영역 추출을 위한 Hough 변환 기반 에지 검출과 영역 확장을 통합한 방법

(A Combined Hough Transform based Edge Detection and Region Growing Method for Region Extraction)

Nguyen, N.T.B.[†] 김 용 권[†] 정 진 완[††] 이 석 룡[†††] 김 덕 환[††††]
(Nguyen, N.T.B.) (Yongkwon Kim) (Chin-Wan Chung) (Seok-Lyong Lee) (Deok-Hwan Kim)

요 약 CBIR(Content-based Image Retrieval) 시스템의 질의 처리에 사용되는 모양 특징은 크게 경계 기반과 영역 기반 등 두 가지로 나눌 수 있다. 경계기반 특징은 간단하지만 영역 기반 특징에 비해 효과적이지 않다. 영역 기반 모양 특징을 사용하는 대부분의 시스템은 먼저 영역을 추출해야 한다. 하지만 기존의 영역 기반 시스템들은 구현이 복잡하고, 특히 정확한 영역 추출이 어려우며 영역 간의 위치적인 관계가 거리 모델(distance model)에 반영되어 있지 않다.

본 논문에서는 Canny 에지 검출과 Hough 변환에 기반하여 목표 내부의 에지를 검출하고, 이와 함께 영역확장을 이용하여 목표 물체 내부의 영역을 정확히 추출할 수 있는 방법을 제안하였다. 또한 영역 간의 인접 관계를 이용한 수정된 IRM(Integrated Region Matching) 기법을 제안하였다. 이는 모양 특징을 이용한 유사성 검색에서 영상 간의 거리 모델로서 사용된다. 그리고 실험을 통해 수정된 IRM 기법과 우리의 영역 추출 기법이 효과적임을 보였다. 실험 결과는 새로운 영역 추출 방법이 기존의 다른 방법보다 훨씬 우수함을 보여준다.

키워드 : CBIR, 영역 추출, IRM, shape feature

Abstract Shape features in a content-based image retrieval (CBIR) system are divided into two classes: contour-based and region-based. Contour-based shape features are simple but they are not as efficient as region-based shape features. Most systems using the region-based shape feature have to extract the region first. The prior works on region-based systems still have shortcomings. They are complex to implement, particularly with respect to region extraction, and do not sufficiently use the spatial relationship between regions in the distance model.

In this paper, a region extraction method that is the combination of an edge-based method and a region growing method is proposed to accurately extract regions inside an object. Edges inside an object are accurately detected based on the Canny edge detector and the Hough transform. And the modified Integrated Region Matching (IRM) scheme which includes the adjacency relationship of regions is also proposed. It is used to compute the distance between images for the similarity search using shape features. The experimental results show the effectiveness of our region extraction method as well as the modified IRM. In comparison with other works, it is shown that the new region extraction method outperforms others.

Key words : content-based image retrieval, region extraction, IRM, shape features

· 본 연구는 한국과학기술원 영상정보센터를 통한 방위사업청과 국방과학연구소의 연구비 지원으로 수행되었습니다(계약번호 UD070007AD).

† 학생회원 : KAIST 전자전산학부 전산학전공
nguyen@islab.kaist.ac.kr
yongkwon@islab.kaist.ac.kr

†† 종신회원 : KAIST 전자전산학부 전산학전공 교수
chungcw@kaist.edu

††† 종신회원 : 한국외국어대학교 산업경영공학과 교수
sllee@hufs.ac.kr

†††† 정 회 원 : 인하대학교 전자공학과 교수
deokhwan@inha.ac.kr

논문접수 : 2008년 10월 17일
심사완료 : 2009년 4월 6일

정보과학회논문지: 데이타베이스 제36권 제4호(2009.8)

1. Introduction

Recent years have witnessed the ever increasing amount of digital images, as a result the demand for efficient organizing and fast accessing to their content has emerged as a critical requirement. The research on content-based image retrieval (CBIR) has been active since the early 1990s.

In a typical CBIR system, the content of an image is characterized by low-level visual features. The features such as color, texture, shape are most common low-level visual features, and known as primitive features. Although color and texture are significant features for retrieval purposes, they lose their original ability in some situations where the image is a rough sketch with only black and white lines, or the color and texture of images are not the discriminating factors. In these cases, the shape features of images and the method of extracting these features become critical for the effective retrieval.

The shape is an important feature for image representation. The shape representation techniques are broadly divided into contour based and region based shape methods. Contour based shape techniques only exploit shape boundary information. In contrast to contour based methods, region based shape techniques deal with shapes with interior content. It is generally agreed that image retrieval using features extracted from not only the contour pixels but also the interior pixels is more rational and desirable, especially for complex images.

A critical part of a system using the region based shape feature (a region-based system) is the region extraction procedure. Various kinds of low-level features such as intensity, color, and texture have been employed to obtain the accurate shape of desired regions. However, extracting regions from an image is a difficult problem. An intrinsic problem is that objects of interest are not homogeneous with respect to these low-level features. Thus it is very hard to acquire correct regions by utilizing the conventional segmentation algorithm. An accurate region extraction method is necessary because of its important role in the region-based system.

Significant edges in an image are also used to represent the shape attribute, especially in the cases where the color features are not distinctive. A very popular edge feature is the edge histogram introduced as a classification tool in the work by [1]. It computes edge orientations from a basis filter and uses the Canny Edge Detector to determine significant edges. Edge orientations are then put into a histogram of chosen dimensions. The edge histogram is an interesting way of capturing the general shape information of the image. Although this basic feature performs quite well, it suffers from several drawbacks. Firstly, it is not rotation invariant. Secondly, it preserves only global information about the direction of edges in the image. It is possible to have two images with totally different edge patterns but have the same histogram of edge directions. A different approach of using edge is proposed in [2]. After extracting the edge information, the Hough transform is applied to obtain a set of line segments. The histogram (on the angle of line segments and on the length of line segments) and the co-occurrence matrix built from these line segments are considered as the features of the image. However, this approach has a high complexity.

To use the edge feature in a more efficient way as well as to obtain an efficient method for separating desired regions, a combination of the edges and the region growing method is proposed to extract the regions inside an object. In this approach, the edge is considered as the boundary of regions. From these boundaries of regions, the region growing algorithm is applied to extract regions. The shape feature values of regions using existing shape features are extracted and the IRM model distance [3] with some proposed modifications is used to compute the distance between images.

The contributions of the paper are as follows:

• *Propose an effective way to use the edge feature as well as a method for extracting regions inside an object.* The method gives a new combination of edge-based and region-based techniques in extracting regions. After detecting edges in an image by using the Canny Edge Detector, the Hough transform is applied to overlap small size windows (sliding window) to detect lines which are considered as the boundaries of regions in an object.

The region growing method will detect these regions as well as their corresponding adjacent regions. As a result, the precision of image retrieval is increased by 15% and the recall is increased by 10.4% compared with the best precision and the best recall of 3 recent methods for extracting regions inside an object.

• *Improve the accuracy of the IRM scheme for calculating distance between images based on their set of regions.* The modified IRM incorporates the neighbours of regions for calculating the distance between regions. By this way, the modified IRM improves the accuracy of computing the distance between two images and, as a result, improves the accuracy of retrieval. The modified IRM improved the precision of the image retrieval by 4% and the recall by 3.5% compared with the IRM.

• By combining the above two improvements, the precision of the proposed system is 19% better and the recall is 13.9% better than the best of the 3 recent methods.

The remainder of this paper is organized as follows. Section 2 gives an overview of the research background by providing a description of CBIR systems, and related works on region extraction. In Section 3 and Section 4, we introduce two important components of a region-based retrieval system: the proposed method for extracting region and the distance model for computing the similarity measure between a query image and a database image. In Section 5 we show its effectiveness with the experimental results. A conclusion is provided in Section 6.

2. Research Background

2.1 CBIR Systems

The purposes of a typical CBIR system are to provide an efficient storage and retrieval of a large collection of images. Figure 1 shows an overview of basic concepts of a typical CBIR system.

For storage, a segmentation method is employed first to identify objects or regions within images. The segmented images are then passed to the feature extractor to extract their actual contents. This data is then passed to a data manager which stores the feature data and the image into a database. The online image retrieval process is similar to the storage. A query image is submitted to the system. This image is segmented and features are extracted. The distance calculator then compares the query image with images in the database using a distance model. According to the distances, the result set of closest matches are then displayed to the user.

2.2 Region-based Image Retrieval System

As the use of digital image information grew rapidly in recent years, it became more important to manage multimedia databases efficiently. CBIR has been widely studied. However, if the CBIR system only uses global properties of the image

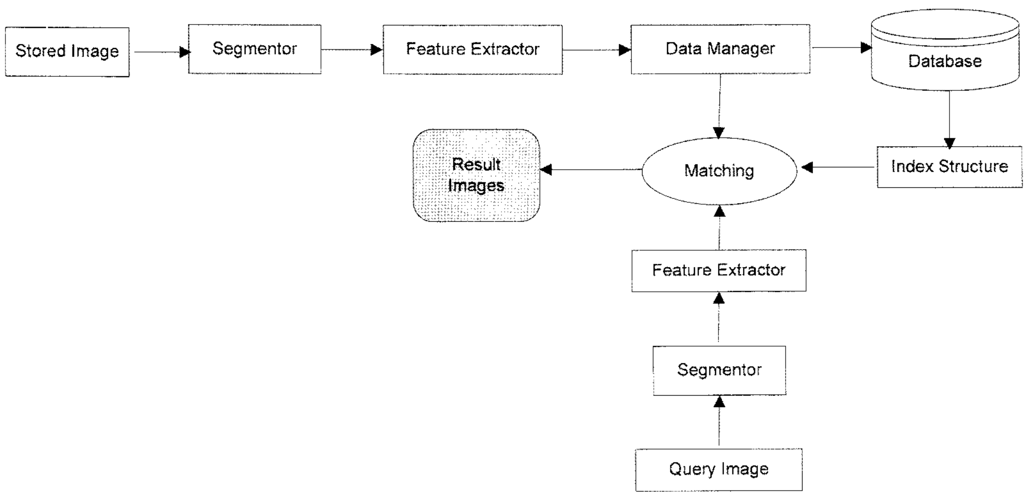


Figure 1 Basic Concepts of a Typical CBIR System

without looking at regions in the image, it may be easy to miss many similar images. This is because many images include some objects and when a user specifies an image for retrieval, the user often pays attention solely on regions of the image. So, most recent CBIR systems have focused on object or region based image retrieval to cope with the cases in which users want to retrieve images based on information of their regions.

Most of region based image retrieval systems basically work in the same way: (1) a segmentation method is applied to decompose an image into regions, (2) some feature vectors are extracted from each region, (3) the set of all feature vectors are organized into a database index. At the query time, feature vectors are extracted from regions of the query image and are then matched against the feature vectors in the index. The region matching scheme may base on either individual regions of images or the combined information from all regions of images.

2.3 Related Work

A critical part of any region based image retrieval system is the region extraction procedure. Although segmentation is a difficult problem, there exists a great deal of prior work for decomposing an image into its regions proposed in the literature [4-6].

Generally, image segmentation techniques can be broadly classified into three main classes: cluster-based, edge-based and region-based methods [7]. Cluster-based methods use thresholding, i.e. a threshold is estimated from a grey level intensity histogram of an image. The valley in the histogram is taken as the threshold. Edge-based methods depend on the assumption that pixel values change rapidly at the boundary between two regions. It is suitable for detecting linear features in the image. Region-based methods are based on the assumption that neighbouring pixels in a region have similar characteristics with respect to color, intensity, and texture. Some region growing methods use the edge as a growth-stopping condition, and growing seeds are selected manually.

2.3.1 Edge based region growing method

The combination of edge detection and region growing in region extraction produces much better results than by either technique alone. The general principle of integration is well accepted in vision. But carrying out the general principle is quite challenging.

In [8], a novel edge based region growing method is presented. The image is first smoothed by an averaging filter to reduce the effect of high frequency noise in the image. The Canny filter then detects the edge points in the smoothed image. Sobel operator is used for obtaining the edge region image, which is then binarised into two kinds of areas. One is the edge region which is defined as the place where the region growing seeds will be selected and the other is the homogenous region. After the edge detection step, the next step is the region growing. Based on the edge points and the edge regions, the seed pixels for region growing are selected with the assumption that the objects to be segmented always appear red (hot) and the background appears black (cold). And finally, from the seed pixels, the region growing is employed based on the fact that the grey levels of the hot seeds are lower than the pixels not far away from the edge region in the hot object and the grey levels of the cold seeds are higher than the pixels not far away from the edge region of the cold background. Thus the seeds grow into their respective regions to give a segmented binary image, which is the final output image.

This method gives a new combination of edge detection and region growing. However, it still has some problems. If the object contains high intensity changes inside its boundary, the hot and cold regions may form inside the object, not giving proper segmentation. If the image contains shadows, the hotness degree cannot be calculated properly, because in the shadowy region, the intensity of the object becomes colder than the background itself. This can give problems in segmentation.

2.3.2 Watershed segmentation

Watershed transform [9,10] is a popular division tool based on mathematical morphology and has been widely used in many fields of image segmentation. It is a region-based method based on the growth of the region from a seed pixel. In watershed transform, an image is interpreted as a topographical surface in which the altitude of every point is generally equal to the gradient value of corresponding pixel. In such a topographic interpretation, there are 3 types of points: points belonging to a regional minimum, points at which a drop of water would fall to a single minimum (the catchment-basin or watershed of that minimum), and points at which a drop of water would be equally likely to fall to more than one minimum (the divide lines or watershed lines). Figure 2 gives an example of these types of pixels.

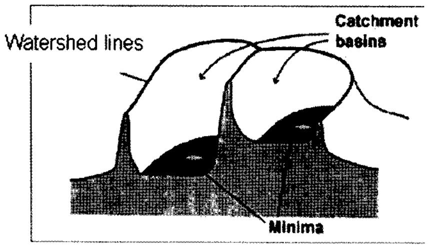


Figure 2 Three Types of Pixels in the Watershed Transform

The objective of watershed segmentation algorithm is to find watershed lines, which are considered as the boundaries of result segments. It supposes that a hole is punched in each regional minimum and the entire topography is flooded from below by letting water rise through the holes at a uniform rate. When rising water in distinct catchment basins is about the merge, a dam is built to prevent merging. These dam boundaries correspond to watershed lines. The local gradient minima are detected and used as seeds for region growing. However, the performance of the watershed-based image segmentation method depends largely on the algorithm used to compute the gradient. It is sensitive to gradient noises, which yield many catchment basins, resulting in over-segmentation.

2.3.3 The hybrid of mean shift filtering and efficient graph-based clustering

The mean shift segmentation and the efficient graph-based segmentation are considered to be very efficient segmentation techniques and have become widely-used in the vision community. The mean shift segmentation considers a joint domain representation that includes spatial and range (color) domains. Under this feature space, the mean shift filtering analyzes the probability density functions (pdf) underlying the image data by using kernel density estimation to find modes of these underlying pdf(s). After mean shift filtering, each data point has been replaced by its corresponding mode. The clustering is performed to cluster these modes into segments.

The efficient graph-based segmentation is another method of performing clustering in feature space. A graph G = (V, E) is defined for the image where each pixel is a vertex and the edge set is constructed by connecting pairs of neighbour pixels. Initially, each vertex is a region. Two regions are merged if there is no boundary between them. There is a boundary between two regions C_1 and C_2 if the difference between them is large relative to a threshold function. The difference between two regions is the defined as the smallest edge weight connecting them. The threshold function is the function of the largest edge weight in the minimum spanning tree of each region and a term of a constant k and the size of each region.

Although the mean shift algorithm and the efficient graph-based segmentation algorithm are efficient and widely used in the vision community, [11] showed that we can combine these two methods to give better results than either method alone. More specifically, the mean shift procedure is applied to filter the image and the efficient graph-based clustering is used then to give the final segmentation. However, this combination still suffers somewhat from the sensitivity to the parameters of mean shift and the graph-based clustering. In addition, choosing the right parameters for different image domain is also a big problem.

3. Region Extraction Method

This section presents the proposed method for extracting regions inside an object. It is the combination of the edge-based method and the region-based method. The edge-based part detects from the edge image the lines which are considered as boundaries of regions. In the region-based part, the region growing algorithm is applied to detect regions based on the boundaries from the first part.

3.1 Canny Edge Detector

An edge is defined by the magnitude of a convolution result with a spatial mask in the image. Neighbour pixels around the current pixel are convoluted with the spatial mask to measure how much it contains the edge information. According to this magnitude, the current pixel is classified as either an edge or non-edge pixel. With this definition, an edge can be considered as the boundary of two or more adjacent regions having distinct brightness or color values.

The Canny Edge Detector [12] is known as the optimal edge detector. In that algorithm, a list of criteria is followed to improve the quality of detecting edges. The first and most obvious criterion is the low error rate. It is important that edges occurring in the image should not be missed and that there be **no responses** to non-edges. The second criterion is that the edge points are well localized. The third criterion is to have only one response to a single edge. To fulfill these criteria, the edge detection process includes the following stages:

- Image Smoothing: the image data is smoothed by a two-dimensional Gaussian function.
- Differentiation: the Sobel operator can be applied to obtain x and y gradients. From them, the magnitude and angle of edges are calculated.
- Non-maximum Suppression: having found the rate of intensity change at each point, edges must now be placed at the points of maxima.
- Edge Thresholding: two thresholds T_1, T_2 ($T_1 > T_2$) are used. Any edge point with the magnitude greater than T_1 is marked as the correct edge. Then, any pixel that is connected to this edge pixel and has a value greater than T_2 is also selected as the correct edge pixel.

3.2 Detecting Straight Line from Edge Points

We can obtain the edge map with the Canny Edge Detector. However, we have not found edge segments, but just only edge points. If an image is noisy or if its attribute differs by only a small amount between regions, edge detection may produce spurious and broken edges. An edge linking technique needs to be employed to bridge short gaps in such a region boundary. This part describes an efficient method for finding straight lines in the edge map.

3.2.1 Hough Transform

The Hough Transform (HT) is an established technique, which evidences a shape by mapping edge points into a parameter space. In the original HT for extracting straight lines [13], each edge point is transformed into a straight line in the slope-intercept parameter space (m, b). Through each edge point (x_0, y_0) in the image, an infinite number of lines may be drawn, which satisfy the equation: $y_0 = mx_0 + b$. In the parameter space (m, b), the equation for a line defines a selected edge point in the image. Another edge point (x_k, y_k) will also be transformed to a line in the (m, b) space, and all collinear edge points will across the same point in parameter space. The HT converts a rather difficult problem of detecting collinear points in the image to a less challenging problem of detecting peaks in the slope-intercept parameter space.

However, using slope-intercept parameters could make application complicated since both parameters are unbounded. As lines get more and more vertical, the magnitudes of m and b grow towards infinity. The modified HT [14] introduced the use of the normal parameterization of lines, defined by the length r and the orientation θ of the normal vector to the line relative to the origin of the image:

$r = x\cos\theta + y\sin\theta$ where

$0 \le \theta < 360°$

$0 \le r$

3.2.2 Sliding Window Hough for Detecting Line Segment

Gradient images can be described by their spatial locations (x, y), magnitudes and directions. We assume that each gradient vector represents a line which is parallel to the direction of that vector and passes through the gradient's spatial location. This directional line can be parameterized by two parameters (r, θ) in the Hough space as shown in the Figure 3.

To detect line segments from the edge magnitude image (image of magnitudes of edges) and the edge direction image (image of directions of edges), a combination of sliding window technique and the Hough transform algorithm is used. Sliding the window through the entire image (the window is moved by half of its size, both horizontally and vertically), the Hough transform algorithm is applied

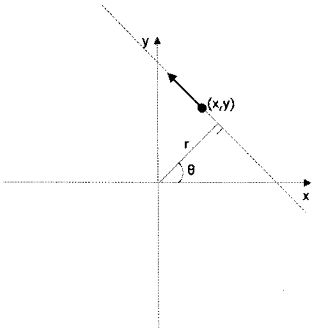


Figure 3 A Representation of the Gradient Vector Using r-θ Variables

within each window to detect line segments which pass through this window. The origin of coordinate of window is chosen as the center of window. The parameter space limits for the window are:

$0° \leq \theta < 360°$

$0 \leq r \leq R_{max}/2$

where $R_{max} = \sqrt{window\_width^2 + window\_height^2}$

The details of the algorithm for each window are as follows:

- Step 1: Quantizing two Hough parameters with suitable intervals Δr, $\Delta\theta$ and then initializing all cells of accumulator matrix to zero (accumulator is a two dimensional matrix, r and θ of the parameter space).
- Step 2: Determining the polar parameters r and θ of gradient vectors within the window. For each gradient vector, the corresponding cell is increased to 1.
- Step 3: Finding m peaks in the accumulator. r and θ (in the coordinate of the current window) of these peaks are parameters of the detected line segments.

Splitting the image into small size windows reduces the size of the accumulator matrix and increases the accuracy of detecting line segments. However, as the result of applying Hough transform locally to sub-images instead of the entire image, a real line can be divided into some line segments. Therefore, we need to group the proximate line segments to obtain the real lines.

3.2.3 Grouping Line Segments

After sliding the window over the entire image, we can get a set of short line segments. Because these line segments are detected from different windows, their parameters r and θ are in different coordinates. To group them together, it is necessary to transform their parameters from the window coordinate to a common coordinate first. We choose the image coordinate whose origin is the center of image as the common coordinate.

- **Transform the equation of line from window coordinate to image coordinate**

Consider line segment ls with two Hough parameters r and θ in its window. The equation of this line segment in the coordinate of its window is:

$$r = x\cos\theta + y\sin\theta$$

Assume that the coordinate of the window's center in the image coordinate is (x_0, y_0), the equation of this line segment in the image coordinate is:

$$r = (x - x_0)\cos\theta + (y - y_0)\sin\theta \quad (1)$$

Let $$r' = r + x_0\cos\theta + y_0\sin\theta \quad (2)$$

(1) becomes: $r' = x\cos\theta + y\sin\theta$ (3)

In the equation (3) of ls in the new coordinate, θ is unchanged, but the value of r' may be positive or negative depending on (x_0, y_0), r and θ. Therefore, we need to adjust so that r' and θ in (3) are in the range of r and θ in section 3.2.1 ($r' > 0$ and $0° \leq \theta < 360°$).

```
if r' > 0 then
      do nothing
      else
      r' := - r'
      // - r' = - (xcosθ + ysinθ) = -xcosθ – ysinθ
      if (0 ≤ θ < 180) then
θ := θ + 180
      else
θ := θ – 180
      end if
      end if
```

After transforming the Hough parameters of line segments to the common coordinate, we obtain a set of line segments whose Hough parameters (r, θ) are in the same coordinate.

- **The proximity of two line segments**

Consider a line segments ls_i with two Hough parameters (r_i, θ_i). The line ls_j with two Hough parameters (r_j, θ_j) is said to be proximate to ls_i if they fulfill three below conditions:

- ○ r_j of ls_j is closer than Δr (pixels) r_i of ls_i.
- ○ θ_j of ls_j is within $\Delta\theta$ (degrees) of θ_i of ls_i.
- ○ ls_i is not far from ls_j

Checking the proximity on r is simple. We just need to compare $|r_i - r_j|$ and Δr.

Checking the proximity on θ includes the following cases of positions of ls_i and ls_j:

- ○ They are in the same quadrant or two adjacent quadrants
- ○ They are in the first quadrant and the fourth quadrant
- ○ They are in opposite quadrants

In the third case, we need to check the condition of r again. In this case, instead of checking $|r_i - r_j|$, we need to check $|r_i + r_j|$ because two line segments are in opposite quadrants.

```
diffTheta = |θi – θj|                //first case
if diffTheta > 180 then              // second case
        diffTheta = 360 - diffTheta
        end if
   //third case
        if (θi < θj) then
        diffTheta1 = |( θi +180) – θj|
        else
        diffTheta1 = |( θj +180) – θi|
        end if
        if (diffTheta ≤ Δθ) then
        return true
        else
        if (diffTheta1 ≤ Δθ)) then
if (ri + rj ≤ Δr) then
return true
end if
end if
        end if
        return false
```

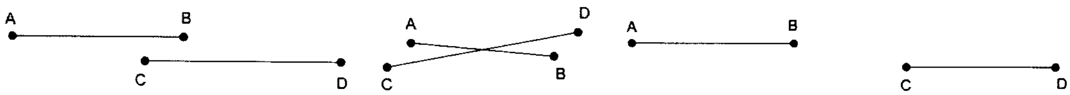
Checking the condition of two line segments closed each other: when ls_i and ls_j satisfy two first conditions, their placement can be in Figure 4. The placement of ls_i and ls_j in Figure 4(a) and Figure 4(b) is considered as satisfying this condition. In Figure 4(c): although they are proximate in r and Θ, these two line segments are far from each other. Therefore, they cannot be considered as proximate. In other words, if the distance between these two line segments is smaller than a specified threshold Δd, they are considered as satisfying the condition of two line segments closed each other. The distance between two line segments is defined as below:

- ○ If they intersect each other (the intersection point is in both of line segments like in Figure 4(b)), the distance between them is 0.
- ○ Otherwise, the distance between them is the minimum shortest distance from an endpoint of one line segment to the other line segment. For example, in Figure 4(a), the distance between AB and CD is the shortest distance from C to AB, in Figure 4(c), the distance between AB and CD is shortest distance from C to AB (=CB) or from B to CD (=BC).

• **Merge short line segments into longer lines**

If two line segments are proximate each other with respect to three above conditions, they will be merged into a new line segment. Its Hough parameters are calculated from those of proximate line segments, and its two endpoints are detected as the longest line created from two of their four end-points. We continue to merge the set of line segments into the longer lines in this way until there are no two proximate line segments.

We will consider these lines as boundaries of regions inside the object. To do that, each line must be closed at two end points. In other words, each line must be extended/narrowed so that it has two intersection points with other lines or with the boundary of object. The algorithm to extend/narrow



(a) (b) (c)

Figure 4 The Placement of Two Line Segments (line by line) all these lines is as follow:

- ○ For each line, get the number of intersection points between this line and the boundary of object and between this line and all other lines (just count the intersection points which are between two end points of lines)
- ○ Choose the line with the maximum number of intersection points to extend/narrow. To update each of two endpoints of this line:
  - · Calculate the distances from it to all intersection points.
  - · Update the endpoint to the intersection point with the minimum distance.
  - · If two updated end points are far from two original end points, this line is considered as a noise line and is removed from the set of lines.

3.3 Extracting Regions from Boundaries

Region growing is one of the most simple and popular algorithms for region-based segmentation. Typically, to extract one region, the first step is choosing a starting point or a seed pixel. The region then grows by adding neighbouring pixels that are similar, according to a certain homogeneity criterion, increasing the size of the region step-by-step. The homogeneity criterion is a function which determines whether or not a pixel belongs to the growing region. The decision to merge is generally based only on the contrast between the current pixel and the region. However, it is not easy to decide when this difference is small (or large) enough to make a decision.

The combination of the region growing and the set of lines obtained from Hough transform, which can be considered as borders between regions, gives a new criterion for detecting regions. It is just that whether the current pixel meets the boundaries or not. With this criterion, the region detection process is much easier and the result is more accurate. The process of detecting regions is started by choosing an arbitrary pixel belonging to the object (not on the boundaries) as a seed pixel of the current region. From the seed pixel, we flood to 8 directions and put its 8 neighbour pixels to the queue. We continue flooding pixels in the queue until meeting the boundaries of the current region. The growth of the region finishes when the queue is empty.

The detail algorithm is presented as follows. This algorithm allows detecting both regions and their corresponding neighbour regions.

- **Input of algorithm:**
  - Object and the boundary of the object.
  - A set of lines inside the object which are considered as the boundaries of the regions. Each line in this set is labelled with a unique label.
- **Output of algorithm:**
  - A set of regions inside the image object. Each region is labelled with a unique number. (Every pixel in each region is assigned to this unique number).
- **Variables, Functions and Notations:**
  - *current_region*: the label of the current region.
  - im_i: the input image with object, its boundary and lines inside the object (these lines are considered as the boundaries of regions).
  - im_o: the output with regions labelled with unique numbers.
  - im_w: the work image is used to mark whether a pixel is explored (1) or not (0).
  - $N_8(p)$: eight-connected neighbour pixels of p.
  - q: a queue for storing pixels belonging the current region during the breadth-first propagation.
  - These are the basic functions for queue:
    fifo_add(p): put the pixel p into the queue q.
    fifo_first(): return the pixel which is at the beginning of the queue q.
    fifo_empty(): returns true if the queue q is empty and false otherwise.
- **Initializations:**
  - Each pixel of im_o is assigned to 0.
  - Each pixel of im_w is assigned to 0.
  - *current_region* $\leftarrow$ 0.

After scanning all pixels in the object, we have the set of regions. Among them, there will be some regions whose areas are small (smaller than a **threshold**). These small regions are usually created because of noise during creating boundary lines (extending or narrowing the boundary lines). In addition, these small regions are usually not significant in further processing. Therefore, we should merge the small regions into the bigger adjacent regions.

```
Algorithm RegionGrowing
for each pixel p in the image im_i
    if p belongs to the object (not in boundary of object, not in any line) and p is not explored (im_x(p) = 0)
        im_x(p) := 1                              // mark p as explored
        current_region := current_region + 1      // start to detect new region
        fifo_add(p)                               // push p to queue
        im_o(p) := current_region                 // p belong to the current_region
        while (fifo_empty() = false)
          p := fifo_first()                       // pop the first element of queue to p
          switch(p):
            case p belongs to a line line_i
              store line_i as a boundary of current_region
              for each pixel p' ∈ N_8(p)
                if p' belongs to boundary of object
                  store p' as intersection of line_i and object
                end if
                if p' belongs to other line line_j
                  store line_j as a boundary of current_region
                  mark the pair (line_i, line_j) as having intersection point
                end if
              end for
            case p belongs to the object (not any line)
              for each pixel p' ∈ N_8(p)
                if p' is not explored (im_x(p') = 0)
                  im_x(p') := 1                   // mark p' as explored
                  im_o(p') := current_region      // p' belongs to current_region
                  fifo_add(p')                    // push p' to queue
                end if
              end for
          end switch
        end while
        detect all line segments (two end points) as boundaries of current_region
    end if
end for
```

4. Distance Model

In this section, the distance model for determining the distance between a query image and a image stored in the database is presented. Based on the IRM, the Gaussian normalization is applied and the incorporation of neighbour regions in distance calculation is introduced.

4.1 Integrated Region Matching (IRM) Model

IRM is a similarity measure developed to evaluate overall similarity between images that incorporated properties of all regions in the image [3]. To compute the similarity between 2 region sets $R_1 = \{r_1, \cdots, r_m\}$ and $R_2 = \{r'_1, \cdots, r'_n\}$ (r_i or r'_i is the description of region i), we first match all pairs region-to-region in R_1, R_2. A matching between two regions r_i and r'_j is assigned with a significance credit $s_{i,j}$ which indicates the importance of the matching for determining similarity between images. The matrix $S = \{s_{i,j}\}$, $1 \le i \le m$, $1 \le j \le n$ is referred to as the significance matrix.

The distance between the two region sets is defined as: $d_{IRM}(R_1, R_2) = \sum_{i,j} s_{i,j} d_{i,j}$ where $d_{i,j}$: the distance between r_i and r'_j; $d_{i,j} = \sum_{k=1}^{n} w_k f_k(r_i, r'_j)$: n is the number of features, w_k is the feature weight of feature k and $f_k(r_i, r'_j)$ is Euclidian distance function between feature k of region r_i and feature k of region r'.

To choose the significance matrix S, two properties are put on $s_{i,j}$ so that the matching yields good results. The first property is the fulfillment of significance. Assuming that the significance of r_i in *Image1* is p_i, and that of r'_j in *Image2* is p'_j, we require that:

$$\sum_{j=1}^{n} s_{i,j} = p_i, 1 \le i \le m, \sum_{i=1}^{m} s_{i,j} = p'_j, 1 \le ji \le n$$

For normalization, we have: $\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} p'_j = 1$.

In the second property, we require that the matching links the most similar regions at the highest priority. Following the "most similar highest priority" principle, the IRM algorithm attempts to fulfill the significance credits of regions by assigning as much significance as possible to the link between two regions with minimum distance. Initially, assume that $d_{i',j'}$ is the minimum distance, we set $s_{i',j'} = \min(p_{i'}, p'_{j'})$. Without loss of generality, assume $p_{i'}$

$\leq p'_{j'}$. Then $s_{i'j} = 0$, for j $\neq$ j' since the link between region i' and j' has filled the significance of region i'. The significance credit left for region j' is reduced to $p'_{j'} - p_{i'}$. The values of $s_{i',j'}$ and $s_{i'j}$ are assigned. The matching problem now becomes: solving $s_{i,j}$ (i $\neq$ i') under these 4 constraints:

$$\sum_{j=1}^{n} s_{i,j} = p_i,\ 1 \leq i \leq m,\ i \neq i'$$

$$\sum_{1 \leq i \leq m, i \neq i'}^{m} s_{i,j} = p'_j,\ 1 \leq j \leq n,\ j \neq j'$$

$$\sum_{1 \leq i \leq m, i \neq i'} s_{i,j} = p'_{j'} - p_{i'}$$

$$s_{i,j} \geq 0; 1 \leq i \leq m, 1 \leq j \leq n$$

We apply the previous procedure again. The iteration stops when all the significance credits p_i and p'_j have been assigned. The problem now is how to choose the value for p_i. The value of p_i is chosen to reflect the significance of region i in the image. If we assume that every region is equally important, then $p_i = \frac{1}{m}$, where m is the number of regions. Another choice of p_i is the percentage of the image covered by region i based on the view that important regions in an image tend to occupy larger areas.

4.2 Gaussian Normalization

Since different features can generate different ranges of value of similarity, a normalization method should be applied to each similarity computation. The distance between regions in the query image and the database image for each feature will be normalized to the new range **[0,1]** by using the Gaussian normalization [15].

Assuming the sequence $d_{i,j}(r_i,\ r'_j)$ to be a Gaussian sequence v_k, we compute the mean μ_k and the standard deviation σ_k of this sequence. We then normalize the original sequence to a N(0,1) sequence as follows: $v_k = \frac{v_k - \mu_k}{\sigma_k}$ After this normalization, the probability of an entry's value being in the range of [-1,1] is 68%. If we use $3\sigma_k$ in the denominator, according to 3-σ rule, the probability of an entry's value being in the range of [-1,1] is approximately 99%. An additional shift $v_k = \frac{v_k + 1}{2}$ will guarantee that 99% of similarity values are within [0,1]. The advantage of this normalization process over the simple max-min normalization is that the presence of a few abnormally large or small values does not bias the importance of a component of the feature vector in computing the similarity between vectors.

4.3 Incorporation of Neighbour Regions to Compute the Distance between Two Regions

Adjacency is an important relationship in the description of regions. Including the information of the neighbour regions makes the similarity of regions more precise. It is possible that two regions of two images have similar shape features but their corresponding neighbour regions are not similar. In this case, these two regions may not similar. Apart from low-level descriptions, it is necessary to include the information about spatial context of regions as well, since it is the way to discriminate between images with similar visual appearance.

The distance model between two regions r_i, r'_j when incorporating their neighbours:

$$d_{ij} = \frac{w_{ij}}{w_{ij} + w'_{ij}} d_{ij} + \frac{w'_{ij}}{w_{ij} + w'_{ij}} d$$

(neighbours of r_i, neighbours of r'_j)

- w_{ij}, w'_{ij} are the weights of d_{ij} and d(neighbours of r_i, neighbours of r'_j), respectively. These weights are used to adjust the effect of two regions r_i, r'_j and their neighbours.
- d(neighbours of r_i, neighbours of r'_j) is the distance of corresponding neighbours of r_i and r'_j. In short, we call it as d(neighbours). This distance is calculated as the average of all distances of respective pairs of neighbour regions (of r_i and r'_j). These pairs are found based on the ascending orders of angles created by the x-axis and the vector whose initial point is the center of considered region and terminated point is its neighbour region. If one neighbour region does not have the respective pair, the distance is calculated as the maximum distance between any two regions of two images.

Choosing good values for w_{ij} and w'_{ij} is very important in this distance model. Assume that the important regions in an object tend to occupy larger areas. With this assumption, the importance of a region in an object is calculated as the percentage of the object covered by that region. Followings are the roles of two regions and their neighbour regions in computing distance between them:

- r_i and r'_j play a more importance role in computing distance between them than their neighbours, so we should choose $w_{ij} > w'_{ij}$. Here we choose $w_{ij} = 1$ and $w'_{ij} < 1$.
- If two regions have small areas, this means their importance in the object is low. In other words, the contribution of matching two these regions for determining the similarity of their objects is small. In this case, their distance should not be increased greatly.

From the role of d(neighbours) and the condition that $w'_{ij} < 1$, we decide to choose w'_{ij} in the above formula as follows:

$$w'_{ij} = (p_i + p_j) / 2$$

p_i, p_j are the importance of the region r_i and r_j in their objects respectively:

p_i = area_of_r_i / area_of_object_to_which_r_i_belongs

p_j = area_of_r'_j / area_of_object_to_which_r'_j_belongs

5. Experimentation

In this section, the experimental results are presented to validate the effectiveness of the proposed method for detecting regions inside an object. In addition, the retrieval performances based on the region-based shape feature values obtained using recent region extraction methods are compared with the performance of the proposed method.

The time complexity of our distance model is related to the numbers of regions of the query image and an image stored in the database. Since the numbers are usually the same or similar, we use k for them. Then, the cost of our distance model is $O(k^2)$. So the total cost to find images which are similar to the query image is $O(n*k^2)$. This is not different to that of the original IRM.

5.1 Experimental Environment and Prototype System

The experimentation was performed on the Windows platform powered by a Pentium4 Dual Core 2.13Ghz CPU using 2GB of RAM. The prototype system is implemented using C++ and the .NET framework. Images and their associated feature data are stored to an Oracle 10g database located remotely. The query image is displayed in the top left corner of the window. The images below are the top-k images retrieved from the database. The images are ranked based on distance.

5.2 Experimental Dataset

The proposed method for detecting regions inside an object can be applied to classes of images which have the lines inside the object. We use a leaf image and a pattern image to test the quality of the region detection. Then, we use the leaf images as the dataset to test the accuracy of the image retrieval. Our image database includes 21 leaf species. For each leaf species, there are 24 different images which are created by scaling to three different sizes and rotating each scale to eight different angles: 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°. The total number of images in the database is 21*24 = 504 images. The leaf images in the database are converted to gray scale images and consist of leaf objects and plain backgrounds'. The boundary of a leaf is easily detected as the border of the white part (background) and the leaf object (gray part).

5.3 Experimental Results

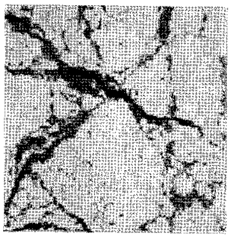
The purpose of our experiments is to compare the effectiveness of the proposed method for detecting regions with recent existing methods for extracting regions inside an object. From the detected regions, the values of region-based shape features are extracted and then used for calculating distance between a query image and an image stored in the database, and the images closest to the query image will be returned.

5.3.1 Effectiveness of the Proposed Method for Extracting Regions

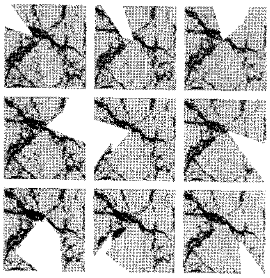
Figure 5 and Figure 6 show the performance result of region detection of the proposed method and the existing methods (the edge based region growing, the watershed method, and the hybrid of mean shift filtering and graph-based clustering method) which are applied to the two images. It is clear that the proposed method is more exact than the other methods.

5.3.2 Effectiveness of the Proposed Method for Image Retrieval Accuracy

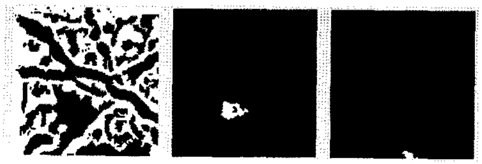
In this section, the effectiveness of image retrieval using values of region-based shape features of regions extracted by the proposed method, the edge



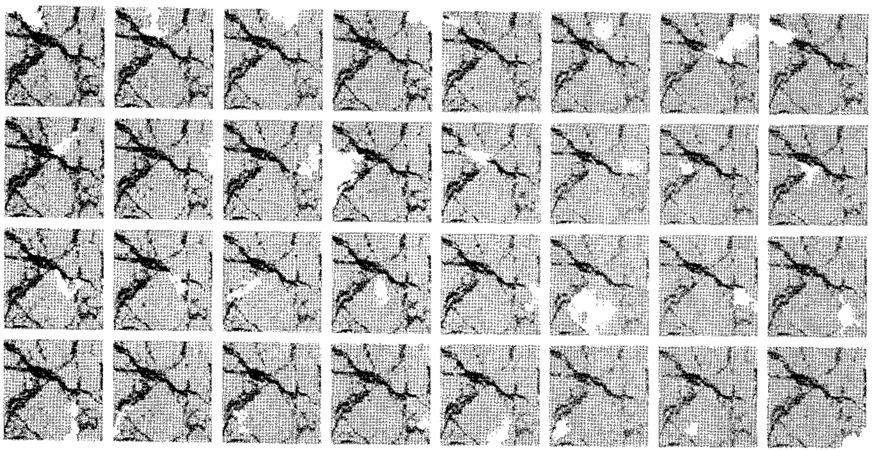
(a) Input image



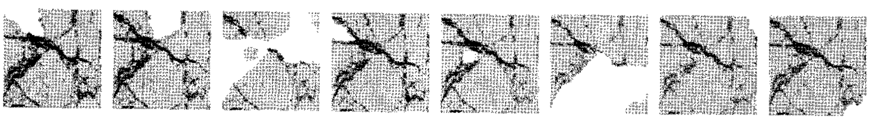
(b) Result of the proposed method



(c) Result of the edge-based region growing method



(d) Result of the hybrid of mean shift filtering and graph-based clustering method



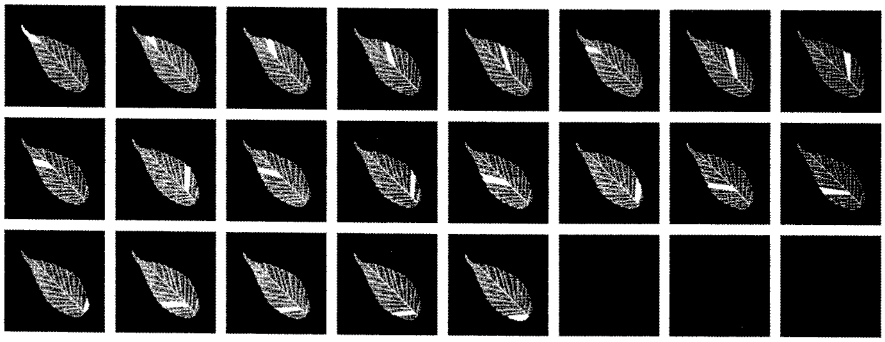
(e) Result of the watershed method

Figure 5 The experimental result of region detection methods for the pattern image

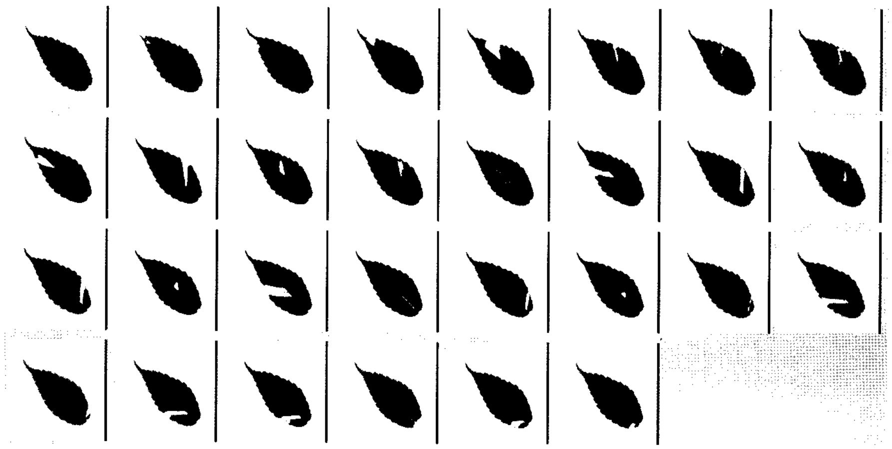
based region growing, the watershed method, and the hybrid of mean shift filtering and graph-based clustering method is presented. The watershed segmentation is mentioned in Section 2.3.2. In order to achieve the meaningful regions, the watershed transformation is performed on the gradient image. The method proposed in [16] is used to produce the gradient image called H-image. After obtaining the initial result by applying watershed transform to H-image, the modified RAG [10] is used to merge

(a) Input image

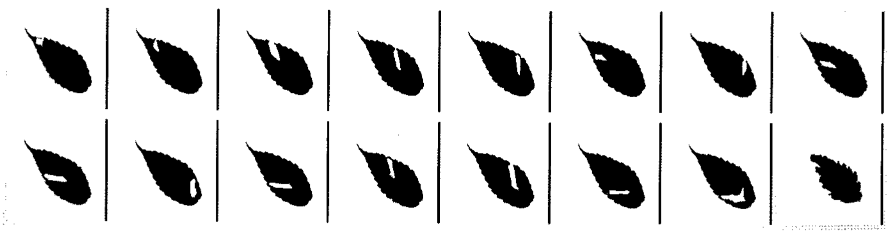
(b) Result of the edge-based region growing method



(c) Result of the proposed method

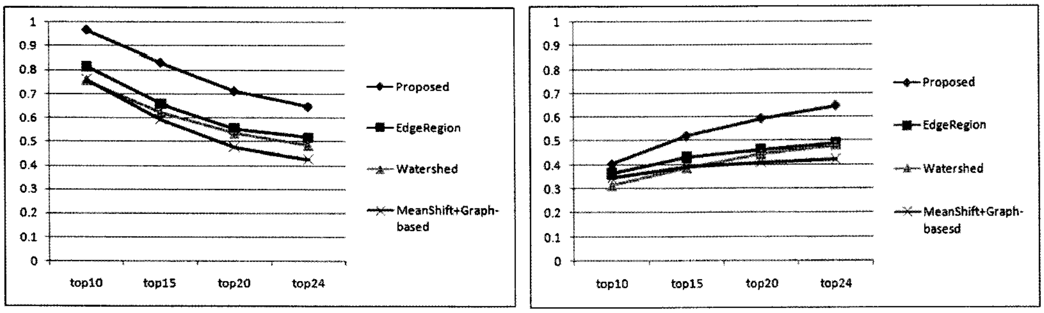


(d) Result of the hybrid of mean shift filtering and graph-based clustering method



(e) Result of the watershed method

Figure 6 The experimental result of region detection methods for the leaf image



(a) Precision (b) Recall

Figure 7 Precision-Recall Comparison between the Proposed Method and other methods

some regions according to a merging cost model to reduce the number of regions.

Figure 7 shows the precision and recall of the retrieval using features of regions extracted by the proposed method (denoted as Proposed), the edge based region growing (denoted as EdgeRegion), the watershed method (denoted as Watershed), and the hybrid of mean shift filtering and graph-based clustering method (denoted as MeanShift+Graph-based). The precision and recall are obtained by getting the average of precisions and recalls of 21 queries whose image queries are chosen from data-set (one query image for each leaf species). The number of images in the returned result is top-k with k = 10, 15, 20, 24.

In the Figure 7, it is clear that the proposed method is more effective than the other methods. The edge based region growing depends largely on the seed pixels, which are detected by the Sobel operator and the Canny edge detector. In the case the boundaries of regions are discontinuous, it cannot fill these gaps and as the result, the detected regions are not correct. The watershed depends largely on the gradient image. In addition, the disadvantage of the watershed method is over-segmentation problem. It is necessary to merge regions; therefore it also depends on the condition to merge. The hybrid of mean shift filtering and graph-based clustering is still sensitive to the parameters of algorithm and also sensitive to the discontinuities of edge points. For the proposed method, when the Hough transform is applied to the edge image, it can bridge short gaps in the edge points. In addition, it is elegant and not sensitive to noise or to

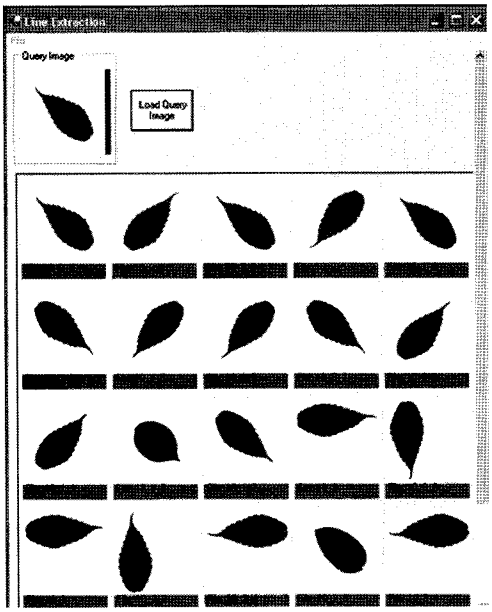
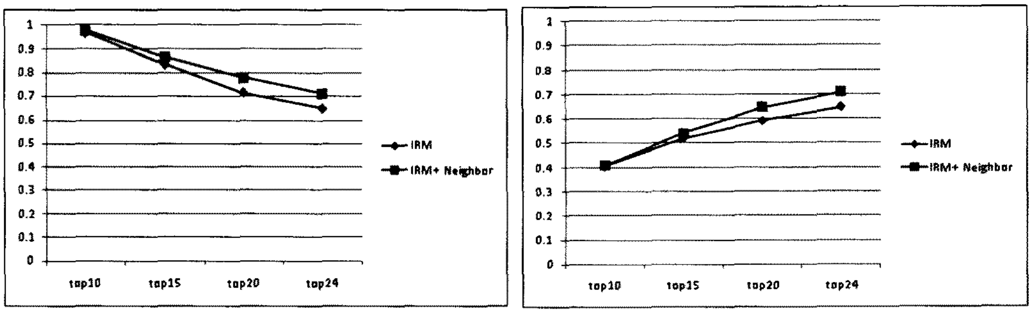


Figure 8 Result of Retrieval

missing edge elements of a prospective line. So the boundaries of regions inside leaf obtained by Hough transform is more exact, and then the regions are detected more exactly. Among the three existing methods, EdgeRegion has the highest precision and recall. By getting the average of differences in the precision between Proposed and EdgeRegion, and the average of differences in the recall between Proposed and EdgeRegion in Figure 8 for top-k with k = 10, 15, 20, 24, the result shows that the precision of image retrieval using the Proposed for detecting regions is absolutely increased by 15% and the recall is absolutely increased by 10.4% compared with using EdgeRegion for detecting



(a) Precision (b) Recall

Figure 9 Precision-Recall Comparison between IRM + Neighbor and IRM

regions. In other words, the precision and recall of Proposed are 30% and 20% relatively better than those of EdgeRegion, respectively.

Figure 8 shows a good result of the proposed method with the query image in the top left corner of the window, and the results are below the query image. The top-k is 20.

5.3.3 Incorporating Neighbour Regions to Compute the Distance between Two Regions

The distance model used to determine the distance between a query image and an image stored in the database is based on the IRM scheme [3]. This scheme is mentioned in the Section 4.1. In this scheme, the distance between two regions is calculated just based on themselves, not using their neighbour regions.

Figure 9 shows the precision and recall of original IRM scheme (denoted as IRM) and the IRM which includes the neighbour regions of two regions when calculating the distance between them (denoted as IRM + Neighbor). By getting the average of differences in the precision between IRM + Neighbor and IRM, and the average of differences in the recall between IRM + Neighbor and IRM for top-k with k = 10, 15, 20, 24 in this figure, the result shows that the modified IRM improved the precision of image retrieval by 4% and the recall by 3.5% compared with the IRM absolutely. In other words, combinating both of the above factors, our proposed method(Proposed region extracting method and the modified IRM) shows 40% relatively better results than the EdgeRegion method which is the best among compared methods in terms of the precision and recall. The low-level features of regions in the image can not describe effectively an image with multiple regions of interest. Incorporating neighbour regions for computing distance between regions gives a more exact similarity measure than just using the low-level features of regions; especially in the case low-level features are shape-based features.

6. Conclusion

In this paper, we proposed a new way to use the edge feature. Instead of using the edge histogram as the edge feature, we considered the edge as the boundary of regions inside an object, detected the regions, and extracted region-based shape features of these regions. In other words, we proposed a new method for extracting regions inside the object, based on the edges of the image. This method gives a new combination of the edge-based method and the region-based method in extracting region. It is divided into two parts. The first part is the edge-based part. After detecting edge points in the image by using the Canny edge detector, the Hough transform is applied to overlapping small windows in the edge image to detect line segments. These lines are considered as the boundaries of regions in the object. The second part is the region-based part which uses the region growing algorithm to detect regions whose boundaries are detected in the first part. From the experimental result, it is shown that the proposed method is better than the edge based region growing method, the watershed method and the hybrid of mean shift filtering and graph-based clustering method for extracting regions.

We have also proposed a new way to incorporate the neighbours of regions for calculating the distance between regions. The proposed scheme assigns the different weights to distance between two regions and to distance between their neighbours so that both two regions and their neighbours have roles in computing distance between two regions. Our experimental result shows that incorporating neighbours of regions improves the effectiveness of IRM.

References

[1] Park, D. K., Jeon, Y. S., and Won, C. S., "Efficient use of local edge histogram descriptor," *Proc. of the 2000 ACM workshops on Multimedia*, pp.51-54, 2000.

[2] Pujol, A., Chen, L., "Line Segment based Edge Feature using Hough Transform," *Proc. of Visualization, Image, and Image Processing (VIIP)*, 2007.

[3] Li, J., Wang, J. Z., and Wiederhold, G., "IRM: Integrated Region Matching for Image Retrieval," *Proc. of ACM Multimedia*, 2000.

[4] Chad, C., Megan, T., Serge, B., Joseph, M. H., and Jitendra, M., "Blobworld: A System for Region-Based Image Indexing and Retrieval," *Proc. of International Conference on Visual Information Systems*, pp.509-516, 1999.

[5] Smith, J. R., and Chang, S. F., "VisualSEEk: A Fully Automated Content-Based Image Query System," *Proceedings of the 4th ACM International Conference on Multimedia*, pp.87-98, 1997.

[6] Ko, B. C., Lee, H. S., and Byun, H., "Region- based Image Retrieval System Using Efficient Feature Description," *Proc. of the 15th International Conference on Pattern Recognition*, pp.283-286, 2000.

[7] Gonzalez, R. C., and Wood, R. E., "Digital Image Processing," Wesley Publishing Company, 2002.

[8] S. Liu, M. Zhihong, and S. Rashi, "Edge Based Region Growing-A New Image Segmentation Method," *Proc. of the 2004 ACM SIGGRAPH international conference on Virtual Reality Continuum and its applications in industry*, pp.302-305, 2004.

[9] Vincent, L., and Soille. P., "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol.13, no.6, pp.583-598, 1991.

[10] Chen, Q., Zhou, C., Luo, J., Ming, D., "Fast Segmentation of High-Resolution Satellite Images Using Watershed Transform Combined with and Efficient Region Merging Approach," *Proc. of International Workshop on Combinatorial Image Analysis*, pp.621-630, 2004.

[11] Felzenszwalb, P., and Huttenlocher, D., "Toward Objective Evaluation of Image Segmentation Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.29, no.6, pp.929-944, 2007.

[12] Canny, J., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.8, no.6, pp.679-714, 1986.

[13] Hough, P. V. C., "A method and means for recognizing complex patterns," U.S. Patent, 1962.

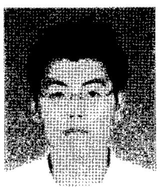
[14] Duda, R. O., and Hart, E. P., "Use of the Hough Transformation To Detect Lines and Curves In Pictures," *Communications of the ACM*, vol.15, no.1, pp.11-15, 1972.

[15] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S., "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Transactions on Circuits and Video Technology*, vol.8, no.5, pp.644-655, 1998.

[16] Jung, F., Li, M. J., Zhang, H. J., Zhang, B., "Unsupervised Image Segmentation Using Local Homogeneity Analysis," *Proc. of IEEE International Symposium on Circuits and Systems*, pp.25-28, 2003.

Nguyen, N.T.B.

2004년 Posts and Telecommunications Institute of Technology, Hochiminh city branch (BS). 2009년 Korea Advanced Institute of Science and Technology (MS). Area of Interest : Multimedia Databases



김 용 권

2005년 한국과학기술원 전자전산학과 전산학전공 학사. 2007년 한국과학기술원 전자전산학부 전산학전공 석사. 2007년 ~현재 한국과학기술원 전자전산학부 전산학전공 박사과정. 관심분야는 멀티미디어 데이터베이스, 데이터마이닝

정 진 완

정보과학회논문지 : 데이타베이스
제 36 권 제 2 호 참조

이 석 룡

정보과학회논문지 : 데이타베이스
제 36 권 제 2 호 참조

김 덕 환

정보과학회논문지 : 데이타베이스
제 36 권 제 2 호 참조