쌍의 꼭지점이 만드는 벡터를 뜻하는 것으로 다음과 같이 보일 수 있다.

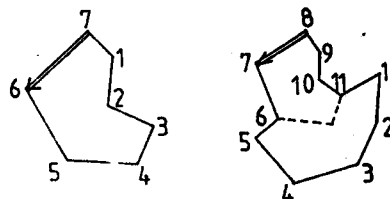


그림 (4.1) 물체의 축벡터

비교물체의 나머지 선분은 축벡터가 이 선분과 만드는 각의 비 내각의 크기와 비교되어 선분인식이 된다. Ripple은 다음과 같이 정의된다.

$ripple(i, i+1) = (s_i + s_{i+1})/s_i$
 물체의 ripple set은 다음과 같다. ripple set은

$$RS = \{k: ripple(i, k) \leq 1.3, k, i \in M\} \cup \{i\}$$

$$M = \{1, \dots, N\} \quad (4.1)$$

선분들의 방향이 유사할때 특히 효과적인 특성으로 쓰인다.

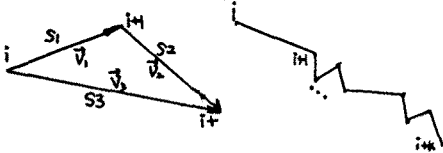


그림 (4.2) ripple의 정의 그림 (4.3) ripple set

비교되는 물체에 존재하는 모든 비교물체를 찾을 때는 속 비교물체를 구한 후 인식이 되지 않은 나머지 선분중 다시 축 벡터를 구하여 선분 인식을 한다.

V. 실험 및 결과

1. 한개의 비교물체를 비교되는 물체에서 검색 (searching)

단순한 다각형인 비교 물체를 찾는 실험을 다음과 같이 하였다. 표 3-2에서 Bir Bhanu의 확률 수렴 결과는 비교물체의 선분 4와 선분 5가 잘못인식이 되었다. 다음 그림에서 비교되는 물체의 선분 8은 영역분할과 다각형 근사화에서 생긴 잘못된 꼭지점이다. 꼭지점의 번호 11은 비교 물체의 선분이 비교되는 물체에서 선분 인식이 되지 못하고 공집합 (nil class)에 배정이 되었음을 나타낸다. 수행 회수 2에 의해 비교물체의 1과 6이 축벡터의 양쪽점으로 나타났다.

| | 제한된 알고리즘 | Bir Bhanu |
|---|----------|-----------|
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 11 | 11 |
| 4 | 11 | 9 |
| 5 | 9 | 10 |
| 6 | 10 | 10 |

표(5.1) 실험 결과

비교 물체를 비교하는 물체에서 찾는 다른 실험의 예를 다음에 든다. 여기서 비교 물체는 자동차 부품을 다각형 근사화한 것이다. 축벡터는 비교 물체에서 19에서 1로 향한것과 비교되는 물체에서 9에서 10으로 향한 두 개가 생긴다.

Bir Bhanu의 경우 비교물체의 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 17, 18, 19가 잘못 인식이 되었다. 제한된 알고리즘의 경우 10이 19로 인식이 안된것은 비교물체의 9에서 11까지의 거리가 비교되는 물체의 19에서 20까지의 거리보다 상대적으로 긴 까닭이다.

축벡터를 사용한 경우의 결과는 Bir Bhanu의 경우에서보다 좋은 것으로 나타났다. Bir Bhanu의 경우와 같이 기울기 추정에 의해 모든 선분을 인식할 경우는 양립함수와 양립벡터가 더 많은 선분의 인권을 가져야만 가능하나 계산시간의 증가가 일어난다.

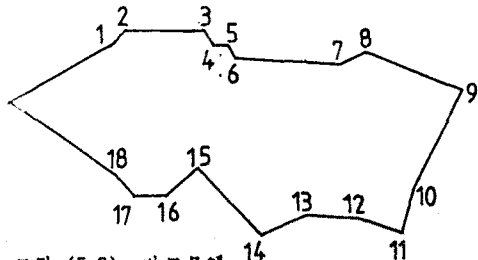


그림 (5.3) 비교물체

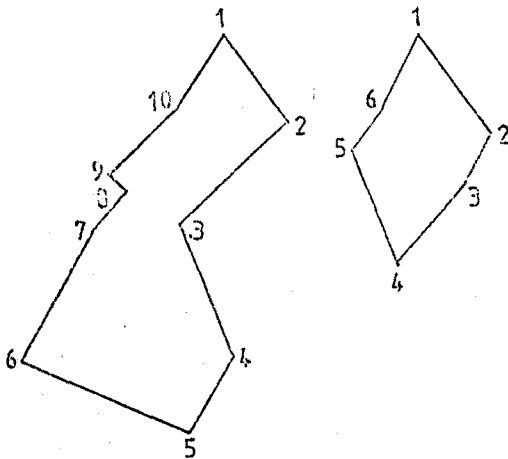


그림 (5.2) 비교되는 물체

그림 (5.1) 비교물체

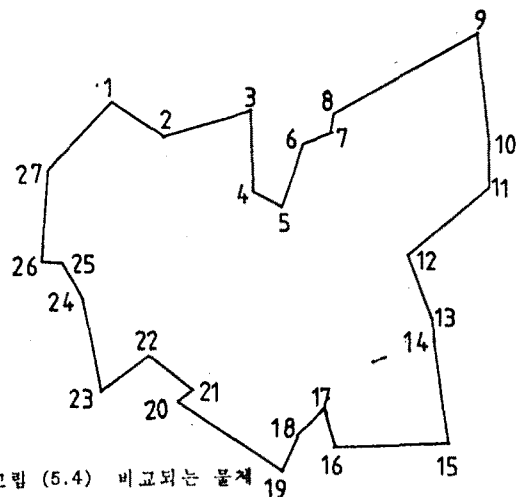


그림 (5.4) 비교되는 물체

표 (5.2) 실험 결과
(A) 제안된 알고리즘 결과
(B) Bir Bhanu의 실험결과
(29에 배정된 선분은 공집합에 들어감)

| 비교물체 | 비교되는 물체 (A) | 비교되는 물체 (B) |
|------|-------------|-------------|
| 1 | 10 | 29 |
| 2 | 10 | 29 |
| 3 | 11 | 11 |
| 4 | 11 | 7 |
| 5 | 11 | 29 |
| 6 | 11 | 29 |
| 7 | 29 | 7 |
| 8 | 17 | 29 |
| 9 | 19 | 29 |
| 10 | 29 | 15 |
| 11 | 29 | 29 |
| 12 | 29 | 19 |
| 13 | 29 | 29 |
| 14 | 29 | 29 |
| 15 | 29 | 5 |
| 16 | 29 | 6 |
| 17 | 29 | 11 |
| 18 | 29 | 7 |
| 19 | 9 | 9 |

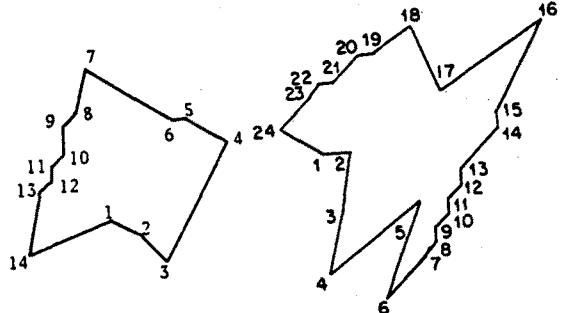


그림 (5.6) 비교물체 (B) 그림 (5.7) 비교되는 물체

표 (5.3) 실험 결과
(a) 제안된 알고리즘 결과
(b) Bir Bhanu의 실험결과
(25에 배정된 선분은 공집합에 들어감)

| 비교되는 물체 | (a) 비교 물체 (A) | (a) 비교 물체 (B) | (b) 비교 물체 (A) | (b) 비교 물체 (B) |
|---------|---------------|---------------|---------------|---------------|
| 1 | 21 | 25 | 5 | 25 |
| 2 | 22 | 2 | 6 | 23 |
| 3 | 25 | 3 | 7 | 24 |
| 4 | 25 | 3 | 8 | 18 |
| 5 | 25 | 4 | 25 | 25 |
| 6 | 25 | 5 | 25 | 25 |
| 7 | 25 | 6 | 25 | 25 |
| 8 | 9 | 7 | 25 | 13 |
| 9 | 9 | 8 | 25 | 25 |
| 10 | 9 | 25 | 25 | 25 |
| 11 | 10 | 25 | 25 | 25 |
| 12 | 11 | 25 | 25 | 19 |
| 13 | 11 | 25 | 25 | 14 |
| 14 | 12 | 25 | 25 | 18 |
| 15 | 12 | | 25 | |
| 16 | 12 | | 14 | |
| 17 | 12 | | 25 | |
| 18 | 13 | | 12 | |
| 19 | 13 | | 25 | |
| 20 | 14 | | 25 | |
| 21 | 15 | | 25 | |
| 22 | 16 | | 25 | |
| 23 | 17 | | 11 | |
| 24 | 18 | | 25 | |
| 25 | 19 | | 25 | |
| 26 | 25 | | 4 | |

2. 비교되는 물체를 이루는 모든 비교물체 검색

비교되는 물체에 있는 모든 비교물체는 순차적으로 형상인식을 하게 된다. 비교물체 A의 26, 1은 비교물체의 축 벡터를 형성한다. A를 비교되는 물체에서 형상인식을 한 후 비교되는 물체의 다른 꼭지점들에 대해서 형상인식을 하면 비교물체 B의 8, 9가 두 번째 축 벡터를 형성한다. 만약 비교되는 물체의 같은 선분에 두 비교물체의 일정한 점이 동시에 배정되면 비교물체의 해당 선분과의 끼인 각의 차이와 꼭지점의 시계 반대방향쪽의 벡터와의 차이가 작을수록 선분 인식을 한다. Bir Bhanu의 실험결과를 보면 비교물체 (A)의 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 23이 잘못된 선분 인식이다. 제안된 알고리즘의 경우 비교물체 (A)의 11은 10에서 11로의 벡터가 비교물체의 축 벡터와 이루는 각의 허용한도를 크게 함으로써 가능하다. 비교물체 (B)의 8과 9는 축 벡터가 되고 비교하는 물체의 7과 8이 역시 축 벡터가 된다. 비교물체 (B)의 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14는 잘못된 선분 인식이다.

Bir Bhanu의 결과에서 25는 비교물체의 해당 선분이 비교되는 물체에 없는 것을 가리킨다. 비교되는 물체의 선분 25는 공집합에 속한다.

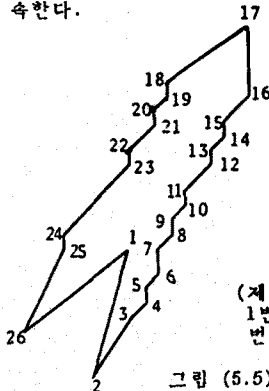


그림 (5.5) 비교물체 (A)

(제안된 알고리즘에서는 1번부터 시계 방향으로 번호가 증가함)

VI. 결론
이상에서 몇 개의 실험을 통하여 제안된 알고리즘의 성능을 살펴 보았다. 제안된 알고리즘은 몇 가지의 특성을 갖는데, 우선 계산량이 다른 확률수렴 알고리즘에 비해 매우 작다. [4], [5]의 경우에 비해 곱셈의 개수는 최소 $1: (2/N)$ (N 은 비교물체의 선분개수)의 비로 된다. 알고리즘에서 전체 목적함수의 값은 축 벡터에 의한 선분 인식으로 실질적인 최대값을 갖게 된다. 영역분할과 다각형 근사화에서 생기는 에러를 보상하는 선분 특성인 ripple은 선분 인식시 효과적인 특성이다. 결국 영역분할과 다각형 근사화에서 큰 오차가 생기지 않는 한 완벽한 형상인식을 기할 수 있다. 축 벡터에 의한 형상인식은 3차원 물체에도 효과적으로 쓰일 것으로 보이며, cacf를 사용한 영역분할 알고리즘은 현재 진행 중이다. cacf가 정확한 기울기를 갖기 위해 취해야 할 값의 범위, 값의 적응 등은 더 많은 연구가 이루어져야 할 것이다.

참고 문헌

[1] B. Bhanu, O.D. Faugeras, "Shape Matching of Two Dimensional Objects", IEEE Trans. PAMI, vol. 6, pp.137-156, March 1984.