

모양 기반의 식물 잎 이미지 검색 시스템

남 윤 영[†] · 황 인 준^{††}

요 약

본 논문에서는 식물 잎 모양을 기반으로 이미지를 표현하고 검색하는 식물 잎 이미지 검색 시스템을 보인다. 보다 효과적인 잎의 모양 표현을 위하여, MPP(Minimum Perimeter Polygons) 알고리즘을 개선하였고, 처리 시간을 줄이기 위하여, NN(Nearest Neighbor) 검색을 개선한 동적 매칭 알고리즘을 제안하였다. 본 시스템은 사용자에게 질의 이미지를 업로드하는 인터페이스를 제공하거나 모양 특징에 기반한 질의를 생성하는 도구를 제공하고 유사도에 따른 이미지를 검색한다. 검색의 편의성을 위해, 웹상에서 잎 모양과 잎 차레를 스케치하여 손쉽게 질의할 수 있게 하였다. 실험에 서는, 한국에 자생하는 식물 이미지 데이터베이스를 구축하였으며, 질의를 통해 검색된 유사한 이미지의 개수를 기반으로 성능을 평가하였다.

키워드 : 이미지 검색, 내용기반검색, 모양기반 검색, 이미지 인덱싱, QBE, QBS

Shape-Based Leaf Image Retrieval System

Yunyoung Nam[†] · Eenjun Hwang^{††}

ABSTRACT

In this paper, we present a leaf image retrieval system that represents and retrieves leaf images based on their shape. For more effective representation of leaf images, we improved an existing MPP algorithm. Also, in order to reduce the response time, we proposed a new dynamic matching algorithm that basically revises the Nearest Neighbor search. The system provides users with an interface for uploading query images or tools to generate queries based on shape features and retrieves images based on their similarity. For convenience, users are allowed to easily query images by sketching leaf shape or leaf arrangement on the web. In the experiment, we constructed an image database of Korean native plants and measured the system performance by counting the number of similar images retrieved for queries.

Key Words : Image Retrieval, Content-based Retrieval, Shape-based Retrieval, Image Indexing, QBE, QBS

1. 서 론

그 동안 이미지의 색상이나 모양, 질감(texture) 등을 이용한 내용기반의 이미지 검색에 관한 연구가 활발하게 진행되어 왔다. 특히, 모양의 외곽선(contour)이나 우세점(dominant point)을 이용한 검색은 얼굴이나 홍채, 지문의 인식 등 다양한 분야에서 사용되고 있다. 이러한, 이미지의 검색에서 색상이 비슷하거나 색상 정보의 양이 많아서 복잡한 경우, 색상보다 모양 특징을 이용하는 방법이 더욱 효과적일 수 있다.

모양을 표현하는 방법은 추출하는 부분에 따라 외부 윤곽선을 이용하는 방법[1]과 내부 골격(skeleton)을 이용하는 방법[2]으로 나눌 수 있다. 외부 윤곽선은 곡률(curvature), 중심거리, 복소수 좌표 함수 등을 이용하여 추출하며[3, 4], 푸리에 변환(Fourier transform)[5]을 거쳐 특징으로 표현된다. 내부 골격

은 평면 영역의 구조적 형태를 그래프로 변환시켜 만든다.

지금까지 다양한 표현 기법들이 연구되어 왔으며 이들은 공통적으로 다음과 같은 성질을 만족하고 있다.

첫째, 모양을 표현하는 기법에는 그래프[6, 7] 또는 트리[8]를 이용하여 구조화하는 방법과 퍼지 또는 확률[9]을 이용하는 방법, 신경망(neural network)[10]을 이용하는 방법, 지역(local)과 전역(global)으로 나누어 처리하는 방법[6, 11] 등 여러 가지가 있으며, 어떠한 방법을 사용하더라도 표현 방법에 의한 결과는 모양에 대해서 유일해야 한다. 즉, 다른 모양에 대한 표현은 서로 달라야 한다.

둘째, 이미지에 포함될 수 있는 왜곡이나 노이즈가 표현에 영향을 주어서는 안 되며, 단순화 과정에서 무시될 수 있어야 한다.

셋째, 이미지의 이동(translation), 조정(scaling), 회전(rotation), 보는 각도(view angle)의 변환, 좌우 또는 위아래 대칭(symmetric transformation)에 대해서 표현 값은 변함없어야 한다(invariance).

넷째, 위에서 조정 가능한 조건들이 변하더라도 유사한 모양을 검색할 수 있어야 한다. 즉, 비슷한 모양을 인식할

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업과 과학기술부 국책연구개발 사업인 유전자원지원 활용사업단의 연구비(no. BDM0100211)의 지원에 의해 수행되었습니다.

† 줄 회 원 : 아주대학교 정보통신전문대학원 박사과정

†† 종 신 회 원 : 고려대학교 전자컴퓨터공학과 조교수

논문접수 : 2005년 4월 8일, 심사완료 : 2005년 12월 6일

수 있도록 다양한 레벨의 정보를 가지고 있어야 한다 (scalability). 예를 들면, 보는 각도와 크기가 변할 것을 예상하여 여러 가지 각도와 크기가 변경했을 때의 값들도 고려해야 한다.

다섯째, 모양을 표현하는 알고리즘은 계산 비용면에서 효율적이어야 한다. 즉, 인덱싱과 매칭 알고리즘의 시간복잡도 T 는 $T \leq O(n \log n)$ 을 만족해야 한다.

여섯째, 매칭은 실시간으로 이루어져야 한다. 표현방법과 매칭방법이 실시간으로 이루어지지 않으면 검색 결과에 대해서 보장할 수 없다.

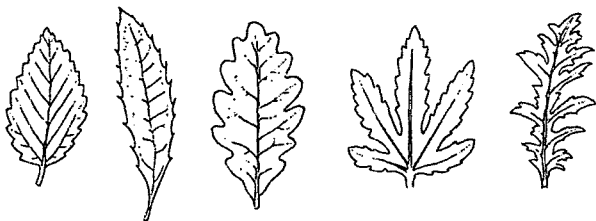
일곱째, 닫힌 곡선(closed curve) 뿐만 아니라 열린 곡선(open curve)에 대해서도 일관성 있는 표현 기법이 사용되어야 한다. 닫힌 곡선은 곡선 $A(\text{Shape } A = \{a_1, a_2, \dots, a_n\})$ 가 세그먼트 $a_i(a_i = [l_i, l_{i+1}])$ 로 이루어진 집합이라 할 때 아래의 식 (1)에서처럼 시작점과 끝점이 동일한 곡선이다.

$$l_n = l_{i+1} \tag{1}$$

이와 반대로 열린 곡선은 아래의 식 (2)에서처럼 동일하지 않은 곡선이다.

$$l_n \neq l_{i+1} \tag{2}$$

본 논문에서는 한국에 자생하는 식물의 잎에 대하여 이러한 조건을 만족하는 검색 시스템을 구현하였다. (그림 1)에서 보는 바와 같이 식물 잎의 경우, 색상이 대부분 초록색이고, 사진으로는 잎의 거칠고 매끈함을 알 수가 없기 때문에, 관찰에 의한 동정(identification)이나 식물명의 결정시, 식물의 색상이나 질감보다는 잎의 모양을 이용하는 것이 더 효과적이다.



(그림 1) 검색에 사용된 식물의 잎

본 논문의 구성은 다음과 같다. 2장에서는 내용기반 검색과 이미지 표현(representation)에 대해서 알아보고, 3장에서는 시스템을 구조와 구현에 사용된 이미지의 인덱싱과 매칭하는 기법에 대해서 설명한다. 4장에서는 구현 환경을 설명하고 성능을 평가한다. 마지막으로 5장에서는 결론과 향후 계획에 대해서 논의한다.

2. 관련 연구

2.1 내용기반 이미지 검색(CBIR)

이미지 검색 기법에는 특징 기반 검색, 유사 검색, 공간

관계 검색, 의미 검색으로 나누어 볼 수 있다. 특징 기반 검색은 이미지의 색상, 질감, 모양, 윤곽선과 같은 이미지의 특성에 기반을 둔 검색 방법으로써 이러한 정보는 자동 및 수동으로 생성될 수 있으며 데이터베이스에 이미지 데이터를 저장할 때 함께 저장되어 검색 시 이용된다. 유사 검색은 사용자가 예제 이미지를 주거나 개략적인 정보를 주면 그와 가장 유사한 이미지 데이터를 데이터베이스에서 검색하는 기법이다. 이 검색은 질의에 정확히 일치하는 하나의 결과를 찾아주는 것이 아니라 질의의 유사 범위 내에 속하는 결과 집합을 찾아주게 된다. 공간 관계 검색은 이미지 내에 존재하는 객체간의 공간 관계를 이용한다. 의미 검색은 이미지에 존재하는 객체에 의미 레이블을 부여해서 검색 시 이를 이용하여 결과를 찾아주는 기법이다.

2.2 윤곽선 검색(Edge Detection)

이미지의 형태를 구조화된 표현으로 바꾸기 위해서는 윤곽선 검색이 선행되어야 하며, 지금까지 여러 가지 윤곽선 검색 기법이 제안되었다.

ACM(Active Contour Model)[12, 13]은 검색이 진행되는 형태가 뱀이 움직이는 모양과 유사하여 스네이크 모델(Snake Model)이라 불리며 스네이크 에너지 함수를 사용한다. 하지만 복잡한 윤곽선을 가진 이미지의 경우, 정확한 형태를 검색하기 어렵다. 소벨(Sobel Operation)[14]은 네 개의 마스크(mask)를 사용하며, 픽셀에 대한 gray-level 함수 값의 증감을 이용한다. 그러나 이미지의 전체를 반복(가로*세로*3²)해서 수행해야 하는 단점이 있다. 이러한 문제 때문에 쉽고 효과적인 기법들이 제안되었으며, 그 중에 하나가 Prompt Edge Detection[15] 이다. 이 기법은 픽셀(x,y)에 대한 이웃(neighbor) 픽셀의 밝기 값(g(x,y) : gray value)을 이용한다. 즉, 하나의 픽셀에는 여덟 방향으로의 이웃 픽셀이 존재하며, 각 방향(d)에 대한 이웃픽셀의 차이 값(difference)은 아래의 식 3에 의해 계산한다. 식에서 차이 값이 임계값(threshold) 이상이면 윤곽선의 점으로 간주한다.

$$difference_d(x,y) = |g(x,y) - g_d(x,y)| \tag{3}$$

본 논문에서는 가우시안 마스크를 이용하여 노이즈를 제거한 후, 소벨 마스크와 같은 윤곽선 검출 마스크를 수행하는 캐니 윤곽선 검출(Canny Edge Detection) 알고리즘[16]을 사용하였다.

2.3 모양 표현(Shape Representation)

윤곽선의 모든 점을 검색에 이용하는 것은 비용면에서 비효율적이기 때문에 특징으로 이용할 수 있는 점들로 간결화하는 기법을 사용한다. 이러한 기법중의 하나인 체인코드(chain code)[17]는 시작점에서의 길이와 방향을 이용하는데, 잡음에 민감하고 임의의 각도에 대해서는 불가능하다. 퓨리에 변환은 윤곽선 기반의 검색방법 중 많이 사용되며, 퓨리에 변환을 통해 생성된 복소수 형태의 상관계수들을 모양

기술자(descriptor)로 이용한다.

또 다른 모양 표현방법으로 CCD(Centroid Contour Distance)[18]가 있다. CCD는 중심점에서 n개의 방향으로 일정한 각도($\Delta\theta, \Delta\theta = 2\pi/n$)로 나누어 중심거리를 계산하는 방법이지만, 불룩한 외곽선이 많은 경우 외곽선의 한 점에 대해서 거리가 여러 가지로 계산되는 문제점이 있다.

MPEG-7에서는 외곽선 기반의 모양 표현방법으로 CSS(Curvature Scale Space)[19]를 제안하고 있다. CSS는 외곽선을 1D 신호로 처리해서 분석하는 방법이며, 제로 크로싱(zero-crossing)을 조사하여 모양의 외곽선에서 오목점과 볼록점을 찾아내며, Scale Normalization, Curvature Derivation, CSS Map, CSS Peaks, Peak Normalization 단계를 거쳐 CSSD를 계산한다. 이 방법은 이미지 윤곽선 스펙트럼을 비교하는 많은 연산 과정을 필요하기 때문에 검색속도가 낮으며, 이미지들의 개수가 증가함에 따라 계산량이 기하급수적으로 증가하게 되는 단점이 있다.

다각형 근사(polygonal approximation)[20]는 닫힌 곡선에 대해서 객체 모양의 지역 특징인 고정된 차수의 우세점들을 추출하여 인접한 우세점 간의 각도, 거리, 좌표들을 산출하는 방법이다. 이 방법은 객체의 모양을 정확하게 표현하고 신속하게 찾기 위한 우세점을 추출하는 알고리즘이 중요하며, 본 연구에서는 MPP(Minimum Perimeter Polygons) 알고리즘[21, 22]을 사용하였다. 또한, 영역기반의 표현 방법인 골격화(skeletonizing)[2]는 평면 영역의 구조화된 모양을 그래프로 변환시킨다. 이러한 골격화는 MAT(medial axis transformation)[23]을 통해 만들 수 있으며, 영역(region) R과 경계(border) b가 있을 때, R안의 각 점 p에 대해서 b의 가장 가까운 이웃을 찾는다. 만약 p가 하나 이상의 이웃을 갖게 되면, 이것을 중앙축(medial axis)이라 하며 골격(skeleton)이라고 한다.

2.4 모양 매칭(Shape Matching)

사용자가 질의한 이미지(U)와 데이터베이스의 이미지(V)의 우세점을 추출한 후, 얼마나 일치하는가에 대한 유사도(similarity)를 측정하게 된다. 유사도를 측정하는 방법 중에 가장 널리 사용하는 방법은 아래의 식 4인 Lp Distance(Minkowski Distance)이다[24]. 이 식에서 p값이 1이면 맨하탄 거리(Manhattan Distance)이고, p값이 2이면 유클리드 거리(Euclidean distance)이며, $p \rightarrow \infty$ 이면 Max Matrix가 된다.

$$D(U, V) = \left(\sum_{i=1}^k |u_i - v_i|^p \right)^{1/p}$$

※ u_i, v_i : feature vectors, k : the number of dimension

(4)

이외에도 최소 가중치 거리와 최소 편차 거리를 이용하고 1:1 대응 함수에서만 사용되는 Bottleneck Distance [25] 함수가 있으며, 1:1 대응이 존재하지 않은 다른 크기의 두 점 집합에 사용되는 Hausdorff Distance[26]가 있다.

본 논문에서는 함수식 5인 유클리드 거리를 사용한다.

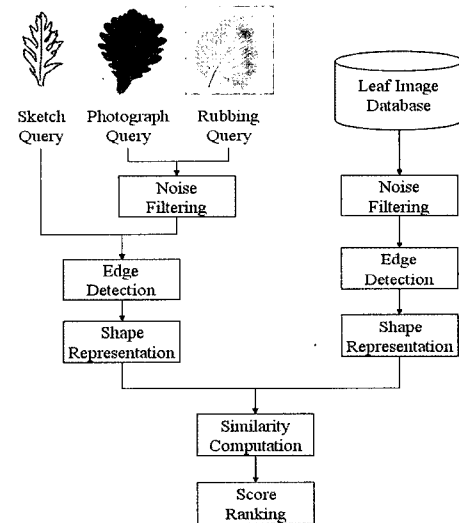
$$D(U, V) = \sqrt{\sum_{i=1}^k (u_i - v_i)^2}$$

(5)

3. 모양 기반의 잎 이미지 검색시스템 설계

3.1 시스템 구조

(그림 2)는 전체적인 시스템의 프로세스 흐름도를 보이고 있다. 프로세스는 크게 질의를 입력받아 우세점을 추출하는 부분과 잎 이미지 데이터베이스의 이미지를 분석하여 우세점을 추출하는 부분으로 나눌 수 있다. 우선, 질의는 스케치, 사진, 탁본 이미지를 입력받을 수 있다. 스케치를 제외한 사진과 탁본 이미지에는 노이즈가 있기 때문에 필터링의 단계를 거친다. 노이즈 필터링 프로세스를 통해 불필요한 점들은 제외되고 윤곽선 검출을 통해 모양을 표현할 수 있는 선들을 추출한다. 추출한 선들은 내부 선은 제외한 외곽선을 대상으로 모양 표현기법을 통해 우세점을 찾는다. 반면, 저장된 이미지도 노이즈 필터링, 윤곽선 검출, 모양 표현을 거쳐 우세점을 추출한다. 마지막으로, 질의한 이미지의 우세점과 이미지 데이터베이스의 우세점에 대한 각각의 유사도를 계산하여 가장 유사한 순서로 결과를 보여준다.



(그림 2) 프로세스 흐름도

3.2 이미지 인덱싱

이미지 검색과 추출에서 이미지를 표현하고 인덱싱하는 단계는 검색 결과의 정확성을 결정하는 부분이다. 본 절에서는 이미지의 모양을 표현하는 방법과 인덱싱하는 방법에 대해서 서술한다.

3.2.1 MPP 알고리즘

이미지에서 MPP를 찾기 위해서는 우선 그 이미지를 셀(cell)로 쪼개어 좀 더 단순화된 형태의 매트릭스(matrix)로 만든다. 이렇게 단순화된 매트릭스에서 특정한 각도 이상의

점들을 찾아서 볼록점(convex)과 오목점(concave)들을 찾은 후, 이 점들을 순회(travel)하면서 점들의 오목하고 볼록한 특징을 통해 이미지의 형태를 단순화한다. 이때, 이미지의 형태를 결정하는데 중요하지 않은 점들은 제거하며, 단순화된 형태의 매트릭스에서 (X,Y)좌표의 집합이 바로 MPP가 된다.

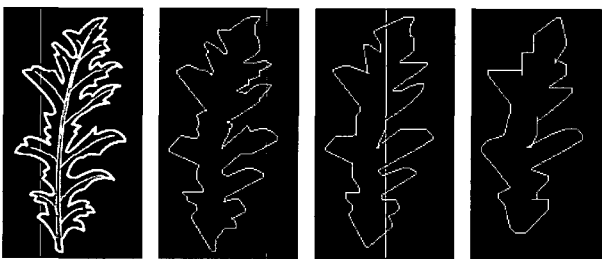
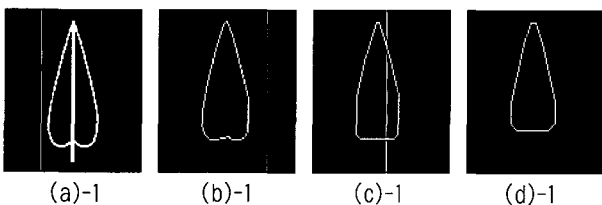
식물의 잎 모양 이미지에서, 식물의 잎은 잎맥을 제외한 잎의 외곽선이 시작점과 끝점이 서로 연결되어있는 폐쇄형(closed-loop)이므로, 식물의 잎 모양 이미지에 대해 MPP 알고리즘을 적용할 수 있다.

MPP 알고리즘은 MATLAB[27]으로 구현하였으며, 아래 <표 1>의 작업 순서대로 점들의 시퀀스(sequence)를 찾는다.

<표 1> MPP에 의해 시퀀스를 추출하는 단계

1. 셀로 쪼개어진 간단한 형태의 매트릭스로 변환한다.
2. 1에서 구한 매트릭스에서 폐쇄형 내부 영역을 구한다.
3. 내부 영역의 경계를 구한다. 이것은 4-connected sequence의 형태이다.
4. 경계에서 프리먼 체인 코드(Freeman Chain Code)를 구한다.
5. 체인코드에서 볼록점(convex)과 오목점(concave)을 구한다.
6. 볼록점들을 연결하여 초기 다각형을 구하고, 다각형의 외부에 있는 오목점은 모두 제거한다.
7. 제거 후 남은 볼록점과 오목점을 연결하여 다각형을 만든다.
8. 연결된 다각형에서 볼록점 중에 실제로는 오목점들을 제거한다.
9. 7번과 8번을 더 이상의 변화가 없을 때 까지 반복하며 점의 각도가 180도인 것들도 제거한다.
10. 최종적으로 남은 점들이 MPP의 시퀀스이다.

이러한 MPP 알고리즘을 식물의 잎 이미지에 적용하면 (그림 3)과 같은 이미지를 얻을 수 있다. 식물의 잎 이미지에 MPP 알고리즘을 적용한 예는 (그림 3)과 같다.



(그림 3) 식물의 잎에 적용한 MPP 알고리즘의 예
(a) 원본이미지 (b) Cell Size=2
(c) Cell Size=3 (d) Cell size=5

3.3.2 우세점 추출

MPP 알고리즘은 점과 점 사이를 연결하는 선을 기준으로 볼록한 점(180° 이하)과 오목한 점(180° 이상)으로만 점

<표 2> 불필요한 점들을 병합하는 알고리즘

```

Input
point : 점들의 (X, Y)좌표
N : 점들의 개수
threshold : 불필요한 점들의 기준 각도

find_sequence(point, N, threshold)
{
  for ( i=0; i<N; i++)
  {
    a=get_distance(point[i-1], point[i+1])
    b=get_distance(point[i], point[i-1])
    c=get_distance(point[i], point[i+1])
    angle=acos( ( b^2 + c^2 - a^2 ) /
                ( 2*b*c ) )
    if(angle < threshold)
      add_point(result, point[i])
  }
  return result
}
    
```

들을 찾기 때문에 추출한 점들 중에는 불필요한 점들이 많이 포함되어 있다. 이러한 점들을 간결화하기 위해 각 점들의 각도를 계산하여 임계값의 범위를 벗어나는 점들은 병합하였다. 아래의 <표 2>는 이러한 점들의 병합 여부를 판단하는 알고리즘이다.

알고리즘에서 $\triangle ABC$ 의 세 점 A, B, C 의 각도를 도출하기 위해 (식 6)과 같이 코사인 제 2법칙(the second law of cosines)을 사용하였다.

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc} \tag{6}$$

$$(a = \overline{BC}, b = \overline{AC}, c = \overline{AB})$$

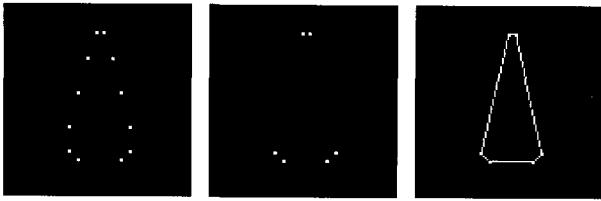
점 B 의 좌표(x_i, y_i)와 점 C 의 좌표를 (x_j, y_j)에 대해 \overline{BC} 의 길이 a 는 식 7에 의해 구할 수 있다.

$$a = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{7}$$

<표 3>과 (그림 4)는 임계값이 160°일 때, (그림 3) (d-1)의 MPP에서 불필요한 점들을 병합하여 추출한 결과를 보여주고 있다.

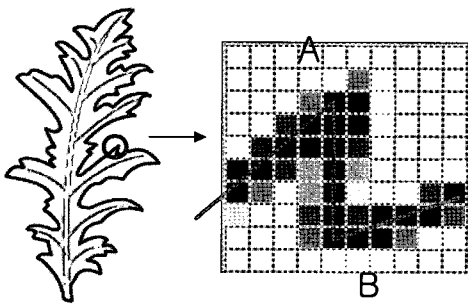
<표 3> 개선된 MPP 알고리즘을 이용한 예제

i	X[i]	Y[i]	각도(°)	병합여부
0	51	16	108	
1	55	16	108	
2	60	31	176	병합
3	65	51	180	병합
4	70	71	166	병합
5	70	85	135	
6	65	90	135	
7	41	90	135	
8	36	85	135	
9	36	71	166	병합
10	41	51	180	병합
11	46	31	176	병합



(병합 전) (병합 후) (점 연결)
 (그림 4) 개선된 MPP 알고리즘을 통해 추출한 시퀀스

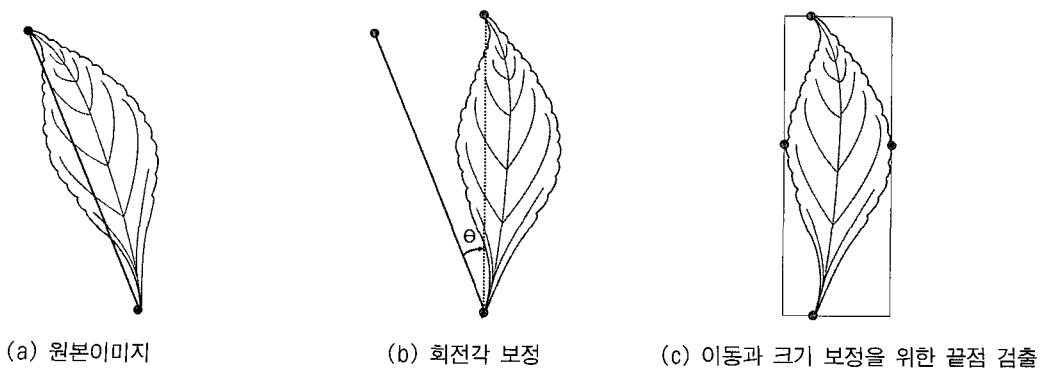
(그림 5)는 잎의 외곽선에서 두 개의 우세점 A, B를 찾은 예를 보이고 있다.



(그림 5) 외곽선에서의 우세점 추출

3.2.3 입력값 보정 (invariance)

입력되는 이미지들은 이동, 크기, 회전에 대해서 값이 불변해야 한다. 예를 들어, 잎이 세로인 경우와 가로인 경우, 모양에 대한 표현 값은 동일해야 한다. 이를 위해, 우세점들 간의 거리에서 가장 최대거리에 있는 점 2개를 주맥(主脈, main vein)의 양 끝점으로 간주하고, 두 개의 점을 이은 선을 기준으로 각도를 보정한다. (그림 6)에서처럼 원본이미지 (a)에서 가장 거리가 먼 2개의 점을 이은 선을 기준으로 하여, (b)에서 보는 바와 같이 시계방향으로 각도 θ 만큼 회전을 시켜 각도를 보정한다. (c)에서는 4 방향으로 가장 끝점을 검출한 4개의 점으로 사각형을 구하여 사각형의 중심점을 통해 이미지 이동값을 보정하고, 가로, 세로 길이의 비율을 조절하여 크기값을 보정한다.



(그림 6) 입력값 보정

3.3 이미지 매칭과 추출

이미지 표현 알고리즘을 통해 시퀀스를 추출하여 이미지 인덱싱이 이루어지면 사용자가 질의한 이미지가 데이터베이스의 이미지와 얼마나 일치하는지를 계산하게 된다. 이러한 유사도 계산을 통하여 결과값을 얻어낸 후, 가장 비슷한 이미지부터 순서대로 보여준다.

유사도를 계산하기 위해 2.4절의 (식 5)의 유클리드 거리를 사용하였으며, 아래의 (식 8)과 같이 질의한 이미지의 점들(u)과 데이터베이스에 저장된 이미지의 점들(v)과의 거리에서 최소값을 가지는 거리(D_i)를 이용하여 계산하였다.

$$S(U, V) = \frac{1}{|u|} \sum_{i=1}^{|u|} \min(D_i(u, v)) \quad (8)$$

여기서 $|u|$ 는 질의한 이미지의 점들의 개수이고, 함수 D_i 의 값은 u_i 에 대한 v_i 의 거리로써 열거형의 형태를 지닌다.

3.3.1 동적인 매칭 알고리즘

질의한 이미지의 점들과 이미지 데이터베이스의 점들에서 거리의 최소값을 찾는 시간 복잡도는 $O(|u| |v|)$ 이므로 $O(n^2)$ 이다. 이러한 brute-force 방식보다 복잡도를 줄이기 위해 ϵ -nearestneighbor (ϵ -NN) 검색을 사용한다. 이 방법을 사용하면, $O(D \text{polylog}(N))$ 의 시간 복잡도를 갖는다[28]. 검색시간을 최소화하기 위해 동적으로 매칭할 수 있도록 하였다. 우선 식물의 잎이 좌우대칭이라는 특징을 이용하여 우세점들(n) 중에 가장 긴 거리를 갖는 직선(축)을 기준으로 한쪽 면의 점들($\frac{n}{2}$)만을 이용한다. 예를 들어 MPP를 통해 검출한 우세점이 200개이면 가장 긴거리인 선을 기준으로 100개의 점들만 매칭에 이용된다.

또한, 유사도 계산시에 유사도가 특정 임계값을 초과하면 더 이상 매칭 프로세스를 진행하지 않고 다음 이미지의 유사도를 계산한다. 예를 들어 임계값이 200(pixel)이고 우세점 100개중 50번째 점까지의 유사도가 200(pixel)이 넘었을 경우에 매칭을 중단하고 다음 이미지와의 매칭을 시작한다.

식물의 잎 중에는 잎의 외곽선이 복잡한 경우가 있다. 이러한 경우, 즉, 점의 개수가 임계 값보다 많으면 샘플링함수(SMP())를 통해 n 에서 m 개만큼 추출한다. 예를 들면, 질의한 이미지의 우세점의 개수가 $|u|$, 매칭 대상이미지의 우세점 개수가 $|v|$ 일 때, $|u| < |v|$ 이면 $SMP(v)$ 를 통해 샘플링한 우세점을 추출하여 유사도를 계산한다. <표 4>는 이러한 동적인 매칭 알고리즘을 보이고 있다.

<표 4> 동적인 매칭 알고리즘

```

Input
input_image : 사용자가 입력한 이미지의 우세점들 (X, Y)좌표
db_image : 데이터베이스에 입력되어 있는 이미지의 우세점들 (X, Y)좌표
N : 사용자가 입력한 이미지의 우세점들 개수
threshold : 유사도를 수용할 수 있는 최대 임계값

dynamic_matching(input_image, db_image, N, threshold)
{
    input_point=condensing_point(input_image)
    db_point=condensing_point(db_image)
    if(sizeof(input_point) < sizeof(db_point))
        SMP(db_point)

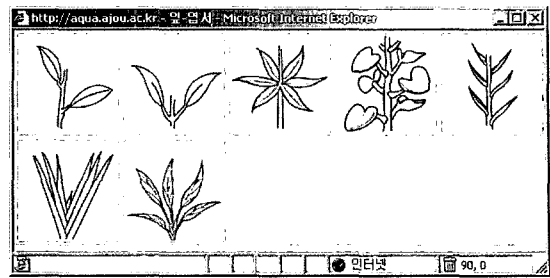
    for ( i=0; i<N/2; i++)
    {
        NN_point=NN_search(input_point[i], db_point)
        Sim = S(input_point[i], NN_point, N/2)
        if(Sim > threshold) {
            Sim = -1
            break;
        }
    }
    return result
}
    
```

4. 시스템의 구현 및 평가

본 절에서는 식물의 잎을 검색하기 위한 웹 기반의 인터페이스를 보이고 여러 가지 실험결과를 기술하고 평가한다.

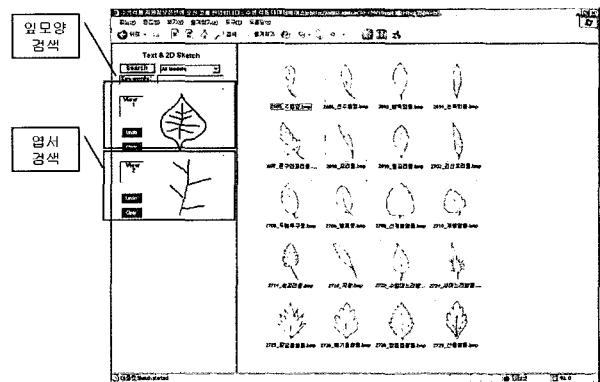
4.1 구현 환경

본 논문에서 제안하는 알고리즘의 성능을 분석하기 위해서 이미지 기반의 검색 시스템을 개발하였다. 검색 시스템은 Xeon CPU 2.8GHz, 1GB RAM의 PC에서 실험하였으며, 데이터베이스는 Microsoft SQL Server 2000을 사용하였다. 스케치 입력은 애플릿으로 구현하였으며 잎 모양과 엽서(葉序, leaf arrangement)로써 검색할 수 있다. 엽서는 잎차레를 뜻하며 아래의 (그림 7)과 같이 어긋나기, 마주나기, 돌려나기, 십자대생, 2방향호생, 과상, 로제트형으로 분류할 수 있다. 엽서 검색에서 사용자가 잎차레를 스케치하면 중심 줄기에서 자라는 잎 들간의 간격과 방향으로 엽서를 인식하게 된다. 그러나 그림에서 보는 바와 같이 어긋나기, 십자대생, 2방향호생을 구분하는 것은 어려울 뿐만 아니라, 입력된 사용자의 2차원 스케치만으로는 인식하기 어렵다. 따라서, 십자대생과 2방향호생은 어긋나기로 포함시켜서 네 가지 엽서(어긋나기, 마주나기, 과상, 로제트형)로만 구분하여 인식하도록 하였다.



(그림 7) 엽서(어긋나기, 마주나기, 돌려나기, 십자대생, 2방향호생, 과상, 로제트형)

(그림 8)은 사용자의 검색 인터페이스이다. 검색은 잎 모양과 엽서를 입력하여 검색할 수 있다. 잎 모양은 3장에서 서술한 기법을 통해 검색을 하며, 엽서는 중심 줄기에서 뻗어나간 잎들의 순서와 방향을 추출하여 위의 (그림 7)에서의 엽서 중에 하나로 분류되어 검색에 이용한다.



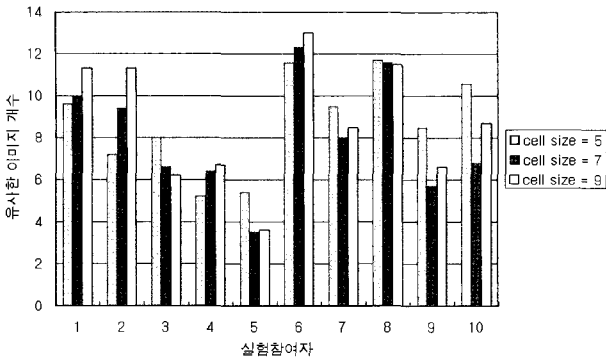
(그림 8) 검색 인터페이스

4.2 성능 평가

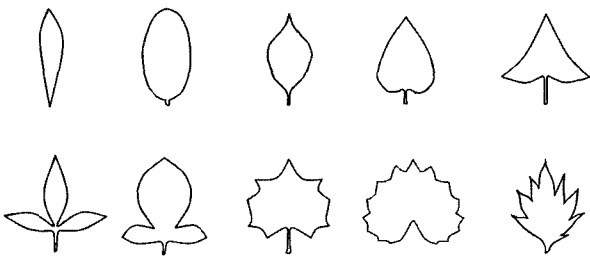
성능평가를 위해 국내에서 자생하고 있는 식물들을 수록하고 있는 대한식물도감[29]에서 1032종의 잎 이미지를 발취하여 500x500 픽셀의 크기로 스케치하여 사용하였다. 스케치 이미지가 정면이 아닌 0°~90°범위에서 보는 각도가 다르게 묘사되어 있어, 모든 이미지의 보는 각도를 0°로 통일시켜 재포맷하였다.

검색의 정확도를 측정하기 위해, 20대의 참가자 10명을 대상으로 실험하였다. 실험방법은 10명의 참가자에게 10번의 질의를 하고 셀 크기가 5, 7, 9에 대한 결과를 각각 보였다. 가장 유사한 이미지 순서대로 20개를 보였으며, 이 중에 비슷한 이미지의 개수를 측정하였다. (그림 9)는 실험 참여자와 셀 사이즈에 따른 추출 이미지중 유사한 이미지의 개수를 보여주고 있다.

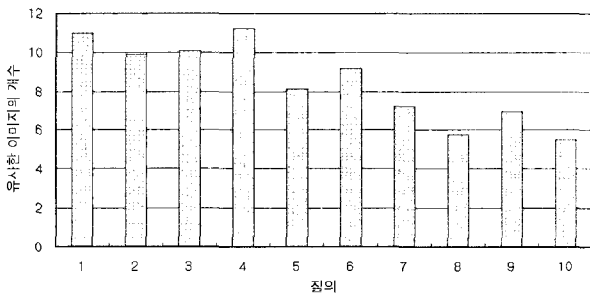
(그림 9)에서 볼 수 있듯이 실험 참여자의 성향에 따라 유사하다고 판단하는 기준이 다르며, 셀 사이즈에 따른 정확성의 차이는 크지 않다는 것을 알 수 있다. 위의 실험자 중에 가장 낮은 평가를 한 5번 실험자와 가장 좋은 평가를 한 6번 실험자를 제외한 8명의 실험자를 대상으로 질의에 따른 유사한 이미지의 개수를 평가하였다.



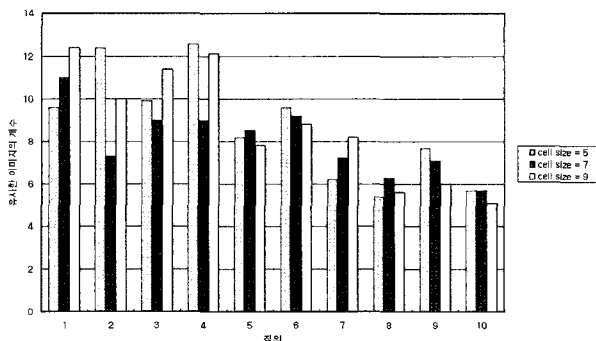
(그림 9) 실험참여자와 셀 사이즈에 따른 유사한 이미지 개수



(그림 10) 질의에 사용된 스케치 이미지



(그림 11) 질의에 따른 유사한 이미지의 개수



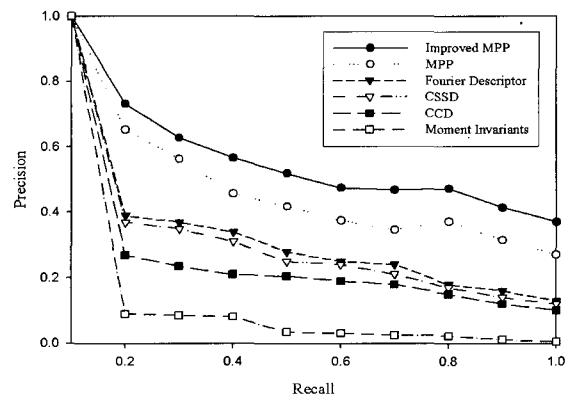
(그림 12) 질의와 셀 사이즈에 따른 유사한 이미지의 개수

(그림 10)은 질의에 사용된 이미지이며, (그림 11)은 질의에 따른 유사한 이미지의 개수이다. 그림에서 보듯이 질의 1은 단순한 모양의 질의였으며, 질의 10은 복잡한 모양의 질

의였다.

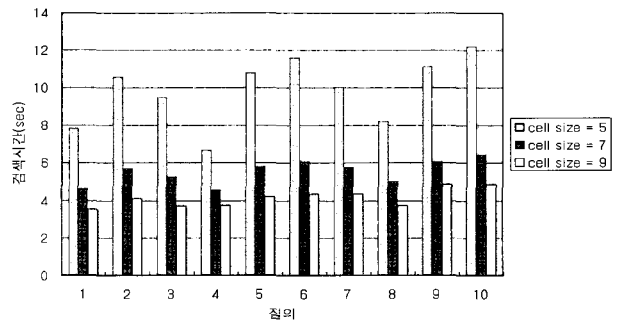
(그림 12)는 질의와 셀 사이즈에 따른 유사한 이미지의 개수를 보이고 있다. 셀 사이즈에 따른 유사한 이미지의 개수를 분석해 보면, 단순한 이미지의 경우 셀 사이즈가 클수록 더 많은 유사이미지를 찾고, 복잡한 이미지의 경우 셀 사이즈가 작을수록 검색의 정확률이 높아짐을 알 수 있다.

(그림 9), (그림 11), (그림 12)의 결과를 종합하여 recall과 precision에 대한 그래프를 (그림 13)으로 표현하였다. 비교에 사용된 표현기법으로는 논문에서 제안한 개선된 MPP, MPP, Fourier Descriptor, CSSD, CCD, Moment Invariants이다. 그림에서 알 수 있듯이 제안한 알고리즘이 다른 알고리즘의 비해 더 좋은 성능을 보이고 있다.



(그림 13) Precision과 Recall

(그림 14)는 질의와 셀 크기에 따른 검색 시간을 보여 주는데, 셀 사이즈에 따라 검색시간이 비례함을 알 수 있다.



(그림 14) 질의와 셀 사이즈별 검색시간

5. 결론 및 향후 연구 과제

본 연구에서는 식물의 잎을 검색하기 위하여 모양을 기반으로 한 이미지 검색 시스템을 구현하였다. 잎 모양을 표현하기 위해 개선된 MPP 알고리즘을 사용하여 이미지의 외곽선에서 우세점을 추출하였으며 검색을 다양화하고 정확도를 높이기 위해 잎차례를 인덱싱에 사용하였다. 성능평가에서 본 시스템의 개선된 알고리즘을 사용했을 때 기존의 방법보

다 더 좋은 성능을 얻을 수 있었다. 향후 연구 과제로는 좀 더 개선된 알고리즘을 개발하고, 잎 모양 뿐만 아니라 잎 내부의 잎맥을 이용하여 검색할 수 있도록 구현하는 것이다.

참 고 문 헌

[1] Ballard, D.H. and Brown, C.M. Computer Vision, Prentice-Hall, 1982.
 [2] Sundar, H., Silver, D., Gagvani, N., Dickinson, S., "Skeleton based shape matching and retrieval," Shape Modeling International, p.130, 2003.
 [3] Nishida, H., "Structural feature indexing for retrieval of partially visible shapes," Pattern Recognition, Vol.35, No.1, pp.55-67, 2002.
 [4] Loncaic, S., "A survey of shape analysis techniques," Pattern Recognition, Vol.31, No.8, pp.983-1001, 1998.
 [5] Chang, C., Wenyin, L. and Zhang, H., "Image Retrieval Based on Region Shape Similarity," 13th SPIE symposium on Electronic Imaging Storage and Retrieval for Image and Video Databases, 2001.
 [6] Han, M. H. and Jang, D., "The use of maximum curvature points for the recognition of partially occluded objects," Pattern. Recognition, Vol.23, pp.21-23, 1990.
 [7] Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., & Zucker, S.W., "Shock Graphs and Shape Matching," International Journal of Computer Vision, Vol.35, No.1, pp.13-32, 1999.
 [8] Gottschalk, P.G., Turney, J.L., Mudge, T.N., "Efficient Recognition of Partially Visible Objects Using a Logarithmic Complexity Matching Technique," The International Journal of Robotics Research, Vol.8, No.6, pp.110-131, 1989.
 [9] Bhanu B. and Faugeras O., "Shape matching of two dimensional objects," PAMI 6, pp.137-155, 1984.
 [10] Bebis, G., Papadourakis, G.M. and Orphanoudakis, S., "Curvature Scale Space Driven Object Recognition with an Indexing Scheme based on Artificial Neural Networks," Pattern Recognition, Vol.32, pp.1175-1201, 1999.
 [11] Petrakis, E., Diplaros, A. and Milios, E., "Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.11, pp.1501-1516, 2002.
 [12] Kass M, Witkin A, Terzopolous D., "Snakes: Active Contour Models. International Journal of Computer Vision," pp.321-331, 1988.
 [13] Choi, W., Lam K. and Siu, W., "An adaptive active contour model for highly irregular boundaries," Pattern Recognition, Vol.34, pp.323-331, 2001.
 [14] Gonzalez, Rafael C., Woods, Richard C., Digital Image Processing, Addison-Wesley, 1992.
 [15] Lin, H. J., Kao, Y. T., "A prompt contour detection method," International Conference On the Distributed Multimedia Systems, 2001.
 [16] Michael Heath, et al., "A Robust Visual Method for Assessing the Relative Performance of Edge Detection Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.12, pp.1338-1359, 1997.
 [17] Freeman, H., Saghri, J., "Comparative Analysis of Line Drawing Modelling Schemes," Computer Graphics and Image Processing, Vol. 12, 1980.
 [18] Zhiyong Wang, Zheru Chi, Dagan Feng, Qing Wang, "Leaf

Image Retrieval with Shape Features," Lecture Notes in Computer Science, Vol.1929, pp.477-487, 2000.
 [19] Mokhtarian F, Abbasi S, Kittler J. "Efficient and robust retrieval by shape content through curvature scale space," Int Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands, pp.35-42, 1996.
 [20] Kurozumi Y., Davis W.A., "Polygonal approximation by the minimax method," Computer Vision, Graphics and Image Processing, pp.248-264, 1982.
 [21] Sklansky, Chazin et al. "Minimum perimeter polygons of digitized silhouettes," 1972.
 [22] Sklansky J., "Finding the Convex Hull of a Simple Polygon," Pattern Recognition Letters, Vol.1 No.2, pp.79-84, 1982.
 [23] Lee, D.T., "Medial axis transformation of a planar shape," IEEE Transactions On Pattern Analysis and Machine Intelligence, pp.363-369, 1982.
 [24] Veltkamp, R., "Shape matching: similarity measures and algorithms," Technical Report UU-CS-2001-03, Utrecht University, the Netherlands, 2001.
 [25] Efrat, A. and Itai, A., "Improvements on bottleneck matching and related problems using geometry," The 12th Symposium on Computational Geometry, pp.301-310, 1996.
 [26] Alt, H., Behrends, B. and Blomer, J., "Approximate matching of polygonal shapes," Ann. Math. Artif. Intell., Vol.13, pp.251-266, 1995.
 [27] The MathWorks-MATLAB and Simulink for Technical Computing <http://www.mathworks.com>
 [28] Indyk, P., Motwani, R., "Approximate nearest neighbors: towards removing the curse of dimensionality," The 30 annual ACM symposium on Theory of computing, pp.604-613, 1998.
 [29] 이창복, 대한식물도감, 향문사, 서울, 1982.

남 윤 영



e-mail : youngman@ajou.ac.kr
 2001년 아주대학교 정보 및 컴퓨터공학과 (학사)
 2001년~2003년 아주대학교 정보통신전문대학원(석사)
 2003년~현재 아주대학교 정보통신전문대학원 박사과정

관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, 영상 처리, 패턴인식, 컴퓨터 비전

황 인 준



e-mail : ehwang04@korea.ac.kr
 1988년 서울대학교 컴퓨터공학과(학사)
 1990년 서울대학교 컴퓨터공학과(석사)
 1998년 Univ. of Maryland at College Park 전산학과(박사)
 1998년~1999년 Bowie State Univ., Assistant Professor
 1999년~1999년 Hughes Research Lab. 연구교수

1999년~2003년 아주대학교 정보통신전문대학원 조교수
 2003년~2004년 아주대학교 정보통신전문대학원 부교수
 2004년~현재 고려대학교 전자컴퓨터공학과 조교수
 관심분야 : 데이터베이스, 멀티미디어 시스템, 정보 통합, 전자상거래, 영상처리, 유비쿼터스 컴퓨팅