

푸리에 시리즈를 사용한 외형기반 이미지 검색

탁윤식⁰ 황인준

고려대학교 대학원 전자전기공학과

{life993⁰, ehwang04}@korea.ac.kr

Shape Based Image Retrieval using Fourier Series

Yoonsik Tak⁰ Eenjun Hwang

School of Electrical Engineering, Korea University, Seoul, Korea

요 약

푸리에 시리즈를 사용하면 이미지의 외곽선 특성을 표현할 수 있다. 이미지의 푸리에 계수를 추출하기 위해서는 우선 이미지를 구성하는 주요 오브젝트를 표현하는 곡선을 추출한다. 이러한 곡선은 오브젝트의 특정 중심점에서 외곽선을 따라 일회전하면서 그 거리를 좌표상에 표시함으로써 얻을 수 있다. 기존의 푸리에 계수를 추출하는 방법들은 추출된 계수를 이용하여 해당 곡선을 복원했을 때 원래의 곡선에 존재하던 상세한 특성을 표현하지 못한다는 단점이 있으며 이는 결국 이미지로부터 추출한 곡선을 사용하여 이미지를 검색할 때 정확도를 상당히 떨어뜨리게 한다. 이러한 문제점을 해결하기 위해서 본 논문에서는 Binary Range Reduction (BRR) 알고리즘을 제안한다. BRR 알고리즘은 원래의 곡선과 푸리에 계수를 통해서 복원된 곡선간의 차이를 줄이기 위해서 전체의 곡선을 통해서 하나의 푸리에 계수 세트를 추출하지 않고, 복원된 곡선이 원래의 곡선과 차이가 일정 크기 이상 나지 않도록 푸리에 계수를 추출하는 구간을 나누어가며 푸리에 계수를 추출한다. 이렇게 추출된 다수의 푸리에 계수 세트를 통해서 복원된 곡선을 사용하여 이미지들 간의 유사도를 비교한다. 실험을 통하여 BRR 알고리즘을 사용하여 곡선에서 추출한 푸리에 계수로 복원한 곡선이 원래 곡선의 특성을 정확하게 표현하고 있음을 보였고, 푸리에 계수와 BRR 알고리즘을 이미지 검색에 적용하였을 때, 높은 검색 결과를 얻을 수 있음을 보였다.

1. 서론 및 관련연구

이미지 데이터베이스로부터 효율적인 이미지 검색과 인덱싱에 관한 연구가 다양한 연구 영역에서 제안되어 왔다. 기존의 연구들은 이미지의 특징이나 이름 등의 메타 데이터를 저장해 두고, 이 메타 데이터와 사용자가 입력한 키워드를 비교하여 이미지를 검색하는 키워드 입력 방식의 이미지 검색 시스템이 주류를 이루었다. 하지만, 같은 이미지에 대해서도 사람마다 보고, 생각하는 것이 서로 다르므로, 서로 다른 키워드를 입력하게 된다. 이런 상황을 통한 검색 시스템이 올바른 결과를 내지 못하게 하였다. 이러한 문제를 해결하기 위해서 키워드가 아닌 내용 기반(Content-Based) 검색이 가능한 다양한 방법이 제시되었다. 외형의 외형 모양을 사용하여 유사한 이미지를 검색하는 방법[1]과 물고기의 외형 모양을 사용하여 물고기 이미지들간의 유사도를 비교하여 가장 유사한 이미지를 검색하는 방법[2] 등이 그 예이다. 이러한 내용기반의 이미지 검색 방법은 색깔, 질감, 외형정보 등의 낮은 레벨의 시각적인 특징 정보 및 이미지 내 물체간의 공간정보 등 인간의 시각관점에

서 사용되는 유용한 정보를 사용하여 유사한 이미지를 검색하는 방법이다. 이러한 특징 정보 중에서도 색깔 정보는 이미지 검색에서 널리 사용되는 속성이다. 유사한 이미지를 검색하기 위해서 공간 색깔 히스토그램[3]과 색깔 히스토그램의 통계학적 분포 예측 방법[4]은 이미지 검색에 색깔 히스토그램을 적용한 예이다. 색깔 정보 외에 외형 정보는 인간이 이미지를 인식함에 있어서 중요하게 인식되는 또 하나의 중요한 특징이다. 그리고, 검색이 비슷한 색깔의 이미지들 사이에서 일어날 경우, 색깔 정보를 사용하는 것 보다는 외형 정보를 사용하는 것이 더 나은 검색 결과를 보여준다. Minimum Perimeter Polygon(MPP) 알고리즘을 사용해서 외형에 가장 가까운 다각형을 얻는 방법[5]과 이미지의 주요 오브젝트로부터 구한 중심점과 외곽선의 거리를 통해 얻은 그래프를 사용하는 방법[6], 중심점과 외곽선의 외부점 사이의 거리를 계산한 Center-Contour Distance Curve(CCD)를 사용하는 방법[7]들은 이미지의 외형정보를 검색에 사용한 예이다. 그리고 외형의 외형을 꼭지점의 위치를 통해서 표현하는 방법[8]과 두개의 인접한 꼭지점들을 연결하는 선들의 집합과 꼭지점들의 각의 정보를 사용해서 외형의 외형을 표현하는 방법[9]이 있다.

본 논문에서는 외형 정보를 특징 정보로 사용하는 외형 이미지 검색 시스템을 제안한다. 이미지를 표현하는 특징의 하나로써 이미지 외형을 잘 나타낼 수 있는 곡선을 추출하고, 이 곡선에서 푸리에 계수를 계산하여 데이터베이스에 저장한다. 이러한 푸리에 계수를 추출하기 위한

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업과 과학기술부 국책연구 개발 사업인 유전자원 지원 활용사업단의 연구비 (no. BDM0100211)의 지원에 의해 수행되었습니다.

곡선은 이미지의 주요 오브젝트의 중심점을 구한 후, 중심점과 오브젝트의 외곽선과의 거리를 계산한 그래프를 통하여 얻는다. 기존의 방법으로 추출된 퓨리에 계수는 곡선 전체를 통해 하나의 퓨리에 계수 세트를 구하기 때문에, 이렇게 구해진 퓨리에 계수로서는 자세한 부분은 생략된 대략의 곡선밖에 복원하지 못한다. 따라서, 본 논문에서는 원본 곡선과 그 곡선에서 추출된 퓨리에 계수로 복원된 곡선과의 오차를 줄이기 위해서 Binary Range Reduction (BRR) 알고리즘을 제안한다. BRR 알고리즘은 원본 곡선과 복원된 곡선의 오차가 일정 크기 이상일 때, 추출하는 구간을 줄여가며 일정 크기 이하의 오차를 가지는 다수의 퓨리에 계수 세트를 구함으로써, 보다 정확한 곡선을 복원할 수 있게 한다.

논문의 구성은 다음과 같다. 2장에서는 퓨리에 계수를 추출하기 위한 곡선을 이미지에서 추출하는 과정을 설명한다. 3장에서는, 곡선으로부터 퓨리에 계수를 추출하는 과정과 BRR 알고리즘에 대한 설명을 한다. 4장에서는 데이터베이스에 저장된 퓨리에 계수와 질의로 입력받은 이미지를 사용한 유사 이미지 검색 방법에 대해서 설명을 하고, 5장에서는 실험을 통해 제안된 알고리즘들에 대한 성능을 평가한다. 마지막으로 6장에서는 본 논문의 결론에 대해 기술한다.

2. 이미지의 주요 오브젝트 및 곡선 검출

이 장에서는 이미지 검색에 사용될 퓨리에 계수를 추출하기 위해서 곡선을 구하는 기법에 대해 기술한다. 이러한 곡선을 구하기 위해서, 이미지내의 주요 오브젝트 및 외곽선을 추출하는 기법과, 주요 오브젝트의 중심점을 찾는 기법을 소개한다.

2.1 주요 오브젝트 및 외곽선 검출 기법

이미지 검색과 인덱싱의 첫 단계는 이미지의 주요 오브젝트 및 외곽선을 검출하는 것이다. 기존의 주요 오브젝트와 외곽선 검출 방법들은 많은 오브젝트들이 겹쳐져 있거나 배경이 복잡할 경우 정확한 오브젝트를 찾을 수 없기 때문에, 외곽선 또한 정확하게 찾을 수 없다는 단점이 있었다. 따라서, 이 논문에서는 배경이 단색이고 단일 오브젝트만 있는 이미지를 사용한다는 제약을 두었다.

주요 오브젝트를 나타내고 중심점을 찾기 쉽게 하기 위해서 검출된 외곽선의 밖은 흰색, 안은 검은색으로 채웠다. 이미지 내의 주요 오브젝트와 외곽선을 검출하기 위해 Sobel, Canny, roberts, Prewitt 알고리즘을 참고하였으며, 알고리즘 1에 그 과정을 기술하였다.

알고리즘 1. 주요 오브젝트 검출

```

procedure Main_Object_Extraction()
begin
    read pixel data in image as a m * n matrix.
    threshold_value ← OtsuCriticalValue() within all
    elements of image.
    transform image into binary information
    (white or black) using threshold value
    if size of separated area < threshold size
        remove the separated area and fill it to
        the color of its around
    fill the outside of boundary to 1 (white).
    fill the inside of boundary to 0 (black).
end
    * OtsuCriticalValue() calculates the threshold that
    minimizes the difference between intraclass of black
    and white pixel. We use entire value of pixels to
    calculate the threshold
    
```

2.2 중심점 찾기

주요 오브젝트를 검출한 후에는, 오브젝트의 중심점을 찾아야한다. 이 중심점은 회전이나 스케일링 작업을 거친 후에도 변하지 않아야 한다. 중심점은 2.1절에서 검출한 오브젝트를 나타내는 검은 픽셀들의 X, Y값들의 평균값을 구함으로써 얻을 수 있다. 중심점을 찾는 과정은 식 (1)에 나타나 있으며, X_i 는 i 번째 검은 픽셀의 X 좌표값이며, Y_i 는 i 번째 검은 픽셀의 Y 좌표 값이다. X_c 는 중심점의 X 좌표값이고, Y_c 는 중심점의 Y 좌표값이다.

$$\begin{cases} X_c = \frac{1}{N} \sum_{i=1}^N X_i \\ Y_c = \frac{1}{N} \sum_{i=1}^N Y_i \end{cases} \quad N : \text{Number of Black Pixels} \quad (1)$$

2.3 곡선 구하기

이미지의 주요 오브젝트의 곡선을 얻기 위해서는, 섹션 2.1에서 얻어진 주요 오브젝트의 외곽선을 1회 회전시킨다. 회전이 이루어지는 동안, 중심점과 외곽선의 현재 점과의 거리를 계산한다. 본 논문에서는 계산된 거리값을 곡선의 진폭으로 한다.

2.4 곡선의 진폭 조정

스케일된 이미지와 원본 이미지가 동일하다고 인식되기 위해서, 이 두 이미지의 곡선 역시 서로 동일하다고 인식이 되어야한다. 모든 곡선의 최대 진폭이 동일하게 하고, 곡선의 최대 진폭이 변화된 비율만큼 다른 곡선의 점들도 변화시킴으로써 이러한 문제를 해결할 수 있다. 그러기 위해, 2.3절에서 얻어진 곡선의 진폭이 최대가 되는 지점의 위치를 찾고 이 위치의 진폭을 $D_{longest}$ 로 한

다. 곡선의 다른 점들의 진폭은 이 값에 맞춰서 스케일링시킨다. 이 방법은 알고리즘 2에 소개 되어 있다.

알고리즘 2. 곡선의 진폭 재조정

```

procedure Scaling_Curve()
begin
    Dlongest <- 0
    for i=0 to No. of points in curve - 1
        if Dlongest < D
            Dlongest <- Di
    end
    R <- Constant Distance / Dlongest
    for i=0 to No. of points in curve - 1
        Di <- Di * R
    end
end
    * We define 'Constant Distance' as arbitrary value
    * Di is the ith amplitude in the curve
    
```

3. 퓨리에 계수 추출

퓨리에 계수 값은 원본 곡선에서 계산된 사인 함수와 코사인 함수의 계수 값들로 이루어지며, 원본 곡선은 퓨리에 계수를 통해 복원된 사인함수와 코사인 함수의 합으로써 복원될 수 있다. 따라서, 2.3절에서 얻어진 곡선으로부터 계산된 퓨리에 계수를 이용하여 원본 곡선을 복원 할 수 있고, 또한 데이터베이스에 저장된 퓨리에 계수 정보를 이용하여 두 개의 이미지의 유사성을 비교 할 수 있다.

3.1 퓨리에 계수 추출 과정

퓨리에 계수들은 A_{zero}의 값과 a[1] ~ a[계수의 수]의 사인 곡선의 계수들, b[1] ~ b[계수의 수]의 코사인 곡선의 계수들로 이루어져 있다. 퓨리에 계수의 값을 얻기 위한 프로그램 코드는 알고리즘 3에 나와 있다.

알고리즘 3. 퓨리에 계수 추출

```

procedure Extract_Coefficient (s_intv, f_intv)
    Azero ← 0
    for i=s_intv to f_intv
        sum Center_to_Contour[i] to Azero
    end
    divide Azero by f_intv - s_intv
    for i=0 to No. of Fourier Coefficient
        initialize Atemp and Btemp to 0.
        for j=s_intv to f_intv
            sum
            Center_to_Contour(i) × cos( $\frac{(i+1) \times 2\pi \times (j-s\_intv)}{f\_intv-s\_intv}$ )
            to Atemp
        sum
    
```

```

        Center_to_Contour(i) × sin( $\frac{(i+1) \times 2\pi \times (j-s\_intv)}{f\_intv-s\_intv}$ )
        to Btemp
    end
    Avalue[j] =  $\frac{2 \times A\_value\_temp}{f\_intv-s\_intv}$ 
    Bvalue[j] =  $\frac{2 \times B\_value\_temp}{f\_intv-s\_intv}$ 
end
end
    * Center_to_Contour is differences between center point and outer points of object.
    
```

3.2 Binary Range Reduction (BRR Algorithm)

3.1절에서 언급한 퓨리에 계수를 구하는 방법은 복잡한 이미지로부터 퓨리에 계수를 추출 한 경우, 추출된 퓨리에 계수로부터 곡선을 복원했을 때, 원본 곡선의 자세한 형태를 완벽히 복원하기 힘든 문제가 있다. 그림1의 예에서 이러한 문제를 찾아볼 수 있다.

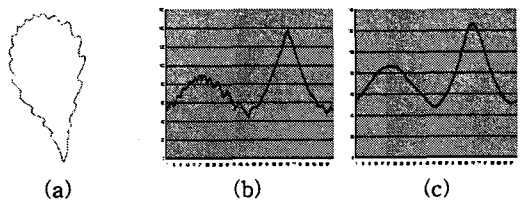


그림 1. 이미지의 정확한 곡선과 그 곡선에서 추출한 퓨리에 계수로 복원한 곡선의 예 (a) 원본 이미지, (b) 원본 이미지의 곡선, (c) 추출한 퓨리에 계수로부터 복원된 곡선

본 논문에서는 퓨리에 계수들을 이용하여 복원한 곡선과 원본 곡선의 차이점을 줄이기 위해서 원본 곡선과 그 곡선에서 추출한 퓨리에 계수들로 복원한 그래프간의 에러를 최소화시키는 Binary Range Reduction(BRR) 알고리즘을 제안한다. 만일 퓨리에 계수를 추출하는 곡선의 영역이 충분히 작다면, 저장되어 있는 퓨리에 계수로부터 원본 곡선과 거의 동일한 곡선을 얻어 낼 수 있다. 그러므로 곡선의 시작점으로부터 끝점까지의 퓨리에 계수 추출 구간을 원본 곡선과 복원된 곡선의 차이가 충분히 작아 질 수 있을 때까지 반으로 줄여 나간다. 줄여진 구간에서의 퓨리에 계수를 추출한 후, 다음 구간에서 동일한 방법으로 퓨리에 계수를 추출하게 된다. 이 과정을 전체 곡선으로부터 퓨리에 계수가 얻어 질 때까지 반복 한다. BRR 알고리즘은 알고리즘 4에 나와 있다.

알고리즘 4. Binary Range Reduction (BRR)

```

procedure Binary_Range_Reduction ()
  s_interval <- 0
  f_interval <- No. of point in a curve
  while(true)
    Extract_Coefficient(s_interval, f_interval)
    for i = s_interval to f_interval
      diff <- diff +
        ABS|Di,query_image - Di,restored_image|
    end
    if diff > threshold value
      f_interval <- (f_interval-s_interval)/2
    else
      s_interval <- f_interval + 1
      f_interval <- No. of points in a curve
      store length of current interval and
        fourier coefficients.
      if s_interval > No. of points in a curve
        break
    end
  end
end

* Dij is amplitude of ith point in a curve of image j
* Threshold value is achieved divide the total
  difference of original curve and restored curve using
  Fourier Coefficients by number of points in a
  cycle.(This threshold value is used to divide space,
  where extracting Fourier Coefficients, whose difference
  is bigger than threshold value * number of point in a
  cycle.)
    
```

그림 2에서는, BRR 알고리즘을 이용해서 얻은 푸리에 계수로부터 복원된 곡선이 BRR 알고리즘을 적용하지 않고 원본 곡선으로부터 추출된 푸리에 계수로부터 복원된 곡선보다 나음을 볼 수 있다.

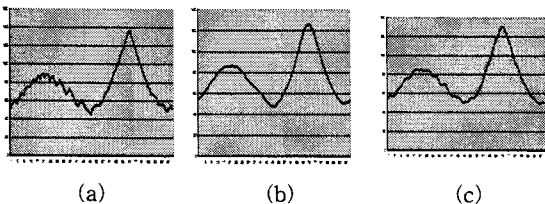


그림 2. BRR 알고리즘을 적용했을 때와 적용하지 않았을 때의 복원된 곡선의 차이를 나타낸 예 (a) 원본곡선, (b) BRR을 적용하지 않고 복원한 곡선, (c) BRR을 적용한 후 복원한 곡선

이렇게 얻어진 푸리에 계수들의 세트의 집합은 각각의 푸리에 계수들의 세트에서 곡선을 부분적으로 복원할 수 있고, 이렇게 복원된 부분적인 곡선들을 모두 연결하면 전체의 곡선을 얻을 수 있다.

4. 유사 이미지 검색

질의로 찾고자하는 이미지가 들어왔을 때, 원본이미지에서 2장에서 제안한 방법을 사용하여 추출한 곡선과, 데이터베이스에 저장된 이미지들의 푸리에 계수 세트들로부터 복원한 곡선들을 1:1로 비교해가며 가장 유사한 순서대로 정렬을 하게 된다. 이 과정은 알고리즘 5에 나와 있다.

알고리즘 5. 유사 이미지 검색

```

procedure Similar_Image_Search()
  while(Read_Next_Image() is true)
    while(Read_Next_Period_coefficients is true)
      Restore part curve
    end
    add all part curves into a unified curve.
    difference <- 0
    for i = 0 to No. of points in a curve
      difference <- difference +
        ABS(Di,restored_image - Di,restored_image)
    end
    end
    sort using calculated difference.
  end
  * Read_Next_Period_coefficients function is read
  partial interval of curve divided by BRR Algorithm.
  If there is a Fourier Coefficients, return true or
  return false.
  * Read_Next_Image() function is read Fourier
  Coefficients of the next image. If there is a Fourier
  Coefficients, return true or return false.
    
```

5. 실험 결과

본 논문에서는 제안된 알고리즘들의 성능을 평가하기 위한 실험을 하였다. 이러한 실험을 위해서, 인텔 펜티엄4 3.0GHz CPU와 1GB RAM의 시스템이 사용되었다. 실험을 위해서, 대한식물도감[10]에서 잎 이미지를 구하여 사용하였고, 푸리에 계수를 사용한 검색 프로그램 구현을 위해, C언어가 사용되었다.

우선 검색률을 얻기 위한 실험을 하였다. 실험을 위해서 스케치한 잎이미지가 질의로 사용되었다. 결과는 스케치한 이미지로 질의를 던졌을 때, 2장과 3장에서 제안된 알고리즘을 적용하여 유사한 이미지의 순으로 정렬하였을 때, 전체 잎 이미지 수의 1% - 10%안에 스케치한 이미지의 원본 이미지가 들어 갈 수 있는지를 계산한 확률로써 나타내었다. 그 결과는 표 1에 나타나 있다.

표 1. 검색률

Method	r = 1%	r = 2%	r = 3%	r = 4%	r = 5%	r = 10%
Search Result	60%	73.33%	73.33%	73.33%	73.33%	93.33%

두 번째, 잎 이미지를 그림 3에서와 같이 3가지 타입으로 나누었다. 첫 번째 타입은 주요 오브젝트의 외곽선

이 굴곡이 거의 없이 매끈한 경우이다. 2번째 타입은 주요 오브젝트의 외곽선이 주기가 비교적 긴 곡선으로 이루어진 것이다. 마지막으로 3번째 타입은 주요 오브젝트의 외곽선이 주기가 짧은 곡선으로 이루어진 것이다.

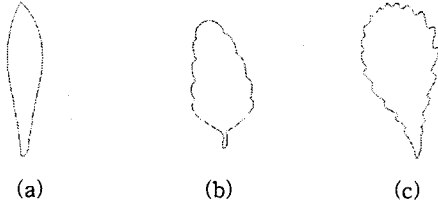


그림 3. (a) Type 1:외곽선이 매끈한 경우, (b) Type 2:외곽선이 주기가 비교적 긴 곡선으로 이루어진 경우, (c) Type 3:외곽선이 주기가 짧은 곡선으로 이루어진 경우, 외곽선의 모양에 따른 3가지 타입의 잎 이미지

다음으로 원본 곡선과 BRR을 사용하지 않은 위의 3가지 타입의 곡선들과의 차이와, BRR의 임계값을 2.5, 2.0, 그리고 1.7을 주었을 때의 차이를 구하였다. 2.5.2 섹션에서 기술하였듯이, BRR을 적용할 때 어느 구간의 원본 곡선과 복원된 곡선과의 차이가 어떤 값보다 작으면 그 구간의 퓨리에 계수를 확정하고 어떤 값보다 크면 그 구간을 다시 반으로 나누어 차이를 다시 계산하는데 여기서 사용되는 값이 임계값이다. 만약, 임계값이 2.5 이고, 그 구간에 속한 점들의 수가 100이면, 2.5 * 100 인 250 이상의 차이가 나게 되면 그 구간을 반으로 다시 나누게 된다.

Type 1의 경우, Type 2, 3에 비해서 원본 곡선과 복원된 곡선과의 차이가 크지 않았다. 이것은 Type 1의 경우 외곽선의 굴곡이 거의 없어서 큰 차이 없이 원본 곡선을 복원할 수 있음을 의미한다. 원본 곡선과 복원된 곡선과의 차이가 크지 않기 때문에 임계값을 낮추더라도 차이가 줄어들지 않았다. 반면, Type 2와 Type 3의 경우, 외곽선의 굴곡이 상당히 있기 때문에, Type1에 비해 원본 곡선과 복원된 곡선과의 차이가 크고, 임계값을 낮출수록 BRR알고리즘에 의해 나누어지는 곡선이 많아져서 두 곡선과의 차이가 줄어들게 된다. Type 2에 비해 Type 3의 굴곡이 더 크기 때문에, Type 3가 원본 곡선과 복원된 곡선과의 차이가 크다. 이 결과는 표 2에 나타나 있다.

표 2. 3가지 타입에 따른 차이

Threshold \ TYPE	Without BRR	T = 2.5	T = 2.0	T = 1.7
TYPE1	140.8	140.8	140.8	140.8
TYPE2	197.2	197.2	179	164.8
TYPE3	240.6	210.6	176.4	159.8

표 3은 표 2에서 계산된 BRR 알고리즘의 임계값을 변화시켰을 때 원본 곡선과 복원된 곡선과의 차이를 사용

하여, BRR알고리즘을 적용하였을 때, 각각의 임계값, 잎 이미지의 타입별로 BRR알고리즘을 적용하지 않았을 때보다 어느 정도 차이가 줄었는지를 비율로써 나타낸 것이다. 이와 같은 결과를 얻기 위해서 식 (2)를 사용하였다.

$$\frac{((\text{Difference with no BRR} - \text{difference with BRR}) / (\text{Difference with no BRR})) * 100}{(2)}$$

표 3. 계수의 변화에 따른 감소율

Threshold \ TYPE	T = 2.5	T = 2.0	T = 1.7
TYPE1	0%	0%	0%
TYPE2	0%	9.23%	16.43%
TYPE3	12.47%	26.68%	33.58%

6. 결론

본 논문에서는 이미지의 주요 오브젝트에서 추출된 곡선과 이러한 곡선에서 얻은 퓨리에 계수를 이용하여 이미지를 검색하는 시스템을 제안하였다. 원본 곡선과 퓨리에 계수를 얻은 후 복원된 곡선과의 차이를 줄이기 위해서 Binary Range Reduction (BRR) 알고리즘을 제안하였다. BRR 알고리즘을 사용하였을 경우, 사용하지 않았을 때보다 원본 곡선과 복원된 곡선과의 차이가 줄어들었고, 검색률 또한 높아졌다.

참고 문헌

1. Y. Nam, E. Hwang, A Shape-Based Retrieval Scheme for Leaf Image. LNCS(PCM2005), Vol. 3767, pp.876-887, Nov. 2005
2. Hiroataka Sonobe, Saeko Takagi, and Fujiichi Yoshimoto, Mobile Computing System for Fish Image Retrieval, in Proc. of International Workshop on Advanced Image Technology (IWAIT) 2004 (poster session), pp. 33-37, Singapore, January 2004.
3. Aibing Rao, Srihari. R.K, Zhongfei Zhang, Spatial Color Histograms for Content-Based Image Retrieval, Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on 9-11 Nov. 1999. pp.183-186
4. Kuo. W.-J, Chang. R.-F, Approximating the statistical distribution of color histogram for content-based image retrieval, Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on Volume 6, 5-9 June 2000 Vol. 4, pp.2007 - 2010
5. Sklansky, Chazin et al.: Minimum perimeter polygons of digitized silhouettes. (1972).

6. Hwei-Jen Lin, Yang-Ta Kao, Shwu-Huey Yen, and Chia-Jen Wang, A Study of shape-Based Image Retrieval, Proceedings of the 24th International Conference on Distributed Computing Systems Workshops(ICDCSW' 04)
7. Zhiyong Wang, Zheru Chi, Dagan Feng, Qing Wang, Leaf Image Retrieval with Shape Features. Lecture Note in Computer Science, Springer-Verlog, Vol.1929.(2000) 477-487.
8. Y. Nam, E. Hwang, A Shape-Based Retrieval Scheme for Leaf Image. Lecture Notes in Computer Science, Springer-Verlog, Vol. 3767, pp.876-887, Nov. 2005.
9. C Im, H Nishida, TL Kunii, Recognizing Plant Species by Normalized Leaf Shapes, Vision Interface ' 99, Trois-Rivières, Canada, 19-21 May, 397-404.
10. 이창복, 대한식물도감, 향문사, 서울, 1982
11. Rajendra Bhatia, Fourier Series. ISBN-0883857405, The Mathematical Association of America, (2004).
12. The MathWorks - MATLAB and Simulink for Technical Computing <http://www.mathworks.com>