

# 굴곡 기반 형태 그래프를 이용한 모양 검색

## (Shape Retrieval using Curvature-based Morphological Graphs)

방 난 호 <sup>†</sup>    엄 기 현 <sup>\*\*</sup>  
(Nanhyo Bang)    (Kyhyun Um)

**요 약** 모양 데이터는 이미지가 나타내는 의미를 가장 잘 반영하는 데이터로서 이미지 검색에 중요한 정보로 사용된다. 특히 구조적으로 표현된 모양 특징은 모양이 갖는 기초적 특성과 그들간의 관계 정보를 잘 나타내므로 폭넓게 연구되고 있다. 그러나 대개의 구조적 모양 특징들은 그래프나 트리와 같은 구조로 표현되므로 모양 데이터 검색에서 효율적인 검색 시간을 보장할 수 없는 문제를 지니고 있다. 이러한 문제를 해결하기 위하여 본 논문에서는 모양의 윤곽선 정보를 기반으로한 굴곡 기반 형태 그래프를 생성하고 이를 일반화한 구조로부터 모양을 클러스터링할 수 있는 키를 설계한다. 제안한 굴곡 기반 형태 그래프는 모양이 가지고 있는 윤곽선 특성과 영역의 형태적 특성을 모두 가지고 있다. 모양 검색은 단계적으로 이루어진다. 클러스터링을 통해 검색 공간을 축소하고 외부 굴곡 특징을 이용한 굴곡의 패턴 매칭을 통해 종합적인 유사도가 결정된다. 다양한 실험을 통해 굴곡 기반 형태 그래프와 클러스터링을 통해 검색 공간과 비용이 줄어드는 것을 보여준다.

**키워드** : 모양 기반 이미지 검색, 굴곡 기반 형태 그래프, 클러스터링

**Abstract** A shape data is used one of most important feature for image retrieval as data to reflect meaning of image. Especially, structural feature of shape is widely studied because it represents primitive properties of shape and relation information between basic units well. However, most structural features of shape have the problem that it is not able to guarantee an efficient search time because the features are expressed as graph or tree. In order to solve this problem, we generate curvature-based morphological graph, and design key to cluster shapes from this graph. Proposed this graph have contour features and morphological features of a shape. Shape retrieval is accomplished by stages. We reduce a search space through clustering, and determine total similarity value through pattern matching of external curvature. Various experiments show that our approach reduces computational complexity and retrieval cost.

**Key words** : Shape-based Image Retrieval, Curvature-based Morphological Graph, Clustering

### 1. 서 론

이미지는 동영상의 기본 단위로 사용할 뿐만 아니라 시각적 정보 검색 시스템(얼굴, 지문 인식 시스템, 예술품, 도안 관리 시스템, 위상 판측 시스템, 웹 등)의 광범위한 활용 등으로 인해 가장 범용적인 데이터가 되었다. 특히 모양 특징은 기본적인 색상, 질감 특징과는 달리 이미지가 내포하고 있는 다양한 의미를 반영하는 논리적 특징으로서 활발하게 연구되고 있다. 모양 특징을 표현하는 다양한 방법 중, 구조적 표현 방법은 모양 객체

가 갖는 기초적인 단위와 그들간의 관계적인 정보를 간결하게 나타낼 수 있다는 큰 장점을 가지고 있다[1]. 대개는 다단계 자료 구조를 이용하거나 그래프, 트리 등의 구조를 사용하여 모양의 구조적 특징을 표현하고 있다. 이러한 특징들은 효과적으로 모양을 표현할 수 있다는 장점은 있으나 고차원 또는 구조의 복잡성으로 인해 검색 비용을 증가시키는 문제(특히, 그래프 매칭)를 가져왔다.

본 논문에서는 구조적으로 분석된 모양 특징을 이용하여 표현성을 높이고, 이로부터 클러스터링을 위한 키를 추출하여 매칭 공간을 줄이는 검색 방법을 제안한다. 구조적 모양 특징을 위해 본 논문에서는 굴곡 기반 형태 그래프(Curvature-based Morphological Graph)를 제안한다. 굴곡 기반 형태 그래프는 모양의 윤곽선으로부터 추출한 외부 굴곡과 내부 영역의 골격선 정보를

<sup>†</sup> 정 회 원 : 유한대학 정보통신학과 교수  
jjang@yuhan.ac.kr

<sup>\*\*</sup> 종신회원 : 동국대학교 컴퓨터멀티미디어공학과 교수  
(Corresponding author)  
khum@dgu.ac.kr

논문접수 : 2005년 6월 13일

심사완료 : 2005년 8월 3일

기반으로 생성된 일종의 ARG(Attributed Relationship Graph)이다. 따라서 윤곽선 기반의 모양 특징 방법과 영역 기반의 모양 특징 방법의 장점을 모두 가지고 있다. 내부 영역의 골격선 추출은 재귀적으로 수행되며 굴곡의 형태에 따라 그래프의 각 노드가 특정한 속성으로 분류된다. 굴곡 기반 형태 그래프는 모양의 전역적인 내부 형태를 나타내므로 이 형태 정보를 일반화한 정보로부터 키를 추출하여 모양들을 여러 개의 클러스터로 분류한다. 검색시에는 유사 클러스터군으로 분류된 모양들에 대해서만 매칭을 수행하므로 검색 공간과 비용을 줄일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 모양의 구조적 특징을 연구한 관련 연구들을 살펴보고 3장에서는 굴곡 기반 형태 그래프의 생성 과정을 설명한다. 4장에서는 굴곡 기반 형태 그래프를 일반화하고 이로부터 클러스터링 키를 생성하는 방법과 이를 이용한 검색 방법을 설명한다. 5장에서는 제안한 방법을 이용한 검색의 효율성을 실험하고 그 결과를 분석하며, 6장에서는 결론을 맺는다.

**2. 관련 연구**

굴곡 기반 형태 그래프의 굴곡은 모양의 윤곽선으로부터 추출된다. 윤곽선 점은 모양에 대한 가장 직관적인 정보를 담고 있으므로 많은 연구에서 사용된다. 윤곽선 점을 이용한 모양 특징 연구에서는 윤곽선을 이루는 점들을 기반으로 한 매칭 전략이나 이를 통해 추출한 기타 정보(곡률 등)를 이용하는 방법, 차원을 줄이면서 모양의 특징을 그대로 표현하는 우세점 특징 추출을 주로 다루고 있다[2,3]. 윤곽선점을 이용하는 모양 표현 방법 중 대표적인 것은 굴곡을 기반으로 모양을 표현하는 것이다[4-7]. 굴곡을 이용한 연구의 가장 큰 문제점은 모양의 내부 정보를 반영하지 않는다는 것이다. 굴곡을 이용한 모양 특징은 대칭과 회전, 크기 변화 등에 민감하다는 단점을 가지고 있다[5].

골격 그래프는 전체 모양의 형태를 구조적으로 잘 표현한 특징이다. 골격 그래프는 영역 기반 모양 표현 기법의 일부이며, 영역 그래프라고도 하고 모양의 형태 구조를 반영할 경우 형태(morphological) 그래프라고도 한다. 골격 그래프는 영역 정보를 바탕으로 모양의 구조 정보를 잘 나타낸다는 장점을 가지고 있어서 모양의 전역적 비교나 클러스터링에 주로 사용된다. 특히 shock 그래프는 기하학적이고 동적인 정보를 가지고 있는 그래프로써 중앙축(medial axis) 그래프보다 더 많은 서술자를 갖는다[5,8-10]. 이 방법은 계산 비용이 높은 단점은 있으나 모양을 이루는 각 부분들간의 배치와 연결 구조 등이 잘 표현되고 매칭된다는 장점이 있다.

윤곽선을 이용한 특징이나 영역 특징들은 다차원의 가변적인 특성을 가지고 있어서 매칭과 유사도 검색시 심각한 성능 저하를 가져온다[11]. 모양의 구조적 정보를 잘 표현하는 shock 그래프는 검색 비용이 증대하는 그래프 매칭의 문제를 그대로 가지고 있다. 대부분의 연구에서 shock 그래프를 매칭하기 위해 편집 연산을 사용하여 그 비용을 계산한다[12]. 그러나 이런 방법은 계산 비용이 너무 높은 단점이 있다. 이로 인해 shock 그래프나 중심 축 그래프를 트리로 변환하여 트리 매칭을 통해 검색의 성능을 높이는 연구도 진행되고 있다 [13,14].

본 논문에서는 모양의 윤곽선 특징을 이용하여 영역의 형태를 표현하는 굴곡 기반 형태 그래프를 생성하고 이를 이용한 효율적인 검색 방법을 제안한다. 제안한 굴곡 기반 형태 그래프는 굴곡 기반 모양 특징의 장점을 그대로 가지면서 클러스터링을 통해 비교적 간단한 비용으로 특징 추출과 매칭이 가능하다. 본 논문에서 형태 그래프 표현을 위해 생성한 골격선은 정확한 영역의 중심축을 나타내는 선은 아니다. 분할된 블록 다각형 기반의 굴곡 세그먼트로부터 골격선에 근사한(approximate) 내부선을 추출한다. 본 논문에서는 골격선 기반의 형태 그래프를 세부 매칭을 위해 사용하는 것이 아니라 영역을 기반으로한 특징 정보를 추출하여 모양의 형태를 분류하기 위해 사용한다.

**3. 굴곡 기반 형태 그래프**

본 연구에서는 모양의 윤곽선 특징을 기반으로 기본 단위인 굴곡을 추출하고, 굴곡들이 이루는 영역 정보로부터 모양의 형태 정보를 생성한다. 영역 특성을 나타내는 형태 정보는 그래프로 표현되고 윤곽선 정보를 포함하고 있는 외부 굴곡은 구조 리스트로 표현되어 세부 매칭에 사용된다. 굴곡을 기반으로 윤곽선 특징과 영역 특징을 모두 사용하므로 기존 연구들의 문제점을 크게 개선하였다. 본 논문에서 제안하는 굴곡 기반 형태 그래프는 모양을 구성하는 내부 영역들을 블록 도형으로 분할하며 이 영역내의 우세점으로부터 중심점들을 연결하여 형태 그래프를 생성한다.

**3.1 논리적 굴곡 세그먼트**

굴곡은 모양의 윤곽선으로부터 가장 기본적인 모양 단위가 되는 직선, 그 직선들의 급격한 관계 변화를 보이는 부분을 단위로 분리한 것이다. 한 객체의 모양을 윤곽선점으로 표현하면 정보량이 너무 크므로 모양의 특성을 반영하는 우세점( $p_i$ )을 추출하여 기본 특징으로 사용한다(우세점 집합,  $IP = \{p_1, p_2, \dots, p_n\}$ ). 우세점은 모양을 구성하는 윤곽선 점에 대해 무게 중심을 이용한 다

각형 근사 알고리즘을 수행하여 추출한다[15]. 본 논문에서는 굴곡 정보의 기본 단위로 가장 쉽게 추출할 수 있는 볼록(convex) 굴곡과 오목(concave) 굴곡을 이용한다. 볼록, 오목 굴곡은 우세점들의 내부각 특징만으로 쉽게 추출이 가능하다. 내부각 특징은 이웃하는 두 점과의 사잇각을 의미한다. 내부각 추출 함수( $fIA()$ )는 다음과 같다:  $fIA(p_i) = \alpha_i = \angle p_{i-1}p_i p_{i+1}, (0 < \alpha_i < 360)$ . 내부각이 180도 이상인 우세점은 오목한 선분 세그먼트를 이루는 것이며, 180도 이하인 점은 볼록 세그먼트를 이루고 있는 것이다.

볼록 굴곡은 굴곡의 시작과 끝점을 제외하고는 모든 점의 내부각이 180도 이하인 굴곡이다. 굴곡의 시작점과 끝점을 연결할 경우 하나의 볼록 도형이 만들어진다. 볼록 굴곡 세그먼트의 구조는 다음과 같다.

$$CbCS = \{p_j | is \leq j \leq ie, fIA(p_{is}), fIA(p_{ie}) > 180, \forall_{k=is+1}^{ie-1} fIA(p_k) < 180\}$$

모양의 외부 굴곡을 가장 잘 표현하는 볼록 굴곡들은 볼록 도형을 표현하는 특징들로 나타낼 수 있다. 오목 굴곡 세그먼트의 구조는 다음과 같다.

$$CcCS = \{p_j | is \leq j \leq ie, \forall fIA(p_j) > 180\}$$

모양의 외부 굴곡은 볼록 굴곡과 오목 굴곡으로 분류되지만 속성을 표현하는 특징들은 같다. 외부 굴곡은 굴곡간의 관계를 나타내는 구조적 특징과 굴곡의 논리적 특성에 따라 정의되는 적응적 특징으로 나타낸다[16]. 기존의 굴곡 세그먼트는 정의된 논리적 특성에 따라 재분류된다. 굴곡이 갖는 기초 특징들을 이용하여 굴곡의 논리적 특성을 분류(완전 볼록, 완전 오목, 구부러진 굴곡, 회전하는 굴곡)한다. 논리적 특성이 정해진 굴곡 세그먼트를 LCS(Logical CS), 논리적인 굴곡 세그먼트라 정의한다. 그러므로 모양은 LCS의 집합으로 이루어지며 각 LCS들은 그 특성을 반영하는 적응적 특징과 이웃 LCS와의 구조적 특징을 갖게 된다. 따라서 모양의 외부 굴곡은 하나의 LCSS(LCS Sequence)로 표현된다 :  $LCSS = \{LCS_i | LCS_i = \langle p_{list}, pt, sflist, aflist \rangle, 1 \leq i \leq s\}$ .

$p_{list}$ 는 우세점 집합,  $pt$ 는 논리적 특성의 유형을 가리키고  $sflist$ 는 구조적 특징,  $aflist$ 는 적응적 특징을 나타낸다. LCSS는 모양과 함께 저장되며 검색시 외부 굴곡 매칭 과정에서 사용된다.

### 3.2 굴곡 기반 형태 그래프

모양의 단위 데이터가 되는 굴곡들간의 관계는 모양의 전역적인 특징으로서 굴곡들간의 구성이나 배치 형태를 표현한다. 모양을 이루는 각 부분들간의 관계를 가장 잘 반영한 것이 중심 축 그래프, 골격선 그래프, shock 그래프이다. 그러나 이들은 생성과 검색 비용이 높은 단점을 가지고 있다. 본 논문에서는 우세점으로 이

루어진 볼록 굴곡 세그먼트를 기반으로 쉽게 중심축과 유사한 골격 그래프를 추출하는 방법을 제안한다. 이는 크게 모양을 볼록 굴곡 단위로 분할하는 과정과 분할된 굴곡내에서의 골격선 추출, 이 골격선들간의 병합 과정으로 이루어진다. 본 논문에서는 이러한 유사 골격 그래프를 굴곡 기반 형태 그래프(Curvature-based Morphological Graph)라 정의한다. 이 그래프는 전역적인 굴곡의 관계와 내부 영역의 특성을 반영한다.

#### 3.2.1 재귀적인 내부 영역 분할

모양의 외부 굴곡으로부터 영역 정보를 담고 있는 내부 영역( $IS$ )을 추출한다. 이를 위해 최초의 내부 영역  $IS$  생성 과정에서 외부 굴곡과 연결할 형태 그래프의 단말 노드( $Inode$ )를 생성한다.

$$Inode = (LCS_i.p_{is} + LCS_i.p_{ie})/2, \text{ which} \\ \text{if } LCS_i.pt \neq \text{concave} (1 \leq i \leq s)$$

[정의 1 :  $Inode$ ] 볼록 굴곡 세그먼트 성질을 갖는 외부 굴곡 세그먼트의 처음과 마지막 우세점을 이은 선분에 대한 이등분할점 □

내부 영역은 볼록 굴곡 세그먼트의 굴곡 기준선 ( $\overline{p_{is}p_{ie}}$ )을 한 번으로 하는 도형이다. 내부 영역  $IS$ 는 다음과 같이 정의된다.

$$IS \stackrel{\text{def}}{=} \bigcup SP_k, SP_k = \langle p_k, mpidx \rangle \\ p_k = \begin{cases} LCS_i.p_{ie} & (1 \leq i \leq s, LCS_i.pt = \text{convex}) \\ \forall_{i=is+1}^{ie} LCS_i.p_i & (LCS_i.pt = \text{concave}) \end{cases} \\ mpidx = \begin{cases} Idx(Inode_i) & (LCS_i.pt = \text{convex}) \\ -1 & (LCS_i.pt = \text{concave}) \end{cases}$$

내부 영역  $IS$ 는 볼록 도형이거나 하나의 선분만 남을 때까지 재귀적으로 분할된다. 분할된 도형들로부터 골격선 추출을 하게 되는데,  $mpidx$ 는 외접하는 볼록 굴곡과 연결하기 위한 포인터 역할을 한다. 본 논문에서 제시하는 영역 분할의 기본 단위는 회전 굴곡을 포함한 볼록 굴곡 세그먼트이다. 외부 굴곡 세그먼트로부터 최초의  $IS$ 를 만든 것과 동일한 메카니즘으로  $IS$ 를 생성한다.

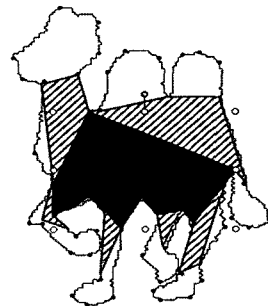


그림 1 IS로부터 이차 영역 분할된 내부영역

위의 그림 1에서 빗금친 부분이 IS이고 가운데에 보이는 색깔한 부분이 이차 분할로 만든 IS'이다.

형태 그래프 생성은 기본적으로는 분할과 병합의 과정이다. 내부 영역 IS로부터 내부 굴곡 세그먼트(ICS: Inner Curvature Segment)를 추출하여 다시 IS'을 생성한다. 분할 과정은 IS'이 하나의 볼록 도형이거나 하나의 선분만 남을 때까지 반복된다. 한번의 IS' 생성을 위해 분할된 내부 굴곡 세그먼트들은 자체적인 골격선 분할 과정을 거쳐 외접하는 골격선들과 병합된다.

```

Algorithm Extract_CMGraph() // 내부 영역 IS의 재귀적 분할
입력 : IS
출력 : IS'
Begin
(1) 초기화
    IS' ← NULL
(2) 재귀 함수의 종결 조건 검사
    if (IS == ConvexPolygon)
        ConvexSplit(IS)
    if (IS의 우세점수가 2이하)
        return
(3) IS내에 모든 우세점들에 대해 내부 굴곡 세그먼트 분할
    else {
        while( ∃ SPk , SPk ∈ IS ) {
            볼록 굴곡 성질을 갖는 ICS 생성
            오목 굴곡 성질을 갖는 ICS 생성 {
                SPk'·pk = ∇i=is+1ie ICSi·pi (SPk' ∈ IS')
                SPk'·mpidx = -1
            }
            continue
        }
        회전 굴곡 성질을 갖는 ICS 생성
(4) 분할된 ICS내의 골격선과 외접하는 골격선과의 병합
        if (ICSi == 회전 굴곡 세그먼트) //내부 굴곡 세그먼트의 골격선 추출
            mpidx = Extract_tICS_Bone(ICSi)
        else
            mpidx = Extract_CvICS_Bone(ICSi)
        SPk'·pk = ICSi·pie
        SPk'·mpidx = mpidx
    }
(5) 재귀 함수 호출
    Extract_CMGraph(IS')
}
End
    
```

(4)번의 과정에서 호출하는 Extract\_tICS\_Bone() 함수나 Extract\_CvICS\_Bone() 함수가 내부 굴곡의 골격선을 추출하는 함수이다. Extract\_tICS\_Bone() 함수나 Extract\_CvICS\_Bone() 함수는 ICS를 대상으로 하는

것으로 골격선을 추출할 때 해당 ICS와 외접하는 굴곡들과의 노드 연결을 고려한다. 재귀적 함수에 의해 실제로 (4)번 과정은 부분적인 골격선들간의 병합을 수행하는 부분이기도 하다.

### 3.2.2 교차연결 기반의 골격선 생성

내부 영역 생성을 위해 분할된 ICS들은 분할과 함께 골격선 추출 과정을 거친다. 생성된 골격선은 모양의 형태를 나타내고 골격선을 이루는 중심점들이 그래프로 표현된다. 본 논문에서 제시하는 골격선은 굴곡의 우세점을 이용하여 굴곡 단위로 재귀적으로 수행되므로 기존 방법보다 계산 비용이 낮은 장점이 있다. 또한 본 논문에서 제시한 골격선은 shock 그래프처럼 풍부한 모양 정보의 서술이 가능하다. 따라서 이러한 정보를 이용한 단계적인 세부 매칭도 가능하다.

형태 그래프를 생성할 골격선의 중심점들은  $GV_{shock}$ 로 나타낸다. 내부 굴곡 세그먼트의 골격선 추출은 굴곡 세그먼트의 기준선( $\overline{p_{is}p_{ie}}$ )으로부터 시작된다.  $\overline{p_{is}p_{ie}}$ 를 이등분하여 중심점  $GV_j$ 를 생성한다. 이때 고려해야할 문제는 다음 중심점  $GV_j$ 를 생성하는 방법이다.  $p_{is}$ ,  $p_{ie}$ 의 굴곡내 다음 전후 우세점으로부터  $GV_j$ 를 생성한다면 이는 영역을 사각형 분할하는 것과 같다. 연속적인 양방향 우세점을 연결할 때는 다음과 같은 문제가 발생한다. 다음 그림 2에서  $\overline{p_{is+2}p_{ie-2}}$ 를 연결한다면 올바른 중심점을 구할 수 없다. 선분의 중심점이 영역의 내부에 있지 않기 때문이다. 예외를 적용하여,  $p_{is+2}$ 와  $p_{ie-1}$ 을 연결한다면 이 부분은 삼각 분할이 되고 그 다음은 사각( $\overline{p_{is+3}p_{ie-2}}$ ), 그리고는 삼각형이 남는다.

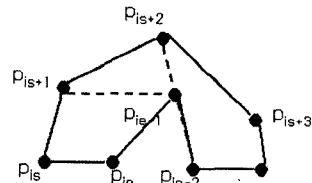


그림 2 연속 우세점으로 중심축을 구하는 예

[골격선의 중심점  $GV_j$ 의 성질 1] 기본적으로 모든 도형은 삼각형으로 분할이 되므로 중심축을 구할 우세점 쌍은 삼각형 분할을 기본으로 하는 것이 적합하다. □

그러므로 본 논문에서는 삼각형 분할을 원칙으로 기준선의 인접 우세점을 교차연결(지그재그)하여 골격선의 중심점을 추출한다. 먼저 기준선을 이루는  $p_{is}$ ,  $p_{ie}$ 의 인접 우세점중 하나를 교차연결하여 삼각형을 분할한다. 이 교차연결선분( $\overline{CL}$ )을 기준선으로 대체하고 다시 한번

교차연결을 하면  $\overline{CL}$ 과  $\overline{p_{is+1}p_{ie-1}}$ 을 변으로 하는 또다른 삼각형이 만들어진다. 이때 고려할 것이 두 삼각형에 맞닿은 선분  $\overline{CL}$ 의 선택이다.  $\overline{p_{is}p_{ie}}$ 의 이등분할점을  $GV_i$ 로 하고  $\overline{p_{is+1}p_{ie-1}}$ 을  $GV_k$ 로 할 때,  $GV_j$ 는  $\overline{CL}$ 의 이등분할점으로부터 생성된다.  $\overline{CL}$ 은  $\overline{p_{is}p_{ie-1}}$ 이거나  $\overline{p_{ie}p_{is+1}}$ 이다. 그러나 일반적인 블록 다각형의 중심점을 고려할 때  $GV_j$ 는 다음과 같은 성질(2)을 만족해야 한다.

[골격선의 중심점  $GV_j$ 의 성질 2]

$$\overline{GV_iGV_jGV_k} \approx \overline{GV_iGV_k} \quad \square$$

이를 위해  $\overline{CL}$ 은 다음의 최소값을 갖는 선분으로 설정한다.

$$\min(\text{dist}(\overline{GV_iGV_k}, (p_{is} + p_{ie-1})/2), \text{dist}(\overline{GV_iGV_k}, (p_{ie} + p_{is+1})/2))$$

교차연결로 우세점을 연결하는 것은 삼각형 단위 분할을 하는 결과가 되고 더욱 정밀하게 모양의 변화를 표현할 수 있다.  $\overline{CL}$ 의 이등분할점으로부터 생성된 노드들을 중간 노드(*mnode*)라 한다.

[정의 2 : *mnode*] 골격 기준선을 중심으로 우세점의 교차연결로 만들어진 선분에 대한 이등분할점  $\square$

생성된 중간 노드는 그 이전 선분의 중간 노드와 연결된다. 두 노드를 연결하는 선분이 골격선이 되고 이것이 형태 그래프의 에지(edge)가 된다. 이 때 외접하는 골곡 세그먼트와 병합이 발생한다. 분할된 삼각형의 세 변 중  $\overline{CL}$ 과 기준선을 제외한 나머지 한 변의 *mpidx*가 더미 값(-1)이 아니라면 외접하는 골곡 세그먼트가 있다는 것이다. 이 외접하는 골곡 세그먼트의 골곡 기준선의 중심점(*mpidx*)과 *mnode*를 에지로 연결, 병합한다.

또한 삼각형 분할을 위한 인접 우세점의 교차연결시에는 교차연결의 기준선과 교차선 사이의 내부각이 180도 이상일 경우가 있다. 이것은 내부각  $\angle p_{is}p_{ie}p_{ie-1}$ 이 180도가 넘었다는 것이다. 이는  $\square p_{is}p_{ie}p_{is+1}p_{ie-1}$ 이 오목 사각형이거나 선분  $\overline{p_{is}p_{ie}}$ 와  $\overline{p_{is+1}p_{ie-1}}$ 이 교차하고 있다는 것을 의미하는데, ICS가 회전하는 골곡 세그먼트인 경우 발생할 수 있다. 이 때는 교차연결을 중단하고 블록 도형으로 재분할되도록 교차선을 역으로 결정한다. 이로부터 만들어진 이등분할점 노드는 회전 노드(*tnode*)라 정의한다.

[정의 3 : *tnode*] 하나의 ICS내에서 블록 도형 분할이 일어난 기준 선분에 대한 이등분할점  $\square$

이로부터 생성된 골곡 기반 형태 그래프의 예는 다음 그림 3과 같다.

다음 그림 3에 나타난 바와 같이 골곡 기반 형태 그래프의 말단 노드(노드의 디그리=1)는 모두 외부 골

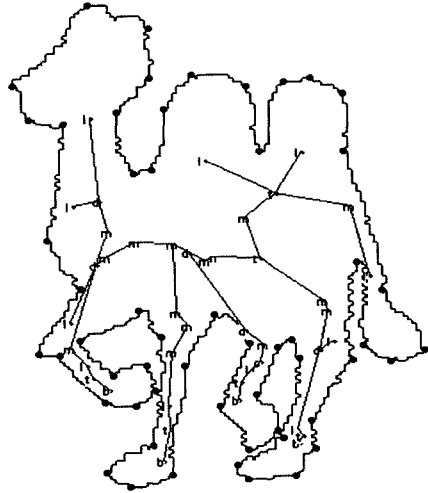


그림 3 낙타의 골곡 기반 형태 그래프 예

곡이다. 외부 골곡들은 우세점의 교차선 이등분할을 하지 않았다. 외부 골곡들은 LCSS를 이용하여 블록 도형 단위의 유사도를 기반으로 골곡 매칭을 수행한다.

#### 4. 골곡 기반 형태 그래프를 이용한 검색

3장에서는 모양의 논리적 특성을 반영하여 윤곽선 기반의 LCSS와 내부 영역의 특성을 반영하는 골곡 기반의 형태 그래프를 생성하였다. 이를 이용하여 검색 단계에서는 전역적인 형태 정보 매칭과 구조적, 지역적인 골곡 정보를 이용한 외부 골곡 매칭을 단계적으로 수행한다. 내부 영역의 형태 정보에 따른 모양의 분류를 통해 검색 공간을 줄이고, 골곡 정보의 구조적, 지역적 특징들을 이용하여 최종 유사도를 계산한다. 전역적인 매칭 단계에서는 형태 그래프를 이용하여 모양의 전체적인 형태가 유사한 클러스터를 찾는다. 지역적인 매칭 단계에서는 클러스터에 포함된 모양들에 대해 외부 골곡 매칭을 수행한다. 외부 골곡 세그먼트들의 유형과 골곡의 구조적 특징을 사용하여 두 모양의 골곡 세그먼트들 중 비교할 가장 유사한 골곡 세그먼트열을 찾아낸다. 이 세그먼트열에 대해 적응적 특징과 매칭되는 모양의 범위 비율을 적용하여 유사도를 계산한다. 사용자에게는 유사도 값을 정렬하여 결과를 보여준다.

##### 4.1 일반화한 CMG의 클러스터링 키

외부 골곡을 제거해 가면서 추출한 내부 영역으로부터 생성한 형태 그래프는 모양의 전체 구조를 잘 나타낸다. 그러나 골곡으로부터 추출한 형태 그래프는 골곡 기반 표현 방법의 문제점인 작은 골곡의 변화에 너무 민감하게 반응하는 문제를 그대로 가지고 있다. 따라서 사람이 볼 때는 전체적으로 같은 모양이지만 형태 그래

프는 크게 다른 결과를 가져온다. 다음의 그림 4는 이와 같은 사실을 나타낸 예이다.



그림 4 전체적으로 모양은 유사하나 서로 다른 형태 그래프의 예

위의 그림에 나타난 모양들은 외부 글꼴의 모양 변화에 따라 서로 다른 형태 그래프가 추출된 것을 보여준다. 본문에서는 유사한 모양에서 추출한 서로 다른 형태 그래프에서 전역적으로 공통된 특징을 구조화하고 이를 클러스터링 키로 하여 유사한 모양 집단을 검색한다.

$mnode$ ( $tnode$  포함)는 내부 영역의 변형을 나타내는 노드로서 에지로 연결된 이웃 노드의 종류에 따라 두 가지로 분류할 수 있다. 첫째는  $Lmnode$ , 외부 글꼴과 연결된 내부 영역의 최근접 노드로서 한 개 이상의  $lnode$ 와 한 개 이상의  $mnode$ 로 연결되어 있다. 두 번째  $i_mnode$ 는 순수한 내부 영역 내에서 만들어진 노드로서 연결된 모든 노드들이  $mnode$ 인 노드이다.

전역적인 형태 특성을 추출하기 위해 형태 그래프를 일반화한다. 내부 영역  $IS$ 는 오목 우세점들로만 구성되어 있다. 내부 영역은 내부 영역의 구조 정보만을 담고 있는  $inArea$ 와 외부 글꼴과의 연결 정보를 담고 있는  $outArea$ 로 구분된다.  $inArea$ 의 구조는 포함된  $mnode$ 들의 특징을 분석하여 구조화할 수 있다. 일반화된 형태 그래프는  $inArea$ 에 속해있는  $mnode$ 들과  $outArea$ 에 속한  $mnode$ 들을 동일한 기준 노드로 매핑하여 생성한다. 먼저,  $inArea$ 에 속한  $i_mnode$ 들( $i_mnode$ 가 없다면  $Lmnode$ 들)을 동일한 기준 노드( $in\_node$ )로 매핑한다.  $in\_node$ 를 기준으로  $outArea$ 에 속한 노드들과의 연결 정보를 재설정한다. 이들간의 매핑 연산식은 다음과 같다.

$$\forall i_mnode_i(x,y) \rightarrow in\_node(x_c,y_c),$$

$$i_mnode_i \in i_mnode, (1 \leq i \leq al)$$

$in\_node$ 로 매핑된 중심  $mnode$ 들과 에지로 연결된 외부 노드들 또한  $in\_node$ 를 기준으로 재매핑된다. 재매핑되는 이 노드들은  $outArea$ 에 속하며 외부 노드( $out\_node$ )라고 한다.

$$\Delta(dx,dy)_i = in\_node(x_c,y_c) - i_mnode_i(x,y)$$

$$out\_node_{ik}(x',y') = out\_node_{ik}(x,y)$$

$$+ \Delta(dx,dy)_i, (1 \leq ik \leq ol)$$

그림 4로부터 추출된 일반화된 형태 그래프는 다음 그림 5와 같다.

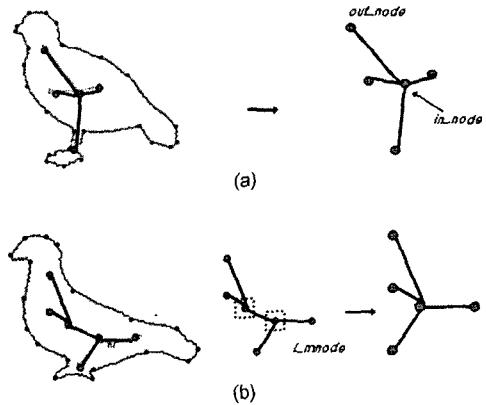


그림 5 일반화된 형태 그래프 예

그림 5의 (b)에서 형태 그래프를 일반화하는 것을 볼 수 있다. 점선 네모로 둘러싸인 노드가  $Lmnode$ 이며, 이러한 두 노드들을 하나의 기준점으로 매핑하여 통합한다. 그림 5의 (a)와 (b)에서 생성된  $out\_node$ 의 각도간 비율이 유사한 것을 볼 수 있다. 일반화된 형태 그래프로부터 클러스터링을 위해 사용할 수 있는 인덱스 키는 다음과 같다.

$$key = (inArea \text{의 type, } out\_node \#, lang, mdegree, fseq)$$

$$lang = \sum_{k=2}^{al-1} fLA(i\_mnode_k),$$

$$mdegree = max(\forall i\_mnode_k.degree)$$

$$fseq = \{f\theta_1, f\theta_2, \dots, f\theta_{ol}\},$$

$$f\theta_i = \angle out\_node_i \in\_node(x_c, y_c) out\_node_2$$

$inArea$ 의 타입은 내부 영역이 하나의 볼록 도형인 경우 생성되는 단일한  $mnode(c)$ , 2개 이상의  $Lmnode$ 의 집합(1),  $i_mnode$ 의 집합(1)이 있다.  $out\_node \#$ 는 기준점과 연결된 외부 노드의 개수를 나타낸다.  $mdegree$ 는 내부 영역이 방사형인지를 나타내며, 선형인 경우  $lang$ 는 회전 정도를 나타낸다.  $fseq$ 는  $out\_node$ 의 방향각 시퀀스로 다음 외부 노드와의 기준점을 중심으로 한 사잇각을 나타낸다. 그림 6은 예제 모양으로부터 추출된 형태 그래프와 그로부터 추출된 원형 기반 특징 정보이다. 낙타와 코끼리의 모양에서 부분적으로 유사한 구조가 나타남을 볼 수 있다.

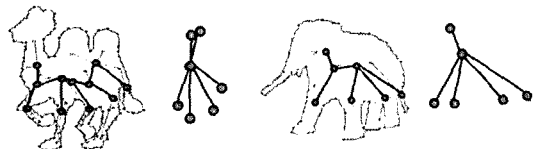


그림 6 부분적으로 유사한 구조를 보이는 일반화된 형태 그래프의 예

#### 4.2 클러스터링을 이용한 단계적 검색

형태 그래프를 이용한 검색은 두 단계로 수행된다. 먼저, 일반화된 형태 그래프의 인덱스 키를 사용하여 후보 집단이 추출된다. 전역적인 모양이 유사한 후보 집단내에서 외부 굴곡을 기반으로 한 매칭이 수행된다. 인덱스 키를 이용한 클러스터링 방법은 다음과 같다.

**Step. 1]** out\_node#가 유사 범위( $k \pm \delta$ )내에 속하는 모양들 추출

**Step. 2]** fseq 비교를 통한 out\_node 들간의 최소 유사도 계산

**Step. 3]** inArea 타입으로 모양들 분류

① 회전 정도(lang)에 따라 모양들 분류

② 방사형(mdegree>>=3)인 모양들 분류

**Step. 4]** 유사 범위( $\pm \delta$ )를 조절하여 검색 공간 조정  
형태 그래프가 유사하다는 것은 내부 영역 정보가 유사하다는 것이므로 외부 굴곡 정보를 이용하여 윤곽선 매칭을 단계적으로 수행한다. 점 대 점으로 매칭하는 것과 달리 굴곡 대 굴곡으로 매칭하는 것은 비교할 대상의 수를 줄일 수 있다. 본 논문에서 사용하는 우세점은 평균적으로 윤곽선점의 1/3이다. 굴곡 세그먼트는 블록 굴곡일 경우, 최소 3개의 우세점으로 구성되며 오목 굴곡일 경우에는 최소 2 개의 우세점으로 구성된다. 오목 굴곡 세그먼트는 연속적으로 나타날 수 없으므로 결국 n개의 우세점으로부터 생성할 수 있는 최대 굴곡 세그먼트 수는 n/2개이다.

굴곡 단위로 모양을 비교할 때는 먼저 비교할 가장 유사한 굴곡 세그먼트 시퀀스를 찾아내고 이 굴곡 세그먼트 시퀀스내에 포함된 굴곡들에 대해 세부 매칭을 수행한다. 본 논문에서는 비교할 가장 유사한 굴곡 세그먼트 열을 찾을 때 연속된 굴곡 세그먼트들간에 구조적 특징을 고려한다. 또한 논리적 특성으로 분류된 굴곡 세그먼트의 유형을 사용하여 매트릭스 내에서의 검색 범위를 줄일 수 있다. 굴곡 세그먼트간의 특징을 비교하기 전에 굴곡 세그먼트의 논리적 패턴에 따라 시작 세그먼트의 쌍을 결정한다. 이것이 세그먼트 패턴 매트릭스(SPM: Segment Pattern Matrix)이다. 매트릭스의 행으로 Q(검색 모양)의 LCSS를, 열축으로 D(데이터베이스내의 모양)의 LCSS를 사용한다. 행의 수는 Q의 s, 열의 수는 D의 s가 된다. r은 매트릭스의 행 인덱스이고( $1 \leq r \leq Q$ 의 s) c는 열 인덱스이다( $1 \leq c \leq D$ 의 s). 매트릭스의 원소는 논리적 굴곡 세그먼트의 유형에 따라 1차적으로 다음과 같이 표현된다 :

$$SPM_{rc}'s\ value = \begin{cases} 1, & Q\ s\ LCS_c.pt = D\ s\ LCS_c.pt \\ 0, & Q\ s\ LCS_c.pt \neq D\ s\ LCS_c.pt \end{cases}$$

본 논문에서는 SPM을 이용하여  $s_q \times s_d$  공간에서 최적의 유사 굴곡 시퀀스(BSCS : Best Similar Curva-

ture Sequence)를 구성할 수 있는 후보 굴곡 시퀀스(CCS : Candidate Curvature Sequence)를 검출한다. CCS는 LCS의 논리적 타입이 연속적으로 모두 일치하는 굴곡 시퀀스이다. BSCS는 CCS 중 유사도가 가장 높은 CCS를 기준으로 전체 LCSS 스캐닝 순서상에서 만나는 다른 CCS들을 병합하여 생성한다. SPM의 원소 중 값이 0인 원소는 CCS가 될 수 없으므로 시작 세그먼트 선택에서 제외된다. SPM의 값이 1인  $SPM_{ij}$  원소에 대해 행, 열 값을 하나씩 증가하며 연속되는 다음 원소의 값이 계속적으로 1로 나타나는지를 검사한다. CCS를 찾기 위해 SPM을 대각선 방향으로 스캐닝하면서 구조적 특징과 적응적 특징의 통합 유사도를 계산한다. 0인 원소의 값을 만나거나 스캐닝한 횟수가 Q와 D의 s 중 하나와 같아지는 LCS까지를 하나의 CCS로 저장한다. 구조는 다음과 같다.

$$CCS = \{CCS_i | CCS_i = \langle r, c, sleng, sval \rangle, 1 \leq i \leq cn\}$$

r과 c는 CCS의 시작 세그먼트 쌍으로서 비교할 시퀀스가 시작되는 Q와 D의 LCS 번호이다. sleng은 시퀀스에 포함된 LCS의 개수이고 sval은 각 CCS의 통합 유사도이다. BSCS는 그 순서로 매칭했을 때 유사도 값이 가장 큰 값이 되었다는 것을 의미한다.

(1) sleng이 min(r,c)와 같다면, 이  $CCS_i$ 가 최적의 유사 굴곡 시퀀스이다. 또한 Q와 D중 어느 한 모양은 모든 LCS가 완전히 매칭에 사용되고 있는 것이다.

(2) sleng이 min(r,c)보다 작다면,  $CCS_i$ 들간의 병합이 진행된다. sleng과 sval 값이 가장 높은  $CCS_i$ 들의 순서대로 병합한다.

BSCS는 질의 모양과 데이터베이스내의 모양에서 매칭할 때 가장 유사도가 높게 계산되는 시퀀스이다. 통합 유사도 합수(TotalCost)는 다음처럼 정의한다.

$$Total\ Cost = \text{Min}\left(\frac{Qr + Dr}{2}, \frac{2Dr}{Qr + Dr}\right) \times BSCS.sval$$

Qr과 Dr은 모양 Q와 D에서 최적의 유사 굴곡 시퀀스에 속한 LCS의 크기 비율을 나타낸다. BSCS의 유사도 값이 완전 매칭으로 1이 나왔다고 하더라도 전체 모양과의 비율이 1이 아니면 전체 유사도 값은 감소된다.

#### 5. 실험 및 분석

본 논문에서 제안한 모양의 형태 그래프와 구조화한 외부 굴곡열의 특징을 실험을 통해 분석한다. 실험할 내용은 크게 두 가지이다. 첫째는 CMG를 통한 클러스터링의 성능을 실험하고 분석한다. 이를 위해 이미 모양이 분류되어 있는 실험 데이터를 사용하여 본 논문에서 제안한 클러스터링 방법의 성능을 분석한다. 둘째는 LCSS를 이용한 전역적-지역적 매칭의 효율성을 실험하

고 분석한다. 검색시 1차적으로 클러스터링된 가장 유사한 후보 모양 집합에 대해 LCSS를 이용한 매칭으로 세부적인 매칭없이 원하는 결과에 접근할 수 있는지를 실험한다.

객관적인 실험을 위해 모양 연구에서 공통적으로 많이 사용하는 MPEG-7의 테스트 데이터를 사용하였다. MPEG-7 테스트 데이터는 새, 사람, 자동차, 낙타, 코끼리 등 이미 의미적으로 분류된 모양들이 다양하게 변형되어 약 20여 개씩 들어있는 테스트 데이터이다. 이미 분류가 되어 있으므로 검색의 정확도를 측정하는 테스트 데이터로 사용한다. 이 데이터들은 흑백의 이미지로 이루어져 있는데 전처리 과정에서 체인코드 찾기를 통해 윤곽선을 추출하고 이로부터 굴곡을 이를 우세점을 추출한다. 검색은 QBE(Query By Example)로 수행하였고 실험을 위한 프로토타입은 MS Visual C++ 6.0으로 구현하였다.

먼저, 일반화된 CMG에서 추출한 키를 가지고 유사한 클러스터군을 생성한다. 이는 검색시 질의 모양과 유사한 클러스터의 생성은 검색 공간을 줄이는데 그 목적이 있다. 클러스터링의 1단계에서  $\delta$ 는  $out\_node\#/2$ 로 실험하였다. 클러스터링 생성시, fseq의 유사도 범위를 조정하여 클러스터의 검색 공간과 오차율의 범위를 실험한 결과는 다음과 같다.

fseq의 유사도를 0.8이상으로 적용하였을 때 생성된 클러스터에 포함된 객체 수는 전체 검색 공간의 평균 13.2% 정도이다. 그러나 평균적인 클러스터링내의 오차율은 8.8%로 높게 나타난다. 이는 비교적 복잡한 모양으로 이루어진 새, 낙타, 코끼리 모양에서 높은 오차율이 나타났기 때문이다. 이러한 현상은 fseq의 유사도를 0.45이상으로 완화하여 검색 공간이 늘어나자(즉 검색할 후보 대상이 증가하자) 크게 감소하였다.

분류된 클러스터 집단 내에서 LCS 시퀀스를 이용한 구조적-적용적 검색을 수행한다. 다음의 그림 7은 대표적인 7개 객체에 대해 분류된 클러스터내에서 LCS 시퀀스간의 유사도 0.7이상인 모양을 검색한 결과를 재현율(recall)과 정확도(precision) 측면에서 분석한 것이다. 클러스터링의 fseq간 유사도가 0.45인 경우는 거의 모든 객체들이 100%의 재현율을 보이므로 그림 7에서는 fseq의 유사도를 0.8이상인 경우로 실험하였다.

LCS 시퀀스를 통한 외부 굴곡의 비교로 후보 집단내

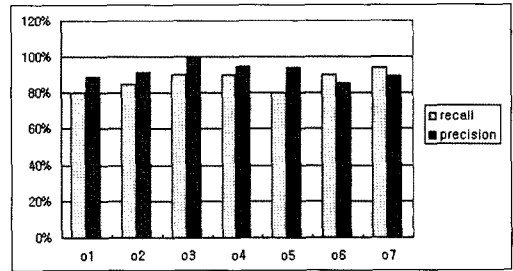


그림 7 검색의 정확도와 재현율 그래프

에 있는 유사한 모양들을 모두 찾아낼 수 있다. 객체 o1과 o5는 새와 낙타 모양의 객체로 CS 수가 많고 이로 인한 다양한 변형이 존재한다. 이는 후보 모양 클러스터의 오차율을 높게 하고 이로 인해 재현율이 낮게 나타난다. 그러나 후보에 포함된 유사 모양에 대해서는 LCS 시퀀스로 비교적 정확하게 찾아내는 것을 볼 수 있다. 높은 재현율을 보이는 o3, o4, o6, o7은 fount, brick, car.class 모양이다. 그러나 o3, o4는 LCS 시퀀스를 통해 검색 결과의 수를 좁혀 정확도가 증가하나, 실제 외형이 거의 유사한 o6, o7은 검색 결과가 조금 늘어나므로 정확도가 감소되는 현상을 나타낸다. 이와 같은 결과는 표 2의 QBE 결과를 통해서도 확인할 수 있다.



































































다음의 표 2는 테스트 데이터셋에 대해 외부 굴곡만을 이용하여 이미지를 검색한 결과이다. 별도의 클러스터링이나 세부적인 점대점 매칭없이도 질의 이미지와 유사하지만 다양한 변형을 갖는 데이터들이 검색되었다. 질의 이미지들에 대해 가장 유사한 결과 이미지 10개를 표 2의 결과로 나타내었다. 표 2의 질의 이미지와 결과 이미지들은 세그먼트 패턴 매트릭스의 LCS 패턴 매칭 전략에 따라 검색된다. 이는 같은 패턴을 가진 연속적인 세그먼트 쌍들 중 특장값의 유사도가 가장 높은 것을 찾아내어 정렬한 것이다. 첫 번째 행의 질의 이미지(fount)는 LCS의 패턴 타입이 "22122212"타입으로 결과에 제시된 이미지들 또한 같은 패턴을 갖는 모양들이 추출되었다. LCSS 매칭시에는 LCS의 패턴열이 같으면서 각 CS들의 유사도 합이 가장 큰 시퀀스를 찾아내게 된다. 가령, 다섯 번째 행의 질의 이미지(misk)는 "22221222"의 패턴 타입을 갖고, 두 번째 결과 이미지(misk10)은 "222122221"의 패턴 타입을 갖는다. 세그먼트 패턴 매트릭스는 모양은 유사하나 중심점에 의해 각

표 1 유사범위 조정에 따른 검색 공간과 오차율 관계

fseq의 유사도	검색 공간(클러스터별)			클러스터링내 평균 오차율
	최소	최대	평균	
≥0.8	5%	19.5%	13.2%	8.8%
≥0.45	11%	46%	30%	1.9%



표 2 두 번째 테스트 데이터를 이용한 QBE 결과

질의 이미지	결과 이미지들				
	 fount13.bmp similarity : 1.00	 fount19.bmp similarity : 0.96	 fount06.bmp similarity : 0.94	 fount20.bmp similarity : 0.94	 fount16.bmp similarity : 0.93
	 fount15.bmp similarity : 0.92	 fount03.bmp similarity : 0.89	 fount05.bmp similarity : 0.89	 fount10.bmp similarity : 0.87	 fount09.bmp similarity : 0.85
	 brick03.bmp similarity : 1.00	 brick18.bmp similarity : 0.96	 brick05.bmp similarity : 0.96	 brick08.bmp similarity : 0.95	 brick16.bmp similarity : 0.95
	 brick15.bmp similarity : 0.94	 brick10.bmp similarity : 0.93	 brick20.bmp similarity : 0.93	 brick04.bmp similarity : 0.93	 brick17.bmp similarity : 0.93
	 class01.bmp similarity : 1.00	 class20.bmp similarity : 0.89	 car15.bmp similarity : 0.83	 fount03.bmp similarity : 0.82	 class02.bmp similarity : 0.82
	 class13.bmp similarity : 0.82	 class10.bmp similarity : 0.82	 car04.bmp similarity : 0.81	 class19.bmp similarity : 0.81	 car10.bmp similarity : 0.81
	 Glas01.bmp similarity : 1.00	 Glas03.bmp similarity : 0.99	 Glas02.bmp similarity : 0.93	 Glas06.bmp similarity : 0.93	 Glas07.bmp similarity : 0.92
	 Glas08.bmp similarity : 0.90	 Glas14.bmp similarity : 0.83	 Glas13.bmp similarity : 0.77	 Glas20.bmp similarity : 0.75	 Glas04.bmp similarity : 0.70
	 Mask02.bmp similarity : 1.00	 Mask10.bmp similarity : 0.92	 Mask13.bmp similarity : 0.89	 Mask20.bmp similarity : 0.88	 Mask03.bmp similarity : 0.88
	 Mask16.bmp similarity : 0.88	 Mask04.bmp similarity : 0.87	 Mask09.bmp similarity : 0.87	 car20.bmp similarity : 0.87	 Mask11.bmp similarity : 0.84
	 hamm12.bmp similarity : 1.00	 hamm13.bmp similarity : 1.00	 hamm08.bmp similarity : 0.99	 hamm03.bmp similarity : 0.97	 hamm04.bmp similarity : 0.95
	 hamm05.bmp similarity : 0.95	 hamm02.bmp similarity : 0.94	 hamm10.bmp similarity : 0.94	 hamm20.bmp similarity : 0.94	 key05.bmp similarity : 0.80

기 시작점이 다른 모양으로부터 최적의 유사도를 찾아 낼 시작 골극을 추출한다. 이를 통해 LCS의 개수가 서로 8,9개로 다르지만 질의 이미지의 첫 번째 CS와 두 번째 결과 이미지의 5번째 CS를 시작으로 총 8개의 CS를 매칭한 결과가 최적의 유사도를 낸다는 것을 찾아낸다. 이 때, LCSS의 slist 평균이 0.98이 되고 alist 평균이 0.95가 된다. 질의 이미지는 Qr=1로 전체 골극이 매칭에 참여하며 Dr=0.95로 한 개의 오목 세그먼트가 매칭에서 제외되는 것을 알 수 있다.

검색의 비용적인 측면을 분석하기 위해 대표적으로 사용하는 두 가지 모양 특징 방법과 비교해 본다. [17]에서 그리드 기반 모양 특징 방법과 Fourier 서술자 기반 방법을 저자들이 제시하는 TPVAS 방법과 비교하였다.

표 3 모양 특징 방법별 비용 분석

기준	추출 비용
Grid 기반	major 축 찾는 비용 : $O(N^2)$ minor 축 찾는 비용 : $O(N)$ 회전과 스케일 : $O(N)$ 이진 숫자 생성 : $O(N^3)$
Fourier Descriptors 기반	중심점 찾는 비용 : $O(N \log N)$ 모든 반지름과 FD 계수 : $O(r^2)$ 시그너처 생성 : $O(r)$
TPVAS	MBC를 찾는 비용 : $O(N)$ 각도 시퀀스 계산과 그 시퀀스의 모든 DFT 상관계수 계산 비용 : $O(N)$
Ours	우세점 추출 비용 : $O(N \log N)$ LCSS 추출 비용 : $O(n)$ 형태 그래프 추출 비용 : $O(\text{slogs})$

N은 객체를 표현하는 좌표의 수이다. 퓨리에 기법의 r은 사용하는 반지름의 수이다. 본 논문에서 사용하는 n은 우세점의 수이며 s는 논리적 골극 세그먼트의 수이다. 앞서 말한 바와 같이 실험을 통해 평균적으로 나온 s는 우세점의 1/3 정도이다. 검색시 비용은 세그먼트 패턴 매트릭스 생성 비용이  $O(s^2)$ 이고 후보 골극 CCS로부터 최적의 유사 골극 시퀀스를 찾는 비용이  $O(s^2)$ 이다.

## 6. 결론

본 논문에서는 모양의 윤곽선과 영역 특징을 분석하고 이를 기반으로 설계된 구조적인 특성을 모양 특징으로 사용한다. 모양의 외부 골극이 모양을 판별하는데에 인지적으로는 가장 크게 작용하지만 전체적인 모양의 내부 형태가 다른 모양을 만들 수도 있다는 점을 고려하였다. 이러한 점이 골극 기반 매칭에서 문제로 제기되었던 부분이고, 골극선을 기반으로 하는 형태 그래프에서는 장점으로 강조하던 부분이다. 본 논문에서는 모양을 표현하는 특징으로 모양의 윤곽선을 이용한 외부 골

극 세그먼트 시퀀스와 내부 영역 정보를 반영하는 형태 그래프를 제시하였다. 형태 그래프를 통해 전역적인 분류를 수행하여 검색 공간을 축소하고 외부 골극 매칭을 통한 False hit을 줄일 수 있다. 본 논문에서 제시한 형태 그래프는 골극을 기반으로 한 것으로 외부 골극을 추출하는 방법과 동일한 방법으로 내부 영역의 형태 정보를 추출한다. 또한 우세점만을 이용하여 재귀적으로 추출되므로 기존의 형태 그래프에서 문제로 제기되던 과도한 비용 계산의 문제를 해결하였다. 우세점으로 표현된 골극의 불록 다각형 성질을 이용하여 골극 그래프와 유사한 형태 그래프를 나타내는 방법을 제시하였다. 본 논문에 적용한 일반화된 형태 그래프의 특징 값을 적용한 단계적인 클러스터링 방법은 키 값의 분류에 의한 클러스터링이다. 이를 키 값의 상관관계에 의해 자동적으로 클러스터링하는 연구가 현재 진행 중이다.

논리적 특징으로 표현된 모양 특징은 일반적이고 객관적인 모양 표현이 가능하므로 어떠한 특정 도메인에서도 의미적 특징 표현을 위한 기본 특징으로 사용할 수 있다. 또한 본 논문에서는 모양의 종류를 2차원 폐곡선으로 한정하였으나 우세점 추출 알고리즘을 변경하면 다양한 종류의 모양(개곡선, 홀이 있는 도형 등)으로 확장하여 적용할 수 있다.

## 참고 문헌

- [1] A.Torsello and E.R.Hancock, "Computing approximation tree edit distance using relaxation labeling," *Pattern Recognition Letters*, 24:1089-1097, 2003.
- [2] Belongie, S.; Malik, J.; Puzicha, J. "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 Issue: 4, pp. 509-522, April, 2002.
- [3] Geiger, D.; Liu, T.-L.; Kohn, R.V., "Representation and self-similarity of shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume: 25 Issue: 1, pp. 86-99, 2003.
- [4] E. Milios and E. Petrakis, "Shape retrieval based on dynamic programming," *Proc. Trans.on Image*, Vol. 9, No. 1, pp. 141-146, 2000.
- [5] Sebastian, T.B.; Kimia, B.B, "Curves vs skeletons in object recognition," *International Conference on Image Processing*, Vol. 3, pp. 22-25, Oct. 2001.
- [6] E. Klassen and A. Srivastava and W. Mio and S. Joshi, "Analysis of planar shapes using geodesic paths on shape spaces," *IEEE Pattern Analysis and Machine Intelligence*, 2004.
- [7] R. Basri and L. Costa and D. Geiger and D. Jacobs, "Determining the Similarity of Deformable Shapes," *Vision Research*, 38:2365-2385, 1998.

- [8] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, Steven W. Zucker, "Shock Graphs and Shape Matching," *ICCV*, pp. 222-229, 1998.
- [9] Di Ruberto, C.; "Attributed skeletal graphs for shape modelling and matching," *International Conference on Image Analysis and Processing*, pp. 554-559, Sept. 17-19, 2003.
- [10] B. Kimia and A. Tannebaum and S. Zucker, "Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space," *Int'l Journal of Computer Vision*, Vol. 15, pp. 189-224, 1995.
- [11] Fudos, I.; Palios, L.; Pitoura, E.; "Geometric-similarity retrieval in large image bases," *International Conference on Data Engineering*, pp. 441-450, 2002.
- [12] Thomas B. Sebastian, Philip N. Klein, Benjamin B. Kimia, "Shock-based Indexing into Large Shape Databases," *ECCV* (3), pp. 731-746, 2002.
- [13] Bin Luo; Robles-Kelly, A.; Torsello, A.; Wilson, R.C.; Hancock, E.R. "Learning shape categories by clustering shock trees," *International Conference on Image Processing*, Vol. 2, pp. 672-675, 2001.
- [14] Tyng-Luh Liu, Davi Geiger, "Approximate Tree Matching and Shape Similarity," *IEEE Intl. Conference on Computer Vision*, Vol.1, pp. 456-462, 1999.
- [15] 조성택, 엄기현, "모양 기반 이미지 분류를 위한 최적의 우세점 추출," *한국정보과학회 논문지 : 데이터베이스 제30권 4호*, pp. 362-371, 2003.8.
- [16] Nanhyo Bang, Kyhyun Um, "Structural Analysis and Matching of Shape by Logical Property," *Joint IAPR International Workshops on Syntactical and Structural Pattern Recognition*, Lisbon, Portugal, (LNCS 3138), pp. 521-529, 2004.8.
- [17] Maytham Safar, Cyrus Shahabi, Xiaoming Sun, "Image Retrieval By Shape: A Comparative Study," *IEEE International Conference on Multi-media and Expo* (1), pp. 141-154, 2000.



엄 기 현

1975년 서울대학교 공과대학 응용수학과 공학사. 1977년 한국과학기술원 전산학과 이학석사. 1994년 서울대학교 대학원 컴퓨터공학과 공학박사. 1978년 3월~현재 동국대학교 컴퓨터멀티미디어 공학과 정교수. 2001년 3월~2003년 2월 동국대학교 정보산업대학 학장 역임. 1995년 3월~1999년 2월 동국대학교 정보관리 처장 역임. 2004년 1월~현재 한국 멀티미디어학회 자문위원. 1999년 4월~2005년 4월 Int. Conf. on Database Systems for Advanced Applications Steering Committee 위원. 1998년 8월~2000년 7월 한국 정보과학회 데이터베이스연구회 운영위원장. 관심분야는 멀티미디어 데이터베이스와 응용 시스템, 게임 시스템 디자인 및 응용 시스템 멀티미디어 정보 관리 시스템



방 난 효

1995년 안양대학교 전자계산학과 공학사  
1997년 동국대학교 대학원 컴퓨터멀티미디어공학과 공학석사. 2005년 동국대학교 대학원 컴퓨터멀티미디어공학과 공학박사. 2004년 11월~현재 유한대학 정보통신학과 연구교수. 관심분야는 멀티미디어

데이터 모델링과 검색