

# 모양 기반 이미지 분류를 위한 최적의 우세점 추출

## (Extraction of Optimal Interest Points for Shape-based Image Classification)

조 성 태<sup>†</sup>    엄 기 현<sup>\*\*</sup>  
(Seongtaek-Jo)    (Kyhyun-Um)

**요 약** 이 논문에서는 이미지 데이터베이스에서 모양 특징 기반 이미지 분류와 인덱싱을 위해 객체의 윤곽선 특성을 고려해 임계값을 동적으로 결정하여 최적 우수점을 추출하는 알고리즘을 제안한다. 동적 임계값 결정은 원본 모양의 윤곽선 길이 비와 근사화된 다각형의 둘레 길이 비를 알고리즘 수행시 점진적으로 검사하는 방법을 사용한다. 이 알고리즘은 윤곽선 특징을 반영하여 동적인 임계값 검사를 함으로써 의사점 수를 최대한 줄이며 최소 우수점만으로 모양 특징 정보를 추출할 수 있는 장점을 보인다. 제안한 방법은 객체의 윤곽선을 이루는  $n$ 개의 점에서  $m$ 개의 최적 우수점을 찾는데 평균  $O(n \log n)$ 이 걸린다. 최적화 평가는 7가지 서로 다른 특성을 가지는 70개의 합성 모양과 1,100개의 어류 모양에 대해 알고리즘을 적용하고 그 결과에 대해 평가 함수를 구성하여 수행하였다. 최적화율은 실험 모양들에 대해 평균 0.92를 보였으며 기존 알고리즘에 대해 약 14% 최적화 성능 개선을 보였다. 제안한 알고리즘을 통해 추출한 모양 특징 정보는 정규화를 통해 이미지 분류와 인덱싱, 유사도 검색에 활용할 수 있다.

**키워드** : 우수점, 다각형 근사화, 모양 기반 이미지 분류

**Abstract** In this paper, we propose an optimal interest point extraction method to support shape-based image classification and indexing for image database by applying a dynamic threshold that reflects the characteristics of the shape contour. The threshold is determined dynamically by comparing the contour length ratio of the original shape and the approximated polygon while the algorithm is running. Because our algorithm considers the characteristics of the shape contour, it can minimize the number of interest points. For  $n$  points of the contour, the proposed algorithm has  $O(n \log n)$  computational cost on an average to extract the number of  $m$  optimal interest points. Experiments were performed on the 70 synthetic shapes of 7 different contour types and 1,100 fish shapes. It shows the average optimization ratio up to 0.92 and has 14% improvement, compared to the fixed threshold method. The shape features extracted from our proposed method can be used for shape-based image classification, indexing, and similarity search via normalization.

**Key words** : interest point, polygonal approximation, shape-based image classification

### 1. 서 론

멀티미디어 정보로서의 이미지는 색, 질감, 모양 등 다양한 시각적 정보를 포함하고 있다. 이러한 이미지의 기본 특징 정보를 사용하는 내용 기반 이미지 검색과

분류에서는 비교적 정보 추출과 구성이 쉬운 색, 질감을 주로 활용하고 있다. 모양 특징 정보는 다른 특징 정보에 비해 특징 추출이 쉽지 않다. 모양 특징을 사용하는 관련 연구들을 살펴보면 검색과 분류 성능 또한 다른 특징 정보를 사용하는 경우 보다 낮게 나타남을 알 수 있다. 이러한 현상은 동일 객체 모양에 대한 회전, 확대/축소 등 기하학적 변형에 따라 일관된 특징 정보를 추출하거나 정량화된 값을 구성하기 어렵기 때문에 발생한다. 하지만 이미지에 표현된 객체의 모양은 인간이 시

<sup>†</sup> 정 회 원 : 경민대학 인터넷정보과 교수  
stjo@kyungmin.ac.kr

<sup>\*\*</sup> 종신회원 : 동국대학교 컴퓨터멀티미디어공학과 교수  
khum@dgu.ac.kr  
논문접수 : 2003년 1월 23일  
심사완료 : 2003년 4월 7일

각적으로 이미지를 분류하거나 인식하는데 있어서 가장 중요한 정보이다. 따라서 이미지 데이터베이스에 대한 내용 기반 이미지 검색과 분류 성능을 높이기 위해서는 모양 특징 정보의 사용이 필요하다.

이를 위해 모양 특징 정보를 인덱스로 구성할 때 고차원 특성을 가지는 모양 특징 정보의 차원수를 최대한 줄이고 최적화된 특징 정보를 추출하는 것을 고려해야 한다. 인덱스 키를 구성하기 위한 전처리 과정으로서의 객체 모양에 대한 특징 정보 추출과 표현은 매우 다양하게 연구되었다. 이를 처리 대상 기준으로 분류하면 모양을 이루는 객체의 내부 정보와 윤곽선(contour)을 함께 고려하는 영역 기반(region-based) 방법과 객체의 윤곽선만을 고려하는 경계 기반(boundary-based) 방법으로 분류된다[1,9]. 영역 기반 방법은 모양 특징에 대한 자세한 정보를 표현할 수 있지만 경계 기반 방법보다 정보 추출 시간과 표현에 따른 비용이 많이 필요하다. 따라서 이 논문에서는 빠른 시간에 객체 모양의 특징 정보를 추출하고 차원수의 감소가 용이한 경계 기반 방법에 초점을 맞춘다.

경계 기반 방법에서는 주로 다각형 근사(polygonal approximation) 방법에 기초한 여러 연구 결과들이 제시되었다[2,5,7,8,10,11]. 다각형 근사는 객체 모양을 구성하는 윤곽선을 세그먼트(segment)로 분할하고 이를 기하학적 기본 요소(geometric primitives)들로 표현하는 방법이다[3]. 이 방법에서는 주어진 객체의 모양으로부터 초기 세그먼트의 시작점과 끝점을 결정하는 방법과 근사화한 세그먼트를 구성하는 양 끝점을 결과에 포함시킬지를 결정하는 임계값 선택이 중요하다. 시작점과 끝점의 결정이 잘못되는 경우는 의사점(spurious point)이 발생하여 부가적인 검증 작업이 필요하다. 임계값의 사용에 있어서는 고정된 임계값을 사용하는 방법[7,11]과 부가적인 곡률 추정을 통해 동적으로 임계값을 사용하는 방법[6]이 있다. 고정된 임계값을 사용하는 경우에는 최종 결과에 누락점(miss point) 또는 의사점이 발생할 수 있으므로 원본 객체의 윤곽선 정보를 훼손하거나 불필요한 정보를 포함하게 되어 추출한 정보의 차원수가 높아지는 문제가 생길 수 있다. 이러한 문제점들을 해결하기 위해서는 각 처리 대상 객체 모양의 윤곽선 특성을 반영한 일관된 시작점과 끝점의 선정 및 누락점 또는 의사점을 최소화할 수 있는 임계값 결정을 할 수 있는 모양 특징 추출 방법에 대한 연구가 필요하다.

이 논문에서는 이미지 검색과 분류에 모양 특징 정보를 이용하기 위해 재귀적 윤곽 우세점(interest point) 추출 방법[11]을 개선한 우세점 추출의 최적화 방법을

제시한다. 제안하는 방법은 객체의 모양 특징 정보를 훼손하지 않는 최소의 윤곽점들로 모양 특징 정보를 표현할 수 있는 동적 임계값 결정 방법을 사용하여 우세점에 대한 최적화를 수행한다. 제안한 알고리즘의 평균 시간 복잡도는  $n$ 개로 구성된 윤곽선 점들에 대해  $m$ 개의 우세점을 추출할 경우  $O(n \log n)$ 이다. 이 방법으로 추출한 모양 특징 정보는 우세점에 대한 정규화 방법[11]을 통해 객체 모양의 기하학적 변환이나 크기 변환에 무관하게 모양 특징 정보 표현이 가능하며 모양 기반 이미지 분류, 인덱싱과 검색에 사용할 수 있다.

논문의 구성은 다음과 같다. 2장에서는 경계 기반 방법을 사용하는 우세점 추출에 대한 기존 연구와 다각형 근사를 이용한 모양 정보 추출시 고려할 사항들에 대해 살펴본다. 3장에서는 우세점 추출을 최적화하기 위한 동적인 임계값 결정 방법과 이에 대한 알고리즘을 제시하고 성능을 분석한다. 4장에서는 제안한 알고리즘의 최적화에 대한 실험과 평가 내용을 제시하고 5장에서 결론을 맺는다.

## 2. 관련 연구

### 2.1 경계 기반 방법에서의 우세점 추출 연구

경계 기반 방법은 주어진 객체 모양을 구성하는 윤곽선의 전역적인 특징 또는 지역적인 특징을 사용한다. 전역적인 특징을 사용하는 모양 시그니처 방법은 삼각형을 기본 도형으로 하여 삼각 분할을 이용해 객체의 모양을 미리 정의한 모양으로 변환한다. 이 알고리즘은 객체의 모양이 다각형을 이룰 때 평균 시간 복잡도는  $O(n \log n)$ 이지만 복잡한 곡선을 이루는 모양에 대해서는  $O(n^2)$ 이다[4].

모양의 지역적인 특징을 이용하는 방법은 주로 순환적 또는 재귀적 다각형 근사를 통해 윤곽선에 포함된 점의 수를 줄임으로써 우세점만으로 객체의 모양을 표현하는 방법이다[3]. 이 방법에는 직선 세그먼트를 사용하는 방법[7], 동적 프로그래밍을 사용한 방법[6], 체인 코드를 사용한 다단계 다각형 근사 방법[10], 그래프 기반 방법[5], 재귀적 윤곽 우세점 추출 방법[11] 등이 제시되었다.

직선 세그먼트를 사용하는 방법은 반복적으로 세그먼트에 대한 스택을 사용해 윤곽선을 하나의 고정된 임계값을 만족하는 직선들로 분할한다. 이 방법에서는 초기 세그먼트의 시작점과 끝점의 지정에 따라 불필요한 점들이 우세점 집합에 포함되는 단점이 있으며 알고리즘의 시간 복잡도는 평균  $O(n^2)$ 이다. 동적 프로그래밍 방법은 곡률 크기 공간에서 윤곽선상의 근접한 불룩한 부

분(convex)과 오목한 부분(concave)의 세그먼트를 반복적으로 합병한다. 이 알고리즘은 윤곽선 세그먼트의 합병 단계에서 시간 복잡도는  $O(n^2)$ 이다. 다단계 다각형 근사 방법에서는 객체의 윤곽선을 표현하기 위해 체인 코드를 이용하여 윤곽선상의 모든 점에서 최대 유클리드 거리를 지닌 두 개의 점을 찾는 단계를 포함하는 두 개의 알고리즘을 제안하였다. 이 방법에서는 우세점들의 개수를 제한하고 있어 복잡한 곡선을 가진 모양일 경우 정확한 객체의 모양을 표현할 수가 없다. 또한 객체의 크기가 변하거나 회전될 경우 항상 일정한 값을 유지하기 위해 추출된 우세점들에 대해 윤곽선상의 모든 점에 대한 평균 좌표인 평균 중심점에서 최대 거리를 가지는 시작점을 이용하여 정규화된 인접 변환(affine transformation)을 적용할 때 평균 시간 복잡도는  $O(n^2)$ 이다. 그래프 기반 방법에서는 윤곽선상의 모든 점에 대해 거리를 계산하고 그 거리가 오차 범위 기준 내에 있을 때 비교한 점을 삭제하는 방법을 사용한다. 이 방법은 같은 모양의 객체에 대하여 특정 시작점을 찾을 수 없는 단점이 있다.  $n$ 개로 표현된 객체 모양에 대한 윤곽선상의 모든 점에서 우세점들을 찾는 시간 복잡도는  $O(n^2 \log n)$ 이다. 재귀적 윤곽 우세점 추출 방법에서는 객체 모양의 무게중심(centroid)으로부터 최대 유클리드 거리를 가지는 점을 시작점으로 선정하고 시작점에서 가장 먼 점을 끝점으로 지정한 후 고정 임계값을 사용해 재귀적으로 우세점을 추출한다. 이 방법은 평균 시간 복잡도가  $O(n \log n)$ 으로 기존 알고리즘들에 비해 좋은 성능을 보이지만 고정 임계값을 사용으로 객체의 모양 특징을 정확하게 반영하는 우세점 추출이 어렵다.

2.2 다각형 근사 알고리즘의 고려 사항

다각형 근사 알고리즘을 수행하기 위해서는 일반적으로 세그먼트의 초기 시작점과 끝점을 선정하는 과정과 목표점을 우세점에 포함할 지를 결정하기 위한 임계값 검사 과정이 필요하다. 초기 세그먼트에 대한 시작점과 끝점 지정은 첫째, 지역적으로 최대 곡률(local maxima curvature)을 가지는 두 점을 선택하는 방법, 둘째, 좌측 상단 점과 우측 하단 점을 선택하는 방법, 셋째, 임의의 두 점을 선택하는 방법 등이 있다. 첫째 방법은 지역적인 최대 곡률을 가지는 두 점을 선택하기 위해 부가적인 곡률 추정(curvature estimation) 절차가 필요하므로 우세점 추출 시간이 길어진다. 둘째와 셋째 방법은 주어진 객체의 모양 특성을 고려하지 않으므로 최악의 경우 그림 1과 같은 의사점 문제가 발생할 수 있다[3].

즉, 사각형을 이루는 윤곽점에 대해 최종적으로 (d)와 같은 우세점들이 추출되어야 하나 (a)에서처럼 시작점

A와 끝점 B를 초기 세그먼트로 지정하였을 때의 최종 결과는 (c)가 되어 불필요한 의사점을 포함하게 된다. 따라서, 초기에 (b)와 같이 모양의 윤곽선 특징을 고려하여 시작점과 끝점을 올바르게 지정할 수 있는 일관된 방법이 필요하다. 이러한 문제를 해결하기 위해 Ramer의 방법을 개선한 분할 합병 방법이 제안되기도 하였다. 이는 우선 분할을 통해 임계값을 만족하는 후보 우세점들을 구하고 합병 과정에서 후보 우세점에 대한 다각형 근사를 재수행하는 방법을 사용한다. 이 방법은 임의의 시작점 지정에 대한 1차적인 현상을 줄일 수 있지만 다각형 근사를 재수행하는 데 따르는 부가적인 알고리즘 수행 시간이 필요하다.

초기 세그먼트 지정 문제를 해결한 재귀적 윤곽 우세점 추출 방법에서는 무게중심으로부터 가장 먼 점을 시작점으로 지정하고 시작점으로부터 가장 먼 점을 끝점으로 하여 초기 세그먼트를 결정하는 알고리즘을 제시하였다[11]. 이 방법은 객체의 모양을 반영한 자동적이고 일관된 시작점 지정이 가능하나 고정된 임계값을 사용함으로써 그림 2와 같이 객체의 윤곽점들이 이루는 굴곡(curvature)에 따라 누락점 또는 의사점 문제가 발생한다.

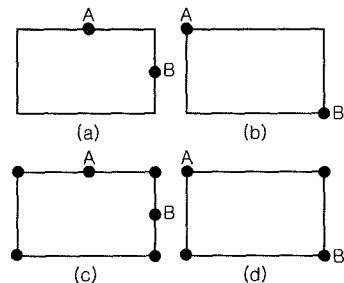


그림 1 초기 세그먼트에 따른 의사점 문제

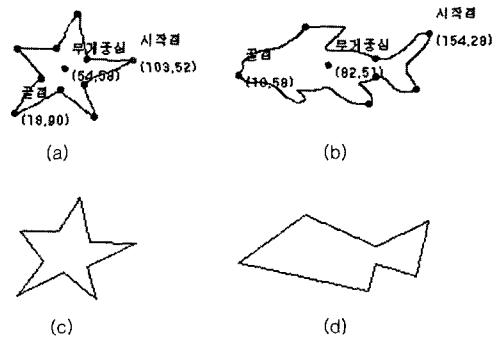


그림 2 고정 임계값 사용의 이상 현상

그림 2에서 (c)와 (d)는 각각 (a)와 (b)의 원본 객체 모양에 대해 재귀적 윤곽 우세점 추출 알고리즘을 적용한 결과이다. (c)의 경우에는 각 세그먼트의 굴곡이 임계값을 모두 만족하므로 근사화한 모양과 원본 객체의 모양이 일치하지만 (d)의 경우에는 완만한 굴곡의 포함이 많으므로 원본 객체의 모양 특징 정보가 훼손된 결과를 볼 수 있다. 이러한 현상을 해결하기 위해서는 윤곽선 특징을 반영하는 동적 임계값 결정 방법이 필요하다.

### 3. 최적 우세점 추출

#### 3.1 최적 우세점의 정의

$n$  개의 윤곽선 점으로 표현한 원래 객체의 모양 특징  $P$ 를 순서있는 점 집합  $\{p_1, p_2, \dots, p_n\}$ 라 할 때,  $P$ 에 대해 다각형 근사 알고리즘으로 얻어진 객체의 모양 특징  $I$ 는  $I \subset P$ 이고  $3 \leq m \leq n$ 인  $m$  개의 점으로 구성된 순서있는 점 집합  $\{q_1, q_2, \dots, q_m\}$ 으로 표현할 수 있다.  $m$ 의 최소 값이 3인 것은 가장 적은 수의 점으로 이루어진 다각형은 삼각형이기 때문이다. 이 때 우세점, 최적 우세점, 누락점과 의사점을 다음과 같이 정의한다.

##### 정의 1. 우세점

객체 모양을 구성하는  $n$ 개의 윤곽선 점에 대해 다각형 근사 알고리즘을 수행하는 함수를  $A$ 라 할 때,  $A: P \rightarrow I$ 이고  $q_i \in I$ ,  $1 \leq i \leq m$ 인 각 점  $q_i$ 를 우세점이라 정의한다. □

##### 정의 2. 최적 우세점

$P$ 에 대해 시각적으로 판별되는  $m'$ 개의 우세점에 대해  $A: P \rightarrow I$ 를 통해  $m = m' + a$ 개의 우세점을 구했을 때  $I \approx P$ 를 만족하고  $m' - a$ 에 대해  $I \neq P$ 가 될 때  $I$ 에 포함된  $m$ 개의 각 우세점  $q_i$ 를 최적 우세점이라 정의한다. □

이 정의는 함수  $A$ 에 의해 얻어지는 우세점의 개수가 인간이 식별한 우세점의 개수와 항상 일치할 수 없다는 것을 가정한다. 따라서 우세점 수 증가에 따른  $P$ 와  $I$ 의 유사도 변화율이 안정화되어 일정한 값으로 수렴하기 시작하는 우세점 개수 구간  $-a \leq m' \leq a$ 일 때  $I$ 에 속한 우세점들을 최적 우세점이라 정의한다. 즉 최적 우세점 개수를  $m \approx m' \pm a$ 로 간주한다. 이 논문에서는 실험을 통해 우세점 추가에 따른  $P$ 에 대한  $I$ 의 유사도 변화를 측정해 유사도에 대한 기울기 변화가 안정화된 구간의  $m$ 을 기초로  $a$ 의 범위를  $-2 \leq a \leq 2$ 로 정하였다.

##### 정의 3. 누락점과 의사점

만약,  $A: P \rightarrow I$ 를 통해  $\beta > a$ 인  $m = \beta$  개의 우세점을 얻

었을 때 최적 우세점 개수  $m$ 이  $m - \beta$ 인 경우  $\beta$ 개의 점을 누락점이라 정의한다. 또한  $m + \beta$ 인 경우  $\beta$ 개의 점을 의사점이라 정의한다. □

#### 3.2 무게중심과 최대 길이 접촉점을 이용한 시작점 선정

이 논문에서는 주어진 객체의 모양 특성을 고려하고 일관된 방법으로 시작점을 지정하기 위해 기존에 본 연구실에서 제안한 재귀적 윤곽 우세점 추출 알고리즘에서 사용하는 모양에 대한 무게중심(center of gravity)을 적용한다. 무게중심은 동일한 객체 모양인 경우 크기나 회전에 무관하며 유사한 객체의 모양인 경우에도 일정한 값을 유지하는 특성을 보인다[11]. 무게중심을 이용한 초기 세그먼트의 시작점과 끝점 선정 방법은 그림 3과 같다.

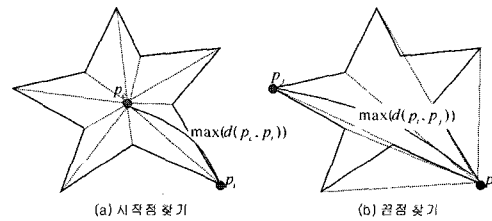


그림 3 초기 세그먼트의 시작점과 끝점 선정

윤곽선 세그먼트를 나누기 위한 초기 시작점은 구해진 무게중심을  $p_c$ 라 할 때,  $p_c$ 와  $n$  개의 점으로 구성된  $P$ 의 임의의 점  $p_i$ 들에 대해  $n$  번의 유클리드 거리  $d(p_c, p_i)$ 를 계산하여  $\max(d(p_c, p_i))$ 인  $p_i$ 를 선택한다. 초기 세그먼트의 끝점은 구해진 시작점  $p_i$ 를 기준으로  $p_i \neq p_j$ 인  $P$ 에 속하는 모든 점  $p_j$ 들에 대해  $n-1$  번의  $d(p_i, p_j)$ 를 계산하여  $\max(d(p_i, p_j))$ 인  $p_j$ 를 지정한다.

#### 3.3 최적 우세점 추출을 위한 동적 임계값 결정 방법

윤곽선 세그먼트 분할을 위한 시작점  $p_i$ 와 끝점  $p_j$ 를 선택하면 이들로 이루어지는 세그먼트  $\widehat{p_i p_j}$ 상의 임의의 점  $p_k$ 를 차례로 조사하여  $p_k$ 를 우세점 집합에 포함할지 여부를 결정해야 한다. 이에 대한 결정은  $p_k$ 로부터 윤곽선 세그먼트의 시작점  $p_i$ 와 끝점  $p_j$ 를 연결한 직선  $\overline{p_i p_j}$ 으로 내린 수선의 길이를 이용하는 방법[7], 무게중심  $p_c$ 에 대해  $\forall p_k \in S$ 이고  $\max(d(p_c, p_k))$ 인 초기 시작점의 길이를  $L$ 이라고 할 때 임의의 고정된  $\delta$  값을 사용하여  $\epsilon = L/\delta$ 를 구하여 사용하는 방법[11] 등이 있다. 이 외에  $P$ 와  $I$ 에 대한 면적비를 사용하는 방법을 고려해 볼 수 있다.

첫째 방법에서는  $d(\overline{p}, p_i, p_k) < \epsilon$  인지를 검사하기 위해 고정된 하나의 상수  $\epsilon$  값을 사용하므로써 각 윤곽선 세그먼트의 굴곡 특성을 정확하게 반영하기 어렵다. 또한 윤곽선 세그먼트 내의 최대 수선 길이를 단순히 임계값과 비교함으로써 약간의 노이즈를 가진 점을 포함하고 있는 경우 불필요한 의사점이 최종 우세점 집합에 다수 포함된다. 둘째 방법은 무게중심으로부터 가장 긴 초기 시작점의 길이  $L$ 을 반영함으로써 객체 모양에 따라  $\epsilon$  이 가변적일 수 있으나  $\delta$ 의 값을 고정시키는 단점이 있다. 면적비를 고려하는 방법은 직선 길이만을 계산하는 방법에 비해 계산량이 많으므로 실용적이지 못하다. 위에서 살펴본 방법들처럼 단순히 임계값을 고정시키는 방법은 간단하고 계산량이 적지만 객체의 모양에 따라 윤곽선 세그먼트 내의 임계값들이 일정치 않으며 윤곽선 세그먼트의 굴곡 특성을 잘 반영할 수 없는 단점이 있다.

이 논문에서는 이러한 문제점을 개선하고 최적 우세점을 추출하기 위해 원래 객체 모양  $P$ 의 윤곽선 길이  $L$ 과  $I$ 에 속한 우세점들로 구성되는 다각형 모양의 윤곽선 길이  $I$ 에 대한 비율  $\sigma = I/L$ 을 동적으로 검사하여 우세점 추가 여부를 결정하는 방법을 제안한다. 이는 우세점 추가에 따라  $I$ 에 속한 점들로 구성되는 근사화된 다각형의 모양과  $P$ 의 윤곽선 모양의 유사도가 점점 높아지며  $\sigma$ 의 값이 1.0에 수렴하는 성질을 이용한 것이다. 예를 들어  $n$ 개의 점으로 구성된  $P$ 를  $m \leq n$ 인  $m$ 개의 우세점들로 구성된 세그먼트  $s_k$ 들로 분할하여 다각형 근사화를 수행한다고 할 때  $I = \sum_{k=1}^m length(s_k)$ 이라 하자. 이 때 최적 우세점 정의에 의해  $m$ 개의 우세점만으로 최적 우세점 집합  $I$ 를 얻을 수 있다면  $I \approx P$ 가 되며 윤곽선 길이 비율은  $\sigma \approx 1.0$ 이 된다. 만약  $m + \beta$ 인 경우는  $P$ 에 대해 이미  $m$ 개의 우세점들로 최적 다각형 근사화가 이루어지므로  $\sigma$ 의 값은  $\beta$ 개의 의사점 추가에 대해 거의 변화가 없다. 최적 우세점의 개수  $m$ 은 논리적으로  $3 \leq m \leq n$ 의 범위를 가진다. 즉, 2차원 평면을 이루는 최소 다각형은 삼각형이므로 최소한 3개의 우세점을 얻을 수 있으며 최대  $n$ 개의 우세점이 구해질 수 있다. 실제로  $n$ 개의 윤곽점에 대해  $m$ 은  $m \ll n$ 으로 나타난다. 실제 실험에서는 객체 모양에 따라  $0.90 \leq \sigma \leq 0.98$ 의 범위에서 최적 우세점들이 구해졌다. 이는 객체 모양에 따라 가변적인 임계값이 적용되어야 함을 의미한다.

3.4 최적 우세점 추출 알고리즘과 성능 분석

우세점 추출시 윤곽선 길이 비를 사용할 때는 우세점

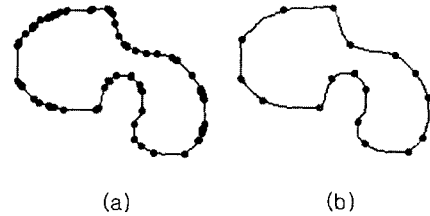


그림 4 우세점 추가 순서에 따른 이상 현상

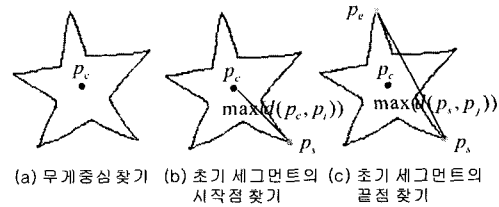


그림 5 알고리즘 초기화 과정의 예

선정 순서가 중요하다. 만약 재귀적 방법을 사용하는 경우에는 한 세그먼트를 더 이상 분할할 수 없을 때까지 지역적인 우세점들을 구한 후 다음 세그먼트에 대해 다각형 근사화를 수행하므로 객체 모양에 따라 그림 4의 (a)와 같이 우세점이 한 쪽으로 몰리거나 불필요하게 많은 의사점이 추출되는 현상이 발생한다. 이러한 현상은 우세점의 선정이 지역적으로 이루어지지 않고 전체 윤곽선 세그먼트에 대해서 균등한 기회로 이루어져야 함을 의미한다. (b)는 이 논문에서 제시하는 최적 우세점 추출 알고리즘을 수행한 결과이다.

제안하는 최적 우세점 추출 알고리즘은 폐곡선으로 이루어진 모양을 입력으로 처리하며 만약 입력된 모양이 개곡선인 경우는 곡선의 시작과 끝점이 연결되어 있는 것으로 간주하여 알고리즘을 수행한다. 알고리즘의 구성은 총 9단계로 크게 초기화 단계와 우세점 추출을 위한 반복적인 검사 단계로 나뉜다. 그림 5는 초기화 단계의 수행 과정을 나타낸 것이다.

알고리즘의 1단계부터 4단계까지가 초기화 단계이다. 1단계에서는 그림 5의 (a)와 같이 입력된 모양  $P$ 로부터 무게중심점  $p_c$ 를 구한다. 단계 2에서는 그림 5의 (b)와 같이  $p_c$ 로부터 가장 먼 점을 초기 세그먼트의 시작점  $p_s$ 로 지정하고 이를 첫번째 우세점으로 삼는다. 단계 3에서는 그림 5의 (c)와 같이  $p_s \neq p_e$ 인  $P$ 에 속한 임의의 점  $p_i$  중에서  $p_s$ 로부터 가장 먼 거리의 점을 찾아 초기 세그먼트의 끝점  $p_e$ 로 지정한다. 구해진  $p_e$ 는 두 번째

우세점이 된다. 단계 4는 초기화 과정의 마무리 단계로 초기  $\sigma$ 값 결정, 알고리즘 종료를 위한 플래그 설정과 세그먼트 정보에 대한 초기치를 결정한다. 알고리즘 초기화 과정 수행 후 우세점 집합  $I$ 에는 두 개의 우세점  $p_s$ 와  $p_e$ 가 순서대로 포함된다. 이 때 우세점 개수  $m$ 의 값은 2이다.

초기화 과정이 끝나면 구해진 초기 세그먼트를 시작으로 실제 우세점 검사와 선정을 위한 반복문을 수행한다. 이 반복문은 다섯 단계를 포함하며 알고리즘에서 5 단계부터 9 단계까지이다. 이 반복문은 더 이상 우세점 추가가 없거나  $\sigma$ 가 1.0에 수렴하면 알고리즘을 종료한다. 단계 5에서는 세그먼트 집합  $S$ 에서 세그먼트  $s_i$ 를 선택한다. 여기서  $i$ 는  $1 \leq i \leq m$ 이며 우세점 집합에 속한  $q_i$ 를 시작점,  $q_{i+1}$ 를 끝점으로 하는  $s_i$  중에서  $\max(d(\overline{q_i q_{i+1}}, p_k))$ 인 점  $p_k$ 를 가지는 세그먼트의 번호이다.  $s_i$ 를 선택하는 과정은  $S$ 에 속한 각  $s_i$ 들이 수선 길이에 대해 오름차순으로 구성되어 있으므로  $S$ 에서 가장 우선 순위가 높은  $s_i$ 를 고르면 된다. 단계 6부터 단계 9는  $s_i$ 에 지정된 수선 길이가  $\tau$ 를 만족하는 경우의 처리 과정과 만족하지 않을 경우 알고리즘 종료를 위한  $flag$ 를 설정하는 과정으로 나누어진다.  $s_i$ 의 수선 길이가  $\tau$ 를 만족할 경우는 단계 6의 우세점 집합 변경, 단계 7의 세그먼트 정보 변경과 단계 8의  $\sigma$ 값 변경을 차례로 수행한 후 반복문을 다시 수행한다. 단계 6의 우세점 집합 변경에서는 우세점으로 선택된  $p_k$ 를 우세점 집합  $I$ 에 추가한다. 추가되는 위치는 선택한  $s_i$ 를 이루는 시작점  $q_i$ 와 끝점  $q_{i+1}$  사이가 된다.  $s_i$ 는 새로 추가된 우세점  $p_k$ 를 기준으로 두 개의 세그먼트  $s_u = \overline{q_i p_k}$ 와  $s_v = \overline{p_k q_{i+1}}$ 로 분할된다. 단계 7에서는 이 두 세그먼트를 세그먼트 집합에 차례로 추가하고 각각에 대한 최대 수선 길이  $\max(d(\overline{q_i p_k}, p_u))$ 와  $\max(d(\overline{p_k q_{i+1}}, p_v))$ 를 구해 세그먼트 정보 집합  $S$ 를 변경한다. 이렇게 함으로써 단계 5 수행시  $\max(d(\overline{q_i q_{i+1}}, p_k))$ 인 점  $p_k$ 를 모든 세그먼트에 대해 다시 계산하지 않아도 되므로 알고리즘 수행 시간을 줄일 수 있다. 단계 8에서는 새로 우세점이 추가되어  $I$ 에 속한 우세점들로 이루어지는 다각형의 둘레 길이가 변경되었으므로 새로  $l$ 을 구하고  $ll$ 을 계산해  $\sigma$  값을 수정한다. 위의 각 단계를 수행하는 알고리즘 OptimalInterestPoint는 다음과 같다.

Algorithm OptimalInterestPoint()  
/\*

$n$  : 원 객체 모양의 윤곽선을 구성하는 점의 개수  
 $m$  : 최적 다각형 근사화된 모양의 우세점의 개수  
 $P = \{p_1, p_2, \dots, p_n\}$  : 원 객체 모양의 윤곽선 점의 순서 집합  
 $I = \{q_1, q_2, \dots, q_m\}$  :  $P$ 의 점들 중  $3 \leq m \leq n$ 인  $m$  개의 점  $q_i$ 들로 구성된 우세점의 순서 집합  
 $S = \{s_1, \dots, s_i, \dots, s_m\}$  :  $I$ 의 점  $q_i$ 와  $q_{i+1}$ 로 구성되는  $s_i = \overline{q_i q_{i+1}}$ 인 세그먼트 순서 집합  
 각  $s_i$ 는  $\max(d(\overline{q_i q_{i+1}}, p_k))$ 일 때의 수선길이와 점  $p_k$ 에 대한 정보를 가짐  
 단,  $m$ 번째 세그먼트의 시작점은  $q_m$ 이며 끝점은  $q_1$ 임  
 $d(p_i, p_j)$  : 임의의 점  $p_i$ 와  $p_j$ 의 유클리드 거리  
 $d(\overline{q_i q_{i+1}}, p_k)$  :  $s_i$ 의 시작점  $q_i$ 와 끝점  $q_{i+1}$ 의 구간 존재하는  $P$ 에 속한 한 점  $p_k$ 에서  $\overline{q_i q_{i+1}}$ 로 내린 수선의 길이  
 $L$  :  $P$ 에 속한 점으로 만들어지는 모양의 윤곽선 둘레 길이  
 $l$  :  $I$ 에 속한 점으로 만들어지는 근사화된 다각형의 윤곽선 둘레 길이  
 $\alpha$  :  $P$ 와  $I$ 에 대한 윤곽선 둘레 길이 비율,  $ll/L$   
 $flag$  : 우세점 추가 여부를 나타내는 플래그  
 $\epsilon$  : 다각형 근사화 정도를 나타내는 임계값  
 $\tau$  : 디지털 영상에서 시각적으로 판별되는 최소 수선 길이  
 \*/  
 begin  
 /\* 알고리즘 초기화 단계 \*/  
 /\* 단계 1: 무게중심 찾기 \*/  
 find centroid  $p_c$  of  $P$   
 /\* 단계 2: 첫 번째 우세점인 초기 세그먼트의 시작점 찾기 \*/  
 find  $p_s$ , where  $p_s$  is  $\forall p_i \in P$  and  $\max(d(p_c, p_i))$   
 $I = \{p_s\}$   
 /\* 단계 3: 두 번째 우세점인 초기 세그먼트의 끝점 찾기 \*/  
 find  $p_e$ , where  $p_e$  is  $p_i \in P$ ,  $p_i \neq p_s$ , and  $\max(d(p_s, p_i))$   
 $I = I \cup \{p_e\}$   
 /\* 단계 4:  $\alpha$  플래그, 세그먼트 정보에 대한 초기치 결정 \*/  
 calculate  $L$  of  $P$  and  $l$  of  $I$

```

 $\sigma \leftarrow l/L, \text{flag} \leftarrow \text{TRUE}, m=2$ 
 $S = \{s_1, s_2\}$ 
for each  $s_i$ , modify segment information, where
 $1 \leq i \leq 2$ 
/* 우세점 결정 단계 */
while ( $\sigma \leq \epsilon$  and  $\text{flag}$ )
begin
/* 단계 5: S로부터 가장 큰 수선 거리를 가지는 세
  세그먼트  $s_i$ 를 선택 */
find  $i$ , where  $\forall s_i \in S, 1 \leq i \leq m$  and
 $\max(d(\overline{q_i q_{i+1}}, p_k))$ 
if ( $\max(d(\overline{q_i q_{i+1}}, p_k)) \geq \tau$ )
begin
/* 단계 6: 우세점으로 선택된  $p_k$ 를 우세점 집합  $I$ 에
  추가 */
 $I = I \cup \{p_k\}$ , where the position of  $p_k$  in  $I$  is
  between  $q_i$  and  $q_{i+1}$ 
 $m \leftarrow m+1$ 
/* 단계 7:  $p_k$ 에 의해  $S$ 에 새로 추가되는 세그먼트
 $s_u$ 와  $s_v$ 의 세그먼트 정보 변경 */
 $S = S - \{s_i\} \cup \{s_u\} \cup \{s_v\}$ 
  modify segment information of  $s_u$  and  $s_v$ 
/* 단계 8:  $I$ 에 속한 우세점들로 구성된 모양의 윤곽
  선 길이와  $\sigma$  값을 수정 */
 $k \leftarrow k - |\overline{q_i q_{i+1}}| + |\overline{q_i p_k}| + |\overline{p_k q_{i+1}}|, \sigma \leftarrow l/L$ 
end /* end of if */
else
/* 단계 9: 우세점의 추가가 없는 경우 플래그를
  FALSE로 지정 */
 $\text{flag} \leftarrow \text{FALSE}$ 
end /*end of while*/
end

```

위의 알고리즘을 수행하는 과정의 실제 예를 세그먼트 분할을 통해 우세점이 추가되는 순서를 중심으로 그림 6에 제시하였다. 이 예는 그림 5의 알고리즘 초기화 이후의 반복문 수행시 우세점 결정 과정을 나타낸다.

알고리즘 초기화 수행 후 얻은 세그먼트 집합에는  $s_1 = \overline{q_1 q_2}$ 와  $s_2 = \overline{q_2 q_1}$ 이 있다. 그림 6의 (a)를 보면  $s_1, s_2$ 에 대한  $\max(d(\overline{q_i q_{i+1}}, p_k))$ 는 알고리즘의 단계 4에서 각각 ①과 ②로 구해진 상태이다. 단계 5에서는 이 둘을 비교해 ① > ②이므로  $s_1$ 을 선택하고  $s_1$ 에 대해

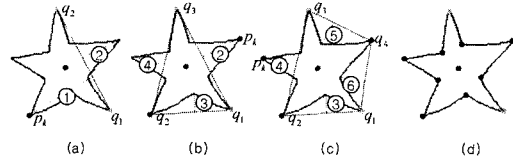


그림 6 우세점 결정 과정

$\max(d(\overline{q_1 q_2}, p_k))$ 가  $\tau$ 보다 크거나 같으므로 단계 6에서는 이 때의 점  $p_k$ 를  $I$ 에 추가하여  $I = \{q_1, p_{k-2}, q_3\}$ 가 된다. 이때 추가되는 위치는  $q_1$ 과  $q_2$ 사이가 되므로 (b)와 같이 우세점의 순서가 바뀌게 되고  $m$ 의 값은 3이 된다. 단계 7에서  $s_1$ 이  $s_u$ 와  $s_v$ 로 분할되므로  $S = \{s_{u=1}, s_{v=2}, s_3\}$ 가 되며 새로 추가된 두 세그먼트에 대한 수선 길이 ③과 ④를 계산해 각 세그먼트의 정보로 지정한다. ②는 이미 구한 값이므로 다시 계산할 필요가 없다. (b)에서는 ②를 계산한 점을, (c)에서는 ⑤를 계산한 점을 각각 우세점으로 추가해 가면서 우세점 결정 과정을 반복 수행한 후 알고리즘이 종료되면 (d)의 결과를 얻는다.

제안한 알고리즘의 시간 복잡도는 다음과 같다. 성능에 영향을 미치는 단계는 초기화 단계, 우선 순위를 가지는 힙(heap)로 운영되는 세그먼트 집합  $S$ 의 입출력 시간을 포함하는 단계 5와 단계 7과 새로 추가되는 두 세그먼트에 대한 수선 길이를 계산하는 단계 7이다. 나머지 단계들은 상수 시간내에 처리가 가능하다. 우세점 결정을 위한 단계 5에서 단계 9까지는 반복 구조를 이루고 있지만 최대  $m$ 개 만큼의 순환 후 알고리즘을 종료하게 된다. 따라서 알고리즘의 수행 시간은  $n + \sum_{k=3}^m (\log k + \frac{n}{2^{k-2}})$ 가 된다.  $k$ 가 3부터 시작하는 이유는 초기화 단계에서 초기 세그먼트의 시작점과 끝점으로 삼는 두 개의 우세점을 이미 계산하였기 때문이다. 이 식을 풀어쓰면  $2n + m \log m - m$ 보다 작거나 같게 된다. 여기서  $m = cn$ 이며 상수  $c$ 는  $0 < c \ll 1.0$ 의 범위의 값을 가진다. 따라서 제안한 알고리즘의 평균 시간 복잡도는  $\approx O(n \log n)$ 이다.

#### 4. 실험과 분석

##### 4.1 실험 방법과 내용

제안한 우세점 최적화 알고리즘에 대한 검증과 평가를 위해 두 가지 실험을 수행하였다. 첫째 실험은 윤곽선 굴곡 특성에 따른 우세점 최적화율을 평가하기 위해 7가지 서로 다른 윤곽선 굴곡 특성 조합에 대해 각각

10개씩의 합성 모양을 만들어 총 70개의 합성 모양을 가지고 수행했다. 굴곡 특성 조합은 원형, 직선으로 구성된 도형, 굴곡으로 구성된 도형, 또아리, 직선과 곡선의 혼합, 전체 윤곽선에 대한 지역적/전역적 노이즈로 분류된다. 각 분류에 속한 10개씩의 모양은 기준 모양에 대해 확대, 축소, 회전, 뒤집기 및 특정 부분에 대한 늘이기와 줄이기 등으로 변형한 모양과 유사 모양을 포함한다. 실험에 대한 평가는 제안한 알고리즘 수행 결과에 대해 우세점 개수, 무게중심 점의 값, 원본 모양과 근사화된 모양의 면적 및 윤곽선 길이에 대한 측정 결과를 분석한 후 이 중 최적 우세점 추출 알고리즘의 특성을 평가할 수 있는 최적화 평가 함수를 고안하여 수행하였다. 둘째 실험은 비교 실험으로 영국 surrey 대학교에서 모양의 윤곽선에 기반한 모양 분류와 검색을 실험하기 위해 공개적으로 제공한 1,100개의 어류 모양[12]을 가지고 재귀적 윤곽 우세점 추출 알고리즘과 제안한 우세점 최적화 알고리즘의 최적화율을 비교하였다. 실험 환경은 NT 기반에서 Visual C++ 6.0으로 구현하였다.

4.2 제안한 알고리즘에 대한 평가 함수

다각형 근사 알고리즘에 대한 평가는 원본과 근사화된 모양의 윤곽점 개수에 대한 압축비(compression ratio; CR), 각 세그먼트를 구성하는 호와 근사화된 다각형에 대한 이차 적분 오류(integral square error; ISE), CR과 ISE의 비를 측정하는 방법, 다각형 근사에 대한 충실도(fidelity)와 효율성(efficiency)을 측정하는 방법 등이 제시되었다[8].

이 실험에서는 평가 함수를 구성하기 위해 우세점 개수 비(interest points ratio; IPR), 무게중심 연관성(relateness of centroid, RC), 원본 모양과 근사화된 모양의 면적비(area ratio; AR)와 윤곽선 길이비(contour ratio; CR)를 분석하였다. 각 평가식은

$$IPR = \left| \frac{alg_{\#pt}}{opt_{\#pt}} \right|, \quad RC = \left| \frac{\sum_{p_i \in alg} d(p_i, p_c)}{\sum_{p_i \in org} d(p_i, p_c)} \right|, \quad AR = \left| \frac{\Delta alg}{\Delta org} \right|,$$

$CR = \frac{l(alg)}{l(org)}$  이다. 평가식의 기호 중 *org*, *alg*, *opt*는 원본 모양, 제안한 알고리즘에 의해 얻은 근사화된 도형, 시각적 판별로 구한 최적 근사화된 도형을 각각 의미한다. #*pt*는 우세점의 수를 의미한다. 기호  $\Delta$ 은 면적, 함수 *l*( $\cdot$ )은 윤곽선 길이를 의미한다. RC에 사용한  $p_i$ 와  $p_c$ 는 각 도형의 윤곽선의 우세점이며  $p_c$ 는 무게중심을 의미한다. 여기서 IPR, RC, AR은 제안한 알고리즘의 측정값이 원본 또는 최적 모양의 측정값 보다 클 수 있으므로 절대비율로 정규화하였다. RC에 대한 측정에서

는 우세점들로만 구성된 모양에 대해서 모두 무게중심의 변동이 없었으며 AR 또한 평균 0.96으로 모양별 특성에 대한 구별이 이루어지지 않았다. IPR과 CR에 대한 측정값에 대해서는 모양 특성이 반영되는 결과를 보였다. 각 평가 기준에 대한 분석을 통해 이 논문에서는 제시한 최적 우세점 추출 알고리즘의 평가를 위해 IPR과 CR을 사용하여 우세점 최적화 평가 함수  $f = \sqrt{IPR * CR}$ 을 구성하였다. 이 함수의 값은 [0:1] 사이며 원본 모양과 근사화된 모양이 완전 일치한다면 1이다.

4.3 모양 특성에 따른 제안한 알고리즘의 우세점 최적화 성능 평가

여기서는 제안한 알고리즘으로 추출한 우세점이 최적 인지에 대한 검증을 수행한 후 70 개의 서로 다른 윤곽선 특성을 가지는 합성 이미지에 대한 실험 결과를 분석한다. 그림 7은 실험에 사용한 모양 특성별 대표 이미지를 표시한 것이다.

최적 우세점 검증에서는 그림 7에 제시한 대표적인 7개의 모양들에 대해 시각적으로 식별한 우세점의 수를 *m*으로 하여 알고리즘 수행 과정에서 +3개의 우세점 변화에 대한  $\sigma$ 값을 측정하였다. 이에 대한 결과는 그림 8에 제시하였다. 이를 분석해 보면 기울기 변화가 급격한 구간은 원본 모양의 윤곽선과 우세점에 의해 근사화되는 다각형의 모양의 유사도가 낮은 경우임을 알 수 있다. 기울기 변화가 안정되어 가는 구간은  $\sigma$ 의 변화가 적은 구간으로 더 이상의 우세점 추가가 불필요하다고 판단할 수 있다. 그림 8을 보면  $m \pm 2$ 의 구간에서  $\sigma$ 의 기울기 변화가 안정화됨을 알 수 있다. 이는 3.1에서 설명한 최적 우세점 정의를 만족한다. 따라서 제안한 알고리즘에 의해 얻어진 우세점들은 최적 우세점이 된다.

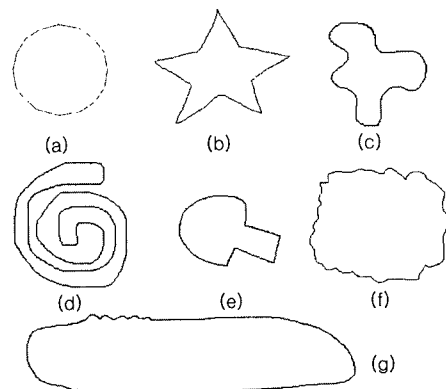


그림 7 합성 모양의 윤곽선 특성



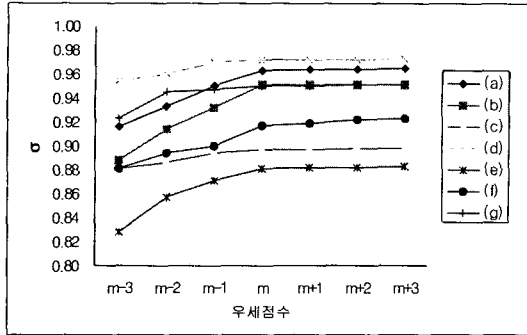


그림 8 우세점 추가에 따른  $\sigma$  변화

표 1 윤곽선 특성에 따른 최적화 성능

구분	(a)	(b)	(c)	(d)	(e)	(f)	(g)
#pt of P	282.80	294.38	474.20	473.00	424.33	517.80	536.00
#pt of I	8.00	10.13	17.30	23.29	11.67	13.80	30.25
CR	0.93	0.95	0.87	0.95	0.91	0.94	0.93
IPR	1.00	0.99	0.85	0.89	0.83	0.94	0.84
f	0.97	0.97	0.86	0.92	0.87	0.94	0.88

제안한 최적 우세점 추출 알고리즘에 대한 모양 특성에 따른 성능 평가는 표 1과 같다. CR은 윤곽선 전체에 대해서 그림 7의 (c)처럼 유선형을 가지는 굴곡이 많을수록 값이 작아지는 특성을 보인다. IPR은 (a), (b)처럼 시각적으로 단순한 모양이나 (f)와 같이 윤곽선 전체적으로 작은 노이즈를 가지는 모양에 대해서 높은 값을 나타냈다. 이는 원본 모양의 윤곽선과 제시한 알고리즘으로 다각형 근사화한 윤곽선이 거의 유사함을 의미한다.

전체적인 평가 척도인 f를 보면 세그먼트내의 곡률 변동이 적은 모양일수록 제시한 알고리즘의 성능이 높게 나타난다. 곡률 변동이 심한 (c), (e), (g)에 대해서는 우세점 개수의 수가 가시적으로 식별한 기준 우세점의 개수보다 많거나 적게 나오는 경우로 인해 성능 평가 값이 약간 낮게 나타났다. 실험에 사용한 70개의 합성 모양에 대한 평균 f는 0.92를 보였다.

4.4 우세점 최적화 성능 비교 분석

우세점 최적화에 대한 비교 실험을 위한 실험은 1,100개의 어류 모양[12]을 가지고 수행하였다. 이 논문에서 제안한 우세점 최적화 알고리즘이 재귀적 윤곽 우세점 추출 알고리즘을 개선한 것이므로 이 알고리즘을 비교 대상으로 선택하였다. 성능 비교 평가는 첫째 실험에서 사용한 CR, IPR, f를 사용하였다. 표 2는 두 알고리즘

표 2 우세점 최적화 성능 비교

구분	CR	IPR	f
재귀적 윤곽 우세점 추출 알고리즘	0.84	0.73	0.78
우세점 최적화 알고리즘	0.90	0.89	0.89

의 우세점 최적화 성능을 나타낸 것이다.

표 2의 성능 결과를 분석해 보면 제안한 알고리즘으로 추출한 우세점 개수가 최적 우세점 개수에 더 가깝게 나타남을 알 수 있다. 이는 재귀적 윤곽 우세점 추출 알고리즘은 실험에 사용한 모양의 곡률이 임계값을 만족하지 못하는 경우 누락점이 많이 발생되며 임계값을 만족하는 윤곽점이 많을 수록 의사점이 증가했기 때문이다. 이미지 검색과 분류에 사용하기 위한 인덱스를 구성하기 위해서는 적은 수의 우세점으로 원래의 모양 특징 정보를 정확하게 반영하기 위한 우세점에 대한 최적화가 우선적으로 고려되어야 한다. 제안한 우세점 최적화 알고리즘은 비교 대상인 재귀적 윤곽 우세점 추출 알고리즘과 같이  $O(n \log n)$ 의 평균 시간 복잡도를 갖지만 우세점 최적화에 대해 14%정도의 성능 향상을 보였다.

5. 결론

이 논문에서는 모양 기반 이미지 검색과 분류를 위한 전처리 과정으로서 인덱스 구성에 필요한 모양 특징 정보인 우세점을 최적화하기 위한 알고리즘을 제시하였다. 이는 본 연구실에서 기존에 제안한 재귀적 윤곽 우세점 추출 알고리즘이 가지는 고정 임계값 사용에 따른 이상 현상을 개선한 것이다. 개선된 알고리즘에서는 원본 객체 모양의 윤곽선 길이와 근사화한 모양의 윤곽선 길이 비를 알고리즘 수행시 동적으로 적용하여 누락점 또는 의사점의 수를 줄임으로써 추출할 우세점의 수를 최적화한다. 제안한 방법에서는 n개의 윤곽점에서 m개의 최적 우세점이 추출되었을 때  $O(n \log n)$ 의 성능을 보인다. 실험을 통해  $m \ll n$ 이며 원본 객체의 n개 윤곽점에 대해 추출한 우세점의 개수 m의 비율은 평균 3%정도로 나타났다. 제안한 알고리즘의 성능 분석을 위한 실험은 7가지의 서로 다른 윤곽선 곡률 특성을 가지는 70개의 합성된 모양과 일반 이미지로부터 추출한 1,100개의 어류 이미지를 가지고 수행하였다. 평가 기준은 실험에 사용한 각 모양에 대한 우세점 개수비와 윤곽선 길이비를 구하고 우세점 최적화 평가 함수를 구성하여 최적화 정도를 측정하였다. 실험 결과 평균 0.92의 최적화율을 보였으며 재귀적 윤곽 우세점 추출 알고리즘과의 최적

화율 비교 평가에서는 약 14% 정도의 개선을 보였다.

우세점 최적화 알고리즘을 통해 추출한 우세점은 정규화를 통해 객체의 크기 변화나 회전에 무관한 모양 특징 값으로 변환할 수 있으며 모양을 이루는 윤곽선의 굴곡에 따라 서로 다른 특징 값을 보이므로 모양 기반 이미지 검색과 분류를 위한 인덱스 키로 활용 가능하다. 향후 연구로는 최적화한 우세점 집합에 대한 정규화된 모양 특징 값에 나타난 모양 클래스별 특성을 분류할 수 있는 방법에 대한 연구가 필요하다.

### 참 고 문 헌

- [1] R. Baldock, J. Graham, *Image Processing and Analysis*, Oxford University Press, 2000.
- [2] Danny Z. Chen, Ovidiu Daescu, "Space efficient Algorithms for Approximating Polygonal Curves in Two Dimensional Space," *The Fourth Annual International Computing and Combinatorics Conference (COCOON)*, pp.55-64, 1998.
- [3] Luciano da Fontoura Costa, Roberto Marcondes Cesar Jr., *Shape Analysis and Classification : Theory and Practice*, CRC Press, 2001.
- [4] Kikuo Fujimura, Yusaku sako, "Shape Signature by Deformation," *Shape Modeling and Applications*, IEEE, pp.225-232, 1999.
- [5] H. Imai, M. Iri, "Computational Geometric Methods for Polygonal Approximation of a Curve," *Computer Vision, Graphics and Image Processing*, pp.31-41, 1986.
- [6] E. Miliotis, E. Petrakis, "Shape Retrieval Based on Dynamic Programming," *IEEE Transactions on Image Processing*, Vol. 9, No.1, pp.141-147, 2000.
- [7] U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Computer Graphics and Image Processing*, Vol.1, pp.244-256, 1972.
- [8] Paul L. Rosin, "Techniques for Assessing Polygonal Approximations of Curves," *Transactions on Pattern Analysis and Machine Intelligence*, IEEE, Vol.19, No.6, pp.659-666, 1997.
- [9] M. Safar, C. Shahabi, X. Sun, "Image Retrieval by Shape : A Comparative Study," *Multimedia and Expo. IEEE*, Vol.1, pp.141-144, 2000.
- [10] J. Wang, W. Chang, and R. Acharya, "Efficient and Effective Similar Shape Retrieval," *Multimedia Computing and Systems*, IEEE, pp.875-879, 1999.
- [11] 김영태, 엄기현, "객체 모양의 특징을 표현하는 재귀적 윤곽 우세 점 추출 방안", *한국정보과학회 봄 학술발표 논문집*, 제28권 제1호, pp.19-21, 2001.
- [12] SQUID, <http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html>



조 성 태

1991년 동국대학교 공과대학 전산학과 공학사. 1993년 동국대학교 대학원 컴퓨터공학과 공학석사. 2001년 동국대학교 대학원 컴퓨터공학과 공학박사. 1995년~1996년 동국대학교 공과대학 컴퓨터공학과 강사. 1996년 경민대학 사무자동화과, 성신여대 전산학과 강사. 1997년 3월~현재 경민대학 인터넷정보과 조교수. 관심분야는 멀티미디어 데이터베이스, 인터넷기반 정보 시스템



엄 기 현

1975년 서울대학교 공과대학 응용수학과 공학사. 1997년 한국과학기술원 전산학과 이학석사. 1994년 서울대학교 대학원 컴퓨터공학과 공학박사. 1978년 3월~현재 동국대학교 컴퓨터멀티미디어공학과 정교수. 1995년 3월~1999년 2월 동국대학교 정보관리처장 역임. 2001년 3월~2003년 2월 동국대학교 정보산업대학 학장 역임. 1997년~2002년 한국정보과학회 이사. 1998년 9월~2000년 8월 데이터베이스 연구회 운영위원장. 2001년~2002년 한국정보과학회 논문지(데이터베이스) 편집부위원장. 관심분야는 멀티미디어 데이터베이스, 정보 시스템