객체 모양의 특징을 표현하는 재귀적 윤곽 우세 점 추출 방안

김 영 태*, 엄 기 현
동국대학교 대학원 컴퓨터공학과

Recursive extraction method for representing shape feature of object

Youngtae Kim*, Kyhyun Um
Dept. of Computer Engineering, Graduate School, Dongguk University
{yyoungs, khum}@dgu.ac.kr

요약

본 논문은 객체의 유사성 비교를 위해 객체의 모양을 표현하는 한 가지 특징인 윤곽선상의 우세 점들을 찾는 재귀적 윤곽선 근사 알고리즘을 제안한다. 이 알고리즘은 같은 모양의 객체에 대하여 그 객체의 무게 중심을 이용하여 항상 일정한 특정 시작점을 찾음으로써 동일한 우세 점들을 재귀적으로 빠른 수행 시간에 찾는다. 또한 이 알고리즘은 열린 곡선, 닫힌 곡선 및 다각형 등 어떤 모양의 평면 도형에도 모두 적용 가능하다. 제안 알고리즘의 평균 시간 복잡도는 $O(nlogn)$이다.

1. 서론

객체의 모양은 이미지 데이타베이스 응용분야에서 유사성 검색을 위해 매우 중요하다. 객체의 모양 표현은 인덱싱, 검색, 유사성 비교시 사용되기도 하며 객체의 모양 질의를 통하여 유사한 객체 집합을 찾기 위해 이미지 데이타베이스에서 빠른 검색을 수행하는데 이용될 수 있다.

객체의 모양을 이용한 유사성 검색은 고정된 차수를 가진 객체의 모양을 표현하는 특징 벡터들을 비교함으로써 이루어진다. 두 객체의 모양이 유사한지를 식별하기 위해 윤곽선상의 모든 점에 대하여 점 대 점 매핑을 한다면 많은 시간이 소비되며 매우 비효율적이다[3, 5]. 윤곽선상의 모든 점을 비교하지 않고 객체의 모양을 변형시키지 않으면서 객체의 모양에 대한 특징을 표현하는 우세 점들을 찾는다면 보다 빠른 유사성 검색이 가능하다. 또한 크기가 다르거나 회전된 객체들간에 모양이 유사한지를 윤곽선 정보만을 이용하여 검색하는 것은 매우 비효율적이다. 이 경우에 두 객체의 모양 비교를 할 때 시작점이 달라짐으로써 객체의 모양이 다르게 표현되므로 윤곽선상의 특정 시작점을 찾아야 정확한 유사성 검색이 가능하다[5, 6].

본 논문에서는 같은 모양의 객체에 대하여 그 객체의 무게 중심을 이용하여 항상 일정한 특정 시작점을 찾음으로써 동일한 우세 점들을 빠른 수행 시간에 찾는 재귀적 윤곽선 근사 알고리즘을 제안한다. 이 알고리즘은 복잡한 모양의 열린 곡선, 닫힌 곡선 및 다각형에 모두 적용 가능하며 열린 곡선은 닫힌 곡선으로 간주한다. 객체를 세그먼트할 때 객체의 모양이 미세하게 변한다면 객체의 모양 비교시 문제점을 야기할 수 있으므로 이미지 처리에서 잡음 제거와 같은 전처리 과정이 필요하다[1].

본 논문의 구성은 다음과 같다. 2장에서는 경계 기반 방법에서 객체의 모양을 표현하는 방법을 제시한 관련 연구를 언급한다. 3장에서는 객체 모양의 특징 추출을 위한 재귀적 윤곽선 근사 알고리즘을 제안한다. 4장에서는 제안 알고리즘을 이용한 실험과 분석 결과에 대해 살펴본다. 5장에서 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

객체의 모양을 표현하는 연구 분야는 매우 다양하며 객체의 모양을 표현하는 방법은 경계 기반 방법과 영역 기반 방법으로 분류된다[2, 3]. 경계 기반 방법은 오직 객체 모양의 윤곽선만을 이용하고 객체 내부에 대해서는 고려를 하지 않는다. 영역 기반 방법은 객체 내부 정보와 윤곽선을 함께 고려한다. 본 논문에서는 경계 기반 방법을 이용하며 객체 모양의 특징은 직경, 면적, 중심, 전체 모양을 나타내는 사각형 또는 삼각형[4] 등과 같은 전체적인 모양으로부터 추출되는 전역적인 특징과 다각형 근사[1, 5, 7], 곡률[1], 모멘트[2, 3], 체인 코드[5] 등과 같은 모양의 지역적인 영역으로부터 추출되는 지역적인 특징이 있다.

객체 모양의 전역적인 특징인 삼각 분할을 이용한 유사성 검색 방법을 제안하였다[4]. 이 알고리즘은 객체의 모양이 다각형일 때 평균 시간 복잡도는 $O(nlogn)$이지만 복잡한 곡선일 경우 $O(n^2)$이다.

다각형 근사는 객체 모양의 유사성 비교시 윤곽선상의 점들을 줄임으로써 얻은 우세 점들을 이용한 객체의 모양을 표현하는 방법이다. 곡률 크기 공간에서 윤곽선상의 근접한 볼록한 부분(convex)과 오목한 부분(concave)의 세그먼트를

본 연구는 1999년도 한국과학재단 특정기초연구(과제 번호: 1999-1-303-002-3) 내용의 일부임

반복적으로 합병을 수행하는 알고리즘을 제안하였다[1]. 이 알고리즘은 세그먼트의 합병 단계에서 시간 복잡도가 $O(n^2)$이다. 윤곽선상의 모든 점에서 최대 유클리디언 거리를 지닌 두 개의 점을 찾는 단계를 포함하는 우세 점 추출 알고리즘을 제안하였다[5]. 이 알고리즘은 윤곽선 상에서 최대 거리를 가진 두 개의 점을 찾을 때의 시간 복잡도는 $O(n^2)$이다. 그래프 기반의 우세 점 추출 알고리즘을 제안하였다[7]. 이 알고리즘은 같은 모양의 객체에 대하여 특정 시작점을 찾을 수 없으며 윤곽선 상의 모든 점에서 우세 점들을 찾는 시간 복잡도는 $O(n^2 \log n)$이다. 최소 경계 원을 이용한 방법으로 최소 경계 원의 특징을 나타내는 각도 시퀀스와 접촉점들의 개수를 찾는 방법을 제안하였다[2, 6]. 이 방법은 의료 이미지와 같은 제한된 도메인에 적용 가능하다.

기존의 연구에서는 삼각 분할, 다각형 근사, 최소 경계 원 등을 이용하여 객체 모양의 특징을 표현하였으며 삼각 분할과 다각형 근사 방법은 $O(n^2 \log n)$에서 $O(n^2)$이며 최소 경계 원을 이용한 방법은 적용 범위가 제한적이다. 본 논문에서 제안한 알고리즘은 객체의 모양을 표현하는 특징인 윤곽선상의 우세 점들을 찾기 위해 객체의 무게 중심을 이용하며 같은 모양을 가진 객체에 대하여 항상 일정한 시작점을 찾음으로써 우세 점들을 빠르게 찾는다.

3. 모양의 특징 추출

윤곽선상의 모든 점을 객체 모양의 식별에 이용한다면 시간이 많이 소모되므로 객체 모양의 특징인 우세 점들을 추출하기 위해 재귀적 윤곽선 근사 알고리즘을 제안한다.

3.1 재귀적 윤곽선 근사 알고리즘의 기본 개념

객체 모양을 표현하는 윤곽선상의 우세 점들을 재귀적 방법을 이용하여 찾는 알고리즘의 동작 과정은 다음과 같다 (그림 1).

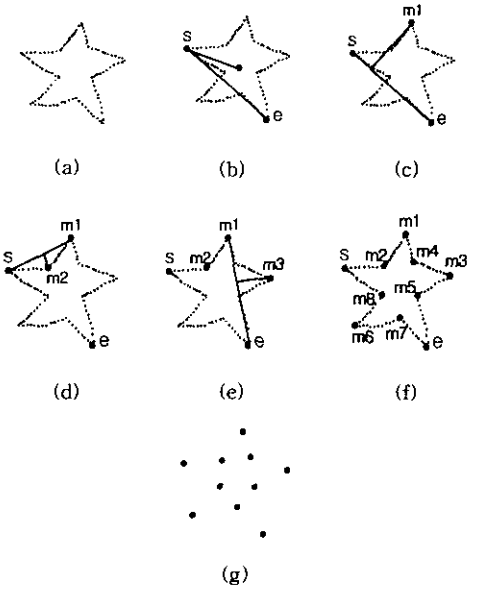

그림 1. 제안 알고리즘의 단계별 예

객체의 모양을 표현하는 윤곽선상의 모든 점을 입력으로 받는다(a). 객체의 무게 중심을 구하고 무게 중심에서 최대 거리인 시작점 s를 찾은 후에 s를 기준으로 모든 점을 시계 방향으로 정렬하고 s로부터 최대 거리를 가진 끝점 e를 찾는다(b). s와 e를 이은 선분으로부터 s에서 e까지의 점들을 시계방향으로 비교하면서 최대 거리를 가진 중간점 m1을 찾는다(c). 시작점과 끝점은 이미 결정되었으므로 비교에서 제외된다. 이 때 최대 거리가 임계값보다 크다면 중간점 m1이 결정된다. m1이 결정되면 s와 m1을 이은 선분으로부터 s와 m1까지의 점들을 비교하면서 중간점 m2를 찾는다(d). 계속적으로 m2를 기준으로 윤곽선을 좌우로 두 부분으로 나누어 수행하게 된다. 만약 임계값보다 작다면 중간점을 찾는 재귀 함수는 종료한다. m1에서 e까지의 점들을 비교하면서 중간점 m3을 찾는다(e). 동일한 방법으로 점 m4, 점 m5를 찾을 수 있다. 지금까지 윤곽선상의 시작점 s와 끝점 e사이의 점들을 비교하여 우세 점들을 찾는 과정을 살펴보았다. 또한 시작점 e와 끝점 s사이의 점들도 동일한 방법으로 적용한다면 모든 점에 대하여 객체의 모양을 표현하는 우세 점들을 모두 찾을 수 있다(f). 최종적으로 객체의 모양은 10개의 우세 점들로 표현된다(g).

3.2 재귀적 윤곽선 근사 알고리즘

제안한 알고리즘은 초기화를 포함하여 8단계로 구성되어 있으며 단계별 수행 절차는 다음과 같다. 이 알고리즘에서 찾은 시작점, 끝점 및 여러 개의 중간점들이 윤곽선상의 우세 점들의 집합이 된다. 거리는 유클리디언 거리이다.

Algorithm InterestPoints()

/* n: 윤곽선상의 모든 점의 총 개수
w: 윤곽선상의 우세 점들의 총 개수
$P = \{p_1, p_2, \ldots, p_n\}$: 윤곽선상의 모든 점에 대한 $p_1, p_2, \ldots, p_n$ 의 집합
$U = \{p_{i_1}, p_{i_2}, \ldots, p_{i_w}\}$: 윤곽선상의 우세 점들의 집합
$I = \{i_1, i_2, \ldots, i_w\}$: 윤곽선상의 우세 점들을 가리키는 인덱스 집합
$D(a,b)$: 점 a와 점 b사이의 유클리디언 거리
$D(\overline{ab},c)$: 선분 $\overline{ab}$와 점 c사이의 유클리디언 거리 */

초기화. $U=I=\varnothing$
단계 1. 객체의 무게 중심 p_0를 찾는다.
단계 2. 최대 $D(p_0, p_i)(1 \le i \le n)$인 시작점 $p_{s \cdot i}$를 찾는다.
단계 3. p_s을 기준으로 윤곽선 상의 모든 점 p_i $(1 \le i < s-1, s < i \le n)$를 시계 방향으로 재정렬한다 $(p_{s-1}, p_{n+1} = p_s)$.
단계 4. 최대 $D(p_s, p_i)$ $(s < i \le n)$인 끝점 $p_{e \cdot i}$를 찾는다.
단계 5. s,e를 I 에 삽입한다. 즉, $I = I \cup \{s,e\}$
단계 6. 시작점 위치 s로부터 끝점 위치 e와 시작점 위치 e로부터 끝점 위치 $n+1$ 에 대하여 *contour* 재귀 함수를 차례로 호출한다. 즉, *contour*(s,e), *contour*($e,n+1$)
단계 7. I 내의 인덱스들을 오름차순으로 정렬한 후 w개의 우세 점들을 얻는다. 즉, $U=\{p_{i_1}, p_{i_2}, \ldots, p_{i_w}\}$

Function contour (s , e)

/* s: 시작점 위치, e: 끝점 위치 */

단계 6a. 최대 $D(\overline{p_s p_e}, p_i)$ $(s<i<e)$인 중간점 p_{m_i}를 찾는다.

단계 6b. 만약 $D(\overline{p_s p_e}, p_m)>E$ 이라면 중간점 위치 m을 I 에 삽입하고 시작점 위치(s)로부터 끝점 위치(m)와 시작점 위치(m)로부터 끝점 위치(e)에 대하여 *contour* 재귀 함수를 차례로 호출한다. 즉, $I=I\cup\{m\}$, *contour*(s,m), *contour*(m,e)

단계 6c. 만약 $D(\overline{p_s p_e}, p_m)\le E$ 이라면 재귀 함수는 종료한다.

단계 1은 윤곽선상의 모든 점을 이용하여 객체의 무게 중심을 구한다. 단계 2는 객체의 무게 중심으로부터 최대 거리를 갖는 윤곽선상의 시작점을 찾는다. 단계 3은 시작점을 기준으로 윤곽선상의 모든 점을 시계 방향으로 재정렬한 후 윤곽선상의 점들의 집합 n+1번째에 시작점을 저장한다. 단계 4는 시작점으로부터 최대 거리를 가진 윤곽선상의 끝점을 찾는다. 단계 5는 시작점과 끝점의 위치를 인덱스 집합에 삽입한다. 단계 6은 시작점과 끝점을 기준으로 윤곽선을 두 부분으로 나누고 중간점들을 찾기 위해 *contour* 함수를 차례로 호출한다. *contour* 함수에서 단계 6a는 시작점과 끝점을 이은 선분으로부터 시작점과 끝점 사이에 있는 윤곽선상의 점들에 대하여 최대 거리를 가진 중간점을 찾는다. 단계 6b는 단계 6a에서 구한 최대 거리가 임계값 보다 크다면 중간점 위치를 인덱스 집합에 삽입하고 시작점과 끝점사이에 있는 점들을 중간점을 기준으로 두 부분으로 나누어 *contour* 함수를 차례로 호출한다. 만약 임계값보다 작거나 같다면 *contour* 함수는 종료된다.

임계값 E를 구하는 공식은 다음과 같다.

$$E=L/\delta$$

δ는 임계 계수, L은 객체의 무게 중심과 시작점간의 거리이다. δ의 값에 따라 윤곽선상의 우세 점들의 개수가 달라진다. δ의 값이 매우 커짐에 따라 임계값 E는 0에 가까워지며 우세 점들의 총 개수는 윤곽선상의 점들의 총 개수와 동일하다. 제안 알고리즘에서는 객체 모양을 적절하게 표현하기 위해 δ의 값을 10으로 하였다.

단계 7은 중간점들을 찾는 *contour* 재귀 함수가 종료되면 윤곽선상의 우세 점들을 가리키는 인덱스 집합을 오름차순(시계 방향)으로 정렬한 후에 최종적으로 우세 점들의 집합을 얻는다.

객체의 유사성 비교를 위해 객체의 모양을 표현하는 우세 점들을 찾는 기준은 시작점이다. 시작점은 객체의 무게 중심으로부터 최대 거리를 가지며 그 위치는 항상 일정하다. 그러므로 이 알고리즘은 객체 모양의 특징 추출을 위해 항상 특정한 시작점을 찾는다.

4. 실험과 분석

객체 모양을 표현하는 우세 점들을 찾기 위해 원, 타원, 대칭성을 가진 도형, 복잡한 도형 등의 닫힌 곡선, 열린 곡선, 대칭성을 가진 도형 및 다각형 등의 여러 가지 모양을 가진 객체에 대하여 실험 및 분석을 하였다. 본 논문에서는 제안한 알고리즘을 적용하여 그림 2에서와 같이 이미지 객체(모양 a)의 대칭(모양 b), 축소(모양 c), 확대(모양 d), 회전(모양 e)시 우세 점들을 찾기 위해 다섯 가지의 모양을 이용하여 실험 및 분석을 하였다.

본 논문에서 제안한 알고리즘을 실험하기 위한 시스템을 윈도우 NT 4.0과 JDK 1.3을 사용하여 구현하였다.

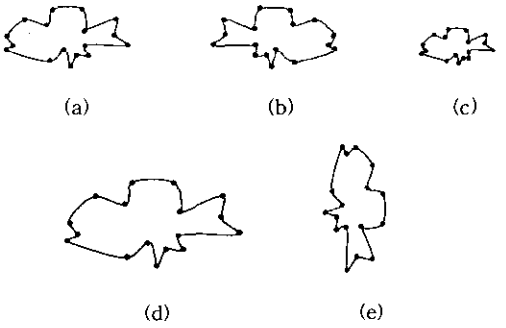

그림 2. 일반적인 객체의 모양

그림 2의 모양 a, 모양 b, 모양 c, 모양 d, 모양 e 모두 시작점은 같은 위치에서 찾아지며 우세 점들의 개수도 17개로 동일하였고 모양 b를 제외한 다른 모양들은 우세 점들의 순서가 같았다. 모양 b는 다른 모양들과 시작점은 같은 위치에서 추출되지만 우세 점들의 순서는 시작점을 기준으로 시계방향으로 정렬됨으로써 다른 모양들과 반대로 나타났다.

기존의 연구에서 객체의 모양, 특히 나선형 모양에 대하여 우세 점들을 찾는 알고리즘의 시간 복잡도는 대부분 $O(n^3)$에서 $O(n^2)$이다[4, 5, 7]. 본 논문에서 제안한 재귀적 윤곽선 근사 알고리즘의 평균 시간 복잡도는 다음과 같다. n은 객체 모양을 표현하는 윤곽선상의 점들의 총 개수라고 할 때 3.2 알고리즘의 단계 1, 단계 2, 단계 3, 단계 4의 시간 복잡도는 제각기 $O(n)$이며 단계 6a와 단계 6b의 평균 시간 복잡도는 $O(2\cdot nlogn)$이다. 따라서 제안 알고리즘의 전체 평균 시간 복잡도는 $O(nlogn)$이다.

5. 결론

본 논문에서는 객체 모양의 특징 추출을 위해 재귀적 윤곽선 근사 알고리즘을 제안하였다. 제안 알고리즘은 객체의 무게 중심을 이용하여 같은 모양의 객체에 대하여 항상 특정 시작점을 찾음으로써 기존의 방법보다 빠른 수행시간에 윤곽선상의 거의 같은 우세 점들을 찾는다. 그러나 무게 중심으로부터 동일 거리를 가진 점들이 나타날 경우 정규화를 적용하는 것이 필요하다.

참고논문

[1] E. Milios, E. Petrakis, "Shape Retrieval Based on Dynamic Programming", IEEE Transactions on Image Processing, Vol. 9, No.1, pp.141-147, 2000.

[2] M. Safar, C. Shahabi, X. Sun, "IMAGE RETRIEVAL BY SHAPE: A COMPARATIVE STUDY", IEEE Multimedia and Expo, Vol.1, pp.141-144, 2000.

[3] R. Baldock, J. Graham, "Image Processing and Analysis", Oxford University Press 2000.

[4] Kikuo Fujimura, Yusaku Sako, "Shape Signature by Deformation", IEEE Shape Modeling and Applications, pp.225-232, 1999.

[5] J. Wang, W. Chang, and R. Acharya, "Efficient and Effective Similar Shape Retrieval", IEEE Multimedia Computing and Systems, pp.875-879, 1999.

[6] Maytham Safar, Cyrus Shahabi, "Two Optimization Techniques to Improve the Performance of MBC-based Shape Retrieval", Multimedia Information System 2000.

[7] H. Imai, M. Iri, "Computational-Geometric Methods for Polygonal Approximation of a Curve", Computer Vision, Graphics and Image Processing, pp.31-41, 1986.