

모양기반 식물 잎 이미지 검색을 위한 표현 및 매칭 기법

(A Representation and Matching Method for Shape-based Leaf Image Retrieval)

남 윤 영 [†] 황 인 준 ^{**}
(Yunyoung Nam) (Eenjun Hwang)

요약 본 논문은 모양 특성을 이용한 효과적인 식물 잎 이미지 검색 시스템을 제시한다. 잎 이미지의 더 효과적인 표현을 위해 개선된 MPP 알고리즘을 제안하고, 매칭에 소요되는 시간을 줄이기 위해 기존의 Nearest Neighbor(NN) 검색을 수정한 동적인 매칭 알고리즘을 제시한다. 특히, 더 나은 정확율과 효율성을 위해, 잎 모양과 잎차레를 스케치하여 질의할 수 있도록 하였다. 실험에서는 제안한 알고리즘과 기존의 알고리즘인 CCD(Centroid Contour Distance), Fourier Descriptor, Curvature Scale Space Descriptor(CSSD), Moment Invariants, MPP와 비교하였다. 1000여개의 식물 잎 이미지를 통한 실험결과는 제안한 방법이 기존의 기법보다 더 좋은 성능임을 보였다.

키워드 : 이미지 검색, 내용기반 검색, 모양기반 검색, 이미지 인덱싱, QBE(Query By Example), QBS(Query By Sketch)

Abstract This paper presents an effective and robust leaf image retrieval system based on shape feature. Specifically, we propose an improved MPP algorithm for more effective representation of leaf images and show a new dynamic matching algorithm that basically revises the Nearest Neighbor search to reduce the matching time. In particular, both leaf shape and leaf arrangement can be sketched in the query for better accuracy and efficiency. In the experiment, we compare our proposed method with other methods including Centroid Contour Distance(CCD), Fourier Descriptor, Curvature Scale Space Descriptor(CSSD), Moment Invariants, and MPP. Experimental results on one thousand leaf images show that our approach achieves a better performance than other methods.

Key words : image retrieval, content-based retrieval, shape-based retrieval, image indexing, QBE(Query By Example), QBS(Query By Sketch)

1. 서 론

이미지에는 색상, 질감, 모양 등 다양한 정보를 담고 있다. 이러한 정보를 이용하여 색상 히스토그램을 만들거나 객체를 추출하여 객체들 간의 공간관계를 통한 검색에 사용하기도 한다. 또한, 이미지 내의 특정 영역의 문자들을 추출하거나 윤곽선의 특징점을 이용한 인식기

술에도 사용된다. 이미지 검색은 검색하고자 하는 이미지에 따라 이용되는 특성이 다를 수 있다. 이미지가 흑백 이미지면 질감이나 모양으로 검출하고, 다양한 색상이 혼합을 이룬 이미지면 색상값을 이용하여 검출한다. 이 중에 모양은 윤곽을 갖춘 객체를 포함한 이미지라면 검색에 이용할 수 있는 특성이다. 특히, 검색하고자 하는 이미지가 색상과 질감이 비슷한 도메인내에 있다면 모양을 이용하는 검색이 가장 효과적이다. 본 논문에서는 그림 1과 같은 식물의 잎 모양을 세그멘테이션하여 특징있는 점들의 좌표를 추출하였으며, 추출한 좌표를 통해 유사도를 계산하여 검색에 활용하였다.

모양기반의 이미지 검색에서는 이미지내의 객체를 표현하는 방법과 점들의 개수에 따라 시간적, 공간적 비용이 다르기 때문에 특징있는 점들만 추출하는 표현방법을 개발하는 것이 중요하다. 지금까지 많은 연구자들에

· 본 연구는 고려대학교 특별연구비와 과학기술부 국책연구개발 사업인 유전자원지원 활용사업단의 연구비(no. BDM0100211)의 지원에 의해 수행되었습니다.

[†] 정 회 원 : 아주대학교 정보통신전문대학원
youngman@ajou.ac.kr

^{**} 종신회원 : 고려대학교 전자컴퓨터공학과 교수
ehwang04@korea.ac.kr
(Corresponding author)

논문접수 : 2005년 4월 15일

심사완료 : 2005년 9월 5일

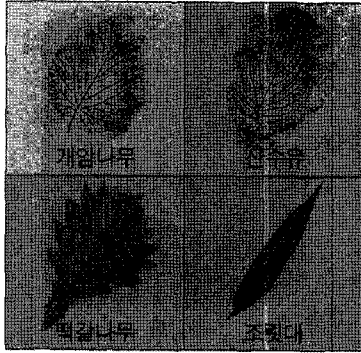


그림 1 여러 가지 식물의 잎

의해 표현방법들이 제안되고 개발되었으며, 이러한 표현 방법은 크게 외부 윤곽선을 이용하는 방법[1]과 내부 골격(skeleton)을 이용하는 방법[2]이 있다. 외부 윤곽선은 곡률(curvature), 중심거리, 복소수 좌표 함수 등을 통하여 추출하며[3,4], 푸리에 변환(Fourier transform)[5]을 거쳐 특징으로 사용된다. 내부 골격은 평면 영역의 구조적 형태를 그래프로 변환시켜 만든다. 본 논문에서는 외부 윤곽선의 곡률을 이용한 방법 중에 하나인 MPP (Minimum Perimeter Polygon)[22,23]를 개선하여 표현 방법으로 사용하였다.

모양을 표현하는 방법은 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 있다. 이러한 표현 방법을 통한 값은 이미지의 이동(translation), 크기(scale) 변경, 회전(rotation), 보는 각도(view angle)의 변환, 좌우 또는 위아래 대칭(symmetric transformation)에 대해서 변함이 없어야 한다(invariance).

이미지 검색에서 전체 모양을 그려 질의하는 것보다는 모양의 일부분만 그리는 부분질의를 하는 것이 더 효율적일 수 있다. 전체 모양을 닫힌 곡선(closed curve)이라 하고 일부분의 모양을 열린 곡선(open curve)이라 한다. 이러한 닫힌 곡선 뿐만 아니라 열린 곡선에 대해서도 일관성 있는 표현 기법이 사용되어야 한다.

또한, 모양을 표현하는 알고리즘과 매칭 알고리즘은 시간 복잡도가 선형시간(linear time)이어야 한다. 즉, 계산 시간과 저장공간에 대해서 효율적이어야 한다. 이를 위해서, 인덱싱과 매칭이 실시간으로 매칭이 이루어져야 하며, 그렇지 않으면 검색 결과에 대해서 보장할 수 없다.

본 논문의 구성은 다음과 같다. 2장에서는 내용기반 검색과 이미지 표현(Representation)에 대해서 알아보

고, 3장에서는 이미지 인덱싱에 대해 서술한다. 4장에서는 이미지와 매칭하는 방법에 대해서 설명하고, 5장에서는 실험 환경을 설명하고 성능을 평가한다. 마지막으로 6장에서는 결론과 향후 계획에 대해서 논의한다.

2. 관련 연구

2.1 윤곽선 검색(Edge Detection)

윤곽선 검색은 다양하며 윤곽선 검색방법마다 장단점이 있다. 본 절에서는 기존의 기법 중 간단하고 효과적인 방법들을 소개한다.

ACM(Active Contour Model)[12,13]은 검색이 진행되는 형태가 뱀이 움직이는 모양과 유사하여 Snake Model이라 불리며 snake 에너지 함수를 사용한다. 하지만 복잡한 윤곽선을 가진 이미지의 경우, 정확한 형태를 검색하기 어렵다. 다른 기법인 소벨(Sobel Operation)[14]은 네개의 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. 픽셀에 대한 이웃(neighbor) 픽셀의 밝기 값(gray value)을 이용한다. 즉, 하나의 픽셀은 8방향에 대한 이웃 픽셀이 존재하며, 각 방향에 대한 이웃 픽셀의 차이 값은 아래의 식 1을 통해 계산한다. 이 값이 임계값(threshold) 이상이면 윤곽선의 점으로 간주한다.

$$difference_d(x,y) = |g(x,y) - g_d(x,y)| \quad (1)$$

x, y : pixel
 $g(x,y)$: gray value
 d : direction

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

2.2 모양 표현(Shape Representation)

데이터를 간결하게 표현하는 것은 검색의 효율성, 정확성, 계산비용 면에서 중요한 요인이다. 일반적으로 모양 표현은 크게 영역 기반과 외곽선 기반으로 나눈다. 영역 기반은 2D 객체 내부 영역의 픽셀 분포(pixel distribution)를 표현하고 객체내부와 경계의 픽셀 모두를 고려하기 때문에 단일영역(single connected region)과 다중영역(multiple region) 모두에 적용 가능하다. 이 방법은 일반적인 영상에 모두 적용 가능하며 상대적으로 계산시간이 빠르며, 잡음(noise)에 영향을 덜 받지만, 단일 폐곡선의 단일 오브젝트에 낮은 검색 성능을 보이며, 객체를 기울여서 본 영상과 찌그러짐, 비틀림에 대

해 취약하다. 외곽선 기반은 닫혀져 있는 2D 객체의 외형을 표현한다. 유연한 형태의 객체 표현, 비틀림, 찌그러짐, 기울어짐에 강건하다. 그러나 잡음에 매우 민감하며, 단일 폐곡선의 단일 오브젝트로 이루어진 지극히 제한된 조건의 영상에 대해서만 검색이 가능하며 검색시간이 많이 소요된다는 것이 단점이다. MPEG-7[17,18]에서는 외곽선 기반의 모양 표현방법으로 CSS(Curvature Scale Space)[19]를 언급하고 있다. CSS는 외곽선을 1D 신호로 처리해서 분석하는 방법이며, 제로 크로싱(zero-crossing)을 조사하여 모양의 외곽선에서 오목점과 볼록점을 찾아낸다. CSS는 Scale Normalization, Curvature Derivation, CSS Map, CSS Peaks, Peak Normalization 단계를 거쳐 CSSD를 계산한다.

모양 표현 기법은 지금까지 많은 연구에서 제안되고 개발되었다. 시작점에서의 길이와 방향을 이용한 체인코드(chain code)[20]는 잡음에 민감하며, 임의의 각도에 대해서는 불가능하다. 퓨리에 변환은 윤곽선 기반의 검색방법 중 많이 사용되며, 퓨리에 변환을 통해 생성된 복소수 형태의 상관계수들을 모양 기술자(descriptor)로 이용한다. 다각형 근사(polygonal approximation)[21]는 폐곡선(closed curve)에 대해서 객체 모양의 지역 특징인 고정된 차수의 특징점(interest point)들을 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 알고리즘[22,23]을 사용하였다. 또한, 영역기반의 표현 방법인 골격화(skeletonizing)[2]는 평면 영역의 구조화된 모양을 그래프로 변환시킨다. 이러한 골격화는 MAT(medial axis transformation)[24]을 통해 만들 수 있으며, 영역(region) R과 경계(border) b가 있을 때, R안에 각점 p에 대해서 b의 가장 가까운 이웃을 찾는다. 만약 p가 하나 이상의 이웃을 갖게 되면, 이것을 중앙축(medial axis)이라 하며 골격(skeleton)이라고 한다.

2.3 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 (2)인 Lp Distance (Minkowski Distance) 이다[25]. 식에서 p값이 1이면 맨하탄 거리(Manhattan Distance)이고, p값이 2이면 유클리드 거리(Euclidean distance)이며, $p \rightarrow \infty$ 이면 Max Matrix가 된다.

$$D(U, V) = \left(\sum_{i=1}^k |u_i - v_i|^p \right)^{1/p} \quad (2)$$

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

이외에도 최소 가중치 거리와 최소 편차 거리를 이용하고 1:1 대응 함수에서만 사용되는 Bottleneck Distance[26] 함수가 있으며, 1:1 대응이 존재하지 않은 다른 크기의 두 점 집합에 사용되는 Hausdorff Distance[27]가 있다.

본 논문에서는 유클리드 거리인 식 (3)을 사용한다.

$$D(U, V) = \sqrt{\sum_{i=1}^k (u_i - v_i)^2} \quad (3)$$

2.4 잎 모양 검색에 대한 연구

지금까지 연구된 식물의 잎 이미지를 위한 검색은 여러 가지 모양 표현방법을 이용하여 이루어졌다[28,29]. [28]의 연구에서는 식물의 잎만을 검색하기 위해 점을 추출하는 방법으로 CCD(Centroid Contour Distance)를 사용하였다. CCD는 잎의 중심점에서 n개의 방향으로 일정한 각도($\Delta\theta$, $\Delta\theta = 2\pi/n$)로 나누어 중심거리를 계산하는 방법이지만, 볼록한 외곽선이 많은 경우 외곽선의 한 점에 대해서 거리가 여러 가지로 계산되는 문제가 있다. 이러한 문제점 때문에 대부분의 연구들은 모양 표현방법으로 CSS를 이용하여 점을 추출하였다. 대표적인 예로 [29]의 연구가 CSS를 사용하였다. [29]의 연구의 주된 내용은 식물의 잎들이 겹치는 경우(Self-Intersections) 외곽선이 다르게 추출되는 문제점을 개선하였다. 이 경우, 겹치지 않는 부분보다 겹치는 부분이 더 어둡게 보이는 것을 이용하여 겹쳐서 가려진 부분의 외곽선을 재구성하였다.

3. 이미지 인덱싱

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

3.1 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 MPP의 시퀀스이다.

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[30]으로 구현하였으며, 아래 표 1의 작업 순서대로 점들의 시퀀스(sequence)를 찾는다.

이러한 MPP 알고리즘을 식물의 잎 이미지에 적용하면 그림 2와 같은 이미지를 얻을 수 있다.

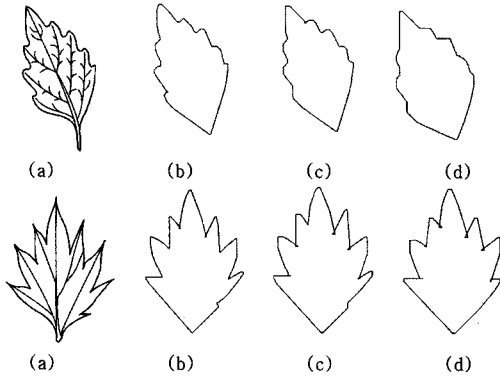


그림 2 식물의 잎에 적용한 MPP 알고리즘의 예, (a) 원본이미지 (b) cell size=2 (c) cell size=3 (d) cell size=5

3.2 우세점 추출

MPP 알고리즘은 점과 점 사이를 연결한 선을 기준으로 볼록한 점(180°이하)과 오목한 점(180°이상)으로만 점들을 찾기 때문에 추출한 점들 중에는 불필요한 점들이 많이 포함되어 있다. 이러한 점들을 간결화하기 위해 각 점들의 각도를 계산하여 임계값의 범위를 벗어나는 점들은 병합하였다. 다음의 표 2는 이러한 점들의 병합 여부를 판단하는 알고리즘이다.

알고리즘에서 $\triangle ABC$ 의 세 점 A, B, C 의 각도를 도출하기 위해 식 (4)와 같이 코사인 제 2법칙(the second law of cosines)을 사용하였다.

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
}
    
```

$$\cos A = \frac{b^2 + c^2 - a^2}{2bc} \quad (4)$$

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

점 B 의 좌표를 (x_j, y_j) 이라 하고, 점 C 의 좌표를 (x_i, y_i) 이라 하면 \overline{BC} 의 길이 a 는 식 (5)와 같다.

$$a = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (5)$$

표 3과 그림 3은 임계값이 160°일 때, 불필요한 점들을 병합하여 추출한 결과를 보여주고 있다.

3.3 입력값 보정(invariance)

입력되는 이미지들은 이동, 크기, 회전에 대해서 값이 불변해야 한다. 예를 들어, 잎이 세로인 경우와 가로인 경우, 모양에 대한 표현값은 동일해야 한다. 본 시스템에서는 우세점들간의 거리에서 가장 최대거리에 있는 점 2개를 기준으로 하여 회전에 대한 각도를 보정한다. 각 점들에 대한 보정값을 바탕으로 좌우측 끝점과 위아래 끝점으로 이동과 크기에 대한 값을 보정한다.

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 sequence 의 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.

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

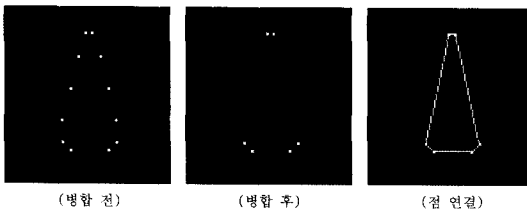


그림 3 개선된 MPP 알고리즘을 통해 추출한 시퀀스

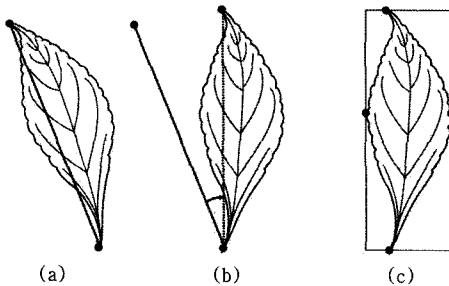


그림 4 입력값 보정의 단계, (a) 원본이미지 (b) 회전각 보정 (c) 이동과 크기 보정을 위한 끝점 검출

표 4 입력값 보정 알고리즘(회전각 보정)

```

Input
    crt_1, crt_2 : 기준선의 양 끝점
Rotating_point (point, N, crt_1, crt_2){
    cos=(crt_2[X]-crt_1[X])/get_distance(crt_1, crt_2);
    sin =sqrt(1- cos^2);
    for (i=0; i<N; i++){
        point[i][X]= cos*(point[i][X]-crt_1[X])+sin
                    *(crt_1[Y]- point[i][Y])-crt_1[X];
        point[i][Y]= sin*(point[i][X]-crt_1[X])-cos
                    *(crt_1[Y]-point[i][Y])- crt_1[X];
        if(crt_1[Y]<=point[i][Y])
            add_point(result,point[i])
    }
    return result;
}
    
```

4. 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.

4.1 유사도 계산

유사도를 계산하기 위해 식 (3)의 유클리드 거리를 사용하였으며, 아래의 식 (6)과 같이 질의한 이미지의 점들(u)과 데이터베이스에 저장된 이미지의 점들(v)과의 거리에서 최소값을 가지는 거리(D_i)를 이용하여 계산하였다.

$$S(U, V) = \frac{1}{|u|} \sum_{i=1}^{|u|} \min(D_i(u, v)) \quad (6)$$

여기서 |u| 는 질의한 이미지의 점들의 개수를 말하며, 함수 D_i의 값은 u_i에 대한 v_i의 거리로써 열거형의 형태를 지닌다.

4.2 최단거리 검색

질의한 이미지의 점들과 이미지 데이터베이스의 점들에서 거리의 최소값을 찾는 시간 복잡도는 O(|u| |v|) 이므로 O(n²)이다. 이러한 brute-force 방식보다 복잡도를 줄이기 위해 ε-nearest neighbor(e-NN) 검색을 사용한다. 이 방법을 사용하면, O(Dpolylog(N))의 시간 복잡도를 갖는다[31].

4.3 동적인 매칭 알고리즘

검색 시간을 최소화하기 위해 동적으로 매칭할 수 있도록 하였다. 우선 식물의 잎이 좌우대칭이라는 특징을 이용하여 우세점들 {x|x=1,2,3,...n} 중에 가장 긴 거리를 갖는 직선(축)을 기준으로 한쪽의 점들 {x|x=1,2,3,...n/2} 만을 이용한다. 또한, 유사도 계산시에 유사도가 특정 임계값을 초과하면 더 이상 매칭 프로세스를 진행하지 않고 다음 이미지의 유사도를 계산한다. 또한, 점의 개수가 특정 최대값보다 많으면 샘플링 함수(SMP())를 통해 n에서 m개만큼 추출한다. 예를 들면, 질의한 이미지의 우세점의 개수가 |u|, 매칭 대상 이미지의 우세점 개수가 |v|, |u| < |v| 이면 SMP(v)를 통해 샘플링한 우세점을 추출하여 유사도를 계산한다(표 5 참조).

5. 실험결과

본 논문에서 제안하는 알고리즘의 성능을 분석하기 위해서 Xeon CPU 2.8GHz, 1GB RAM의 PC에서 실험하였으며, 데이터베이스는 Microsoft SQL Server 2000 을 사용하였다. 검색 시스템의 성능을 분석하기 위해 이

표 5 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
}
    
```

미지는 국내에서 자생하고 있는 식물들을 수록하고 있는 대한식물도감[32]에서 1032종의 잎 이미지를 발췌하여 500×500 픽셀의 크기로 스케치하여 사용하였다. 이 용된 스케치 이미지가 정면이 아닌 0°~90°범위에서 보는 각도가 다르게 묘사되어 있어, 모든 이미지의 보는 각도를 0°로 통일시켜 재포맷화 하였다. 이와 함께 제안한 알고리즘을 이용하여 실험 데이터에 대해 재현률(recall)과 정확률(precision)을 비교분석하였다.

$$Recall = \frac{\text{relevant images retrieved}}{\text{relevant images for the query in the database}} \quad (7)$$

$$Precision = \frac{\text{relevant images retrieved}}{\text{all images retrieved}} \quad (8)$$

그림 5는 식물에서 볼 수 있는 엽서(잎차레)이며 식물의 종마다 서로 다르다. 이러한 엽서의 특징을 이용하여 본 시스템에서는 그림과 같이 어긋나기(a), 마주나기(b), 둘러나기(c)로 분류하여 검색에 이용하였다. 엽서 검색은 사용자가 잎차레를 스케치하면 중심 줄기에서 자라는 잎 들 간의 간격과 방향으로 엽서를 인식하게 된다.

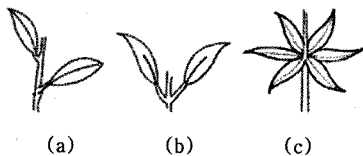


그림 5 검색에 사용된 엽서

그림 6은 잎 모양과 엽서를 스케치하여 검색하여 추출한 결과이다. 그림에서처럼 유사도 값이 가장 작은 순으로 정렬하여 보이고 있다.

그림 7은 본 논문에서 제안한 개선된 MPP, MPP, Fourier Descriptor, CSSD(Curvature Scale Space Descriptor), CCD(Centroid Contour Distance), Moment Invariants를 사용했을 때의 recall과 precision을 보이고 있다. 그림에서 알 수 있듯이 제안한 알고리즘이 다른 알고리즘의 비해 더 좋은 성능을 보이고 있다.

표 6은 셀 크기에 따른 평균 검색시간을 보이고 있다. 셀 사이즈에 따라 검색시간이 비례함을 알 수 있으며, 기존의 방법보다 동적인 매칭 알고리즘의 검색시간이

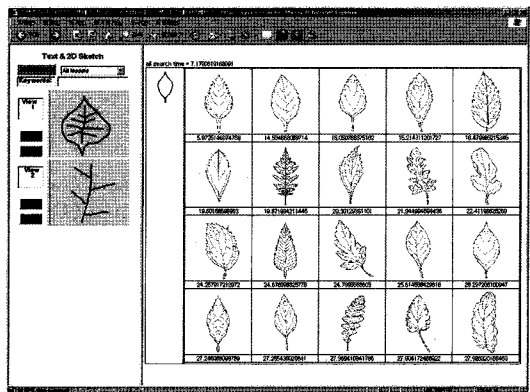


그림 6 스케치를 통한 검색결과

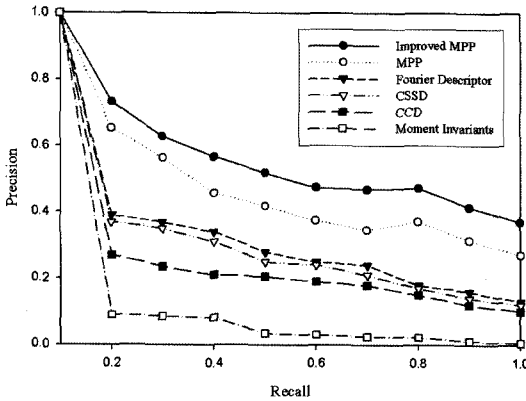


그림 7 Precision과 Recall

표 6 매칭 알고리즘과 셀 크기에 따른 평균 검색시간

셀 사이즈	검색시간		A/B
	NN-search(A)	Dynamic matching(B)	
5	29.57초	13.57초	2.18
7	16.58초	7.26초	2.12
9	12.45초	5.80초	2.15

약 2.2배 빠른 것을 알 수 있다.

6. 결론 및 향후 연구 과제

본 논문에서는 식물의 잎을 검색하기 위해 MPP 알고리즘을 개선하여 이미지의 외곽선에서 우세점을 추출 하였으며 빠른 검색과 유연한 검색을 할 수 있도록 동적인 매칭 알고리즘을 제안하였다. 또한 잎 모양뿐만 아니라 잎 차례를 이용한 방법을 사용하여 검색의 성능을 증대시켰다. 실험에서 개선한 표현 알고리즘이 기존의 방법보다 좋은 성능을 얻었으며 NN-검색과 유클리드 거리를 이용한 매칭 알고리즘보다 동적인 알고리즘이 검색시간을 약 2.2배 단축시켰다. 향후 연구 과제로는 좀 더 개선된 알고리즘을 개발하고, 잎 모양 뿐만 아니라 잎 내부의 잎맥을 이용하여 검색할 수 있도록 개발 하는 것이다.

참고 문헌

[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] Martinez JM (2004) MPEG-7 Overview (version 10). ISO/IEC JTC1/SC29/WG11 N6828.
 [18] B.S. Manjunath, Philippe Salembier, Thomas Sikora, Introduction to MPEG-7: Multimedia Content Description Interface, John Wiley & Sons, 2002.
 [19] Mokhtarian F, Abbasi S, Kittler J. "Efficient and robust retrieval by shape content through cur-

- vature scale space," Int Workshop on Image DataBases and Multimedia Search, Amsterdam, The Netherlands, pp 35-42, 1996.
- [20] Freeman, H., Saghri, J., "Comparative Analysis of Line Drawing Modelling Schemes," Computer Graphics and Image Processing, Vol. 12, 1980.
- [21] Kurozumi Y., Davis W.A., "Polygonal approximation by the minimax method," Computer Vision, Graphics and Image Processing, pp.248-264, 1982.
- [22] Sklansky, Chazin et al. "Minimum perimeter polygons of digitized silhouettes," IEEE Transactions on Computers, Vol.21, No.3, pp.260-268, 1972.
- [23] Sklansky J., "Finding the Convex Hull of a Simple Polygon," Pattern Recognition Letters, Vol.1 No.2, pp.79-84, 1982.
- [24] Lee, D.T., "Medial axis transformation of a planar shape," IEEE Transactions On Pattern Analysis and Machine Intelligence, pp.363-369, 1982.
- [25] Veltkamp, R., "Shape matching: similarity measures and algorithms," Technical Report UU-CS-2001-03, Utrecht University, the Netherlands, 2001.
- [26] Efrat, A. and Itai, A., "Improvements on bottleneck matching and related problems using geometry," The 12th Symposium on Computational Geometry, pp.301-310, 1996.
- [27] Alt, H., Behrends, B. and Blomer, J., "Approximate matching of polygonal shapes," Ann. Math. Artif. Intell., Vol.13, pp.251-266, 1995.
- [28] 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.
- [29] Mokhtarian, F., and S. Abbasi, "Matching Shapes with Self-Intersections: Application to Leaf Classification," IEEE Transactions on Image Processing, vol. 13, no. 5, pp. 653-661, 2004.
- [30] The MathWorks - MATLAB and Simulink for Technical Computing <http://www.mathworks.com>
- [31] 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.
- [32] 이창복, 대한식물도감, 향문사, 서울, 1982.



황 인 준

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 8월~2003년 2월 아주대학교 정보통신전문대학원 조교수. 2003년 3월~2004년 8월 아주대학교 정보통신전문대학원 부교수. 2004년 9월~현재 고려대학교 전자컴퓨터공학과 조교수. 관심분야는 데이터베이스, 멀티미디어 시스템, 정보 통합, 전자상거래, 영상처리, 유비쿼터스 컴퓨팅



남 윤 영

2001년 아주대학교 정보 및 컴퓨터공학과 학사. 2001년~2003년 아주대학교 정보통신전문대학원 석사. 2003년~현재 아주대학교 정보통신전문대학원 박사과정 관심분야는 데이터베이스, 멀티미디어 시스템, 정보 통합, 영상처리. 패턴인식, 컴

퓨터 비전