

# Quadtree를 사용한 색상-공간 특징과 객체 MBR의 질감 정보를 이용한 영상 검색

## (Image Retrieval based on Color-Spatial Features using Quadtree and Texture Information Extracted from Object MBR)

최창규<sup>†</sup> 류상률<sup>\*\*</sup> 김승호<sup>\*\*\*</sup>  
(Chang-Gyu Choi) (Sang-Ryul Ryu) (Sung-Ho Kim)

**요약** 본 논문은 이미지에서 Quadtree를 이용한 색상-공간 특징 추출과 이미지 내에 포함되어 있는 객체의 MBR(Minimum Boundary Rectangle)을 구하여 질감 정보를 추출하는 방법을 제안한다. 제안된 방법은 각 이미지로부터 DC 이미지를 만들고 색상 좌표계를 변환한 후, Quadtree를 이용하여 영역을 분할한다. 영역의 분할 기준은 제안된 조건에 의하여 이루어지며, 각 분할된 영역으로부터 대표 색상을 추출한다. 그리고, 이미지 분할(segmentation)을 통하여 각 이미지의 객체, 객체를 포함한 배경, 또는 일부 배경의 MBR을 구하고, 제안된 알고리즘에 의하여 검색된 MBR의 웨이블릿 계수(wavelet coefficients)를 계산한다. 이 계수들이 MBR의 질감 정보가 되며, 추출된 색상-공간 정보와 질감 정보를 이용하여 제안된 유사도 계산 방법을 통하여 결과를 나타내게 된다. 제안된 방법은 원 이미지(original image)에 비해 특징 정보의 저장 공간을 53% 감소시켰으며, 성능은 유사하게 나타났다. 그리고, 질감 정보를 추가함으로써, 색상-공간 특징의 단점인 객체 정보의 손실을 보완하였고, 질의 이미지의 객체를 포함한 검색 결과를 보였다.

**키워드** : Quadtree, 내용기반 영상 검색 시스템, 색상, 질감, 웨이블릿

**Abstract** In this paper, we present an image retrieval method based on color-spatial features using quadtree and texture information extracted from object MBRs in an image. The proposed method consists of creating a DC image from an original image, changing a color coordinate system, and decomposing regions using quadtree. As such, conditions are present to decompose the DC image, then the system extracts representative colors from each region. And, image segmentation is used to search for object MBRs, including object themselves, object included in the background, or certain background region, then the wavelet coefficients are calculated to provide texture information. Experiments were conducted using the proposed similarity method based on color-spatial and texture features. Our method was able to reduce the amount of feature vector storage by about 53%, but was similar to the original image as regards precision and recall. Furthermore, to make up for the deficiency in using only color-spatial features, texture information was added and the results showed images that included objects from the query images.

**Key words** : Quadtree, Content based Retrieval System, Color, Texture, Wavelet

### 1. 서론

최근 대용량의 멀티미디어 데이터 증가 및 저장 장치

의 고용량화에 따라 관리되는 데이터의 양이 급증하고 있다. 특히, 고용량의 멀티미디어 데이터를 데이터베이스화시키고, 이 데이터들을 효율적으로 검색하는 연구의 필요성이 나타나고 있다. 따라서, 데이터에 있는 색상, 질감, 모양 등의 내용을 자동적으로 추출하고 저장, 검색하는 내용 기반 검색 시스템(content-based retrieval system)이 활발히 연구되고 있다[1,2,3,4,5,6,7].

이러한 연구들 중에 먼저 색상을 자동적으로 추출하고 저장, 검색하는 몇몇 방법들이 제시되고 있다. 첫째

<sup>†</sup> 비회원 : 경북대학교 컴퓨터공학과  
cgchoi@borami.knu.ac.kr

<sup>\*\*</sup> 정회원 : 청운대학교 컴퓨터과학과 교수  
rsr@chungwoon.ac.kr

<sup>\*\*\*</sup> 종신회원 : 경북대학교 컴퓨터공학과 교수  
shkim@knu.ac.kr

논문접수 : 2002년 4월 24일

심사완료 : 2002년 9월 30일

색상의 히스토그램을 이용하여 이미지를 검색하는 방법이 있다[4]. 히스토그램 방법은 이미지 전체의 색상 히스토그램을 구하여  $n$ 차원의 벡터로 저장하는 방법이다. 이 방법은 전체 이미지의 색상 분포를 알 수 있고, 구현이 쉬운 장점이 있다. 그러나, 이 방식은 히스토그램의 특성상 색상의 위치 정보가 사라지고, 유사도 측정 시  $O(n^2)$ 의 연산을 필요로 하는 단점이 있다. 둘째, 색상의 위치 정보를 나타내기 위하여 전체 이미지를  $N \times N$  개의 영역으로 나누는 방법이 제시되었다[4,7]. 이 방법은 각 영역별로 히스토그램을 구하고 대표 색상을 추출하여 색상의 위치 정보와 저장할 데이터의 양을 줄인 것이다. 그러나, 이 방법은 영역을 분할할 때  $N$ 을 얼마로 잡느냐에 따라 검색시간과 검색 결과의 정확도에 차이가 발생한다.  $N$ 을 크게 하면 검색시간이 많이 걸리지만 검색 결과가 정확하고, 반대로  $N$ 을 작게 하면 검색 시간은 짧아지지만 검색 결과의 정확도가 떨어진다.

질감 정보는 임의의 이미지에 대한 질감 분류 또는 질감 분할, 의료 진단 등에 많이 응용된다[5,6]. 최근 많은 연구 중에 질감 정보를 분석하는 방법으로 다해상도의 공간/주파수의 정보를 제공하는 웨이블릿 변환(wavelet transform)이 많이 이용되고 있다[8,9]. 예를 들어, [8]의 경우는 트리 구조의 웨이블릿 변환(tree-structured wavelet transform)을 제안하였고, [9]에서는 웨이블릿 패킷 변환(wavelet packet transform)을 이용하여 질감 정보를 분석하였다. 특히, 질감 정보는 저주파 채널의 성분보다 고주파 채널의 성분이 중요한 경우가 있기 때문에[8], 웨이블릿 변환은 이러한 조건을 만족할 수 있다. 따라서, 본 논문에서는 웨이블릿 변환을 이용하여 질감 정보를 추출하고 이를 저장하게 된다.

본 논문에서는 먼저 색상의 히스토그램과 일정 크기의 영역으로 나누는 방법에 대한 문제점을 해결하기 위하여 Quadtree를 이용하여 이미지의 영역을 가변적으로 나누고, 검색 시 비교할 영역의 수를 줄이면서 정확한 검색 결과를 도출하는 방법을 제시한다. Quadtree는 이미지를 같은 크기 4개의 영역으로 나누는 것으로, 주로 이미지의 압축이나 분할에 많이 이용된다[10,11]. 이미지로부터 색상 정보를 추출하는 방법은, 먼저 원 이미지로부터 DC 이미지를 추출하고, DC 이미지의 화소들의 RGB값을 HSV의 색상 좌표계로 변환한다. 변환한 DC 이미지에서 전역 대표 색상(global representative colors)을 추출한다. 그리고, 변환된 이미지에서 색상의 분포에 따라 Quadtree 형태로 영역을 분할하고 대표 색상을 추출한다. 마지막으로 추출한 색상과 그 색상의 분포값을 영역의 위치에 따라 Quadtree의 단말 노드에 저

장하게 된다. 그리고, 사용자가 질의를 주었을 때 먼저 질의 이미지의 전체 대표 색상과 데이터베이스내의 각 이미지의 전체 대표 색상을 비교하는 필터링(filtering) 과정을 거친다. 필터링 과정을 통과한 데이터베이스내의 이미지와 질의 이미지를 Quadtree에 저장된 정보를 이용하여 본 논문에서 제안한 유사도 측정을 통하여 결과 이미지를 보여 준다.

그러나, 색상만의 정보를 이용하게 되면 이미지 내에 존재하는 객체들의 특징이 모두 사라지게 된다. 이런 문제점을 해결하기 위하여 본 논문에서는 질감 정보를 추출하고 이를 이용하게 된다. 입력 이미지에 대한 영역 분할을 수행하고, 구해진 영역의 MBR을 계산하게 되는데, MBR은 서로 중첩되거나 포함되는 경우가 발생한다. 이러한 경우를 분석하기 위하여 알고리즘을 제안하였으며, 이를 통하여 포함되는 MBR을 제거할 수 있다. 각 MBR의 질감 정보를 웨이블릿 변환을 이용하여 추출하고, 유사도 계산 시 이용하게 된다.

본 논문의 구성은 다음과 같다. 제 2장에서는 구현된 시스템의 개요와 색상-공간 특징 정보의 추출 방법을 서술하고, 3장에서는 객체의 MBR을 구하고 질감 정보를 추출하는 방법을 설명한다. 그리고, 4장에서는 추출된 특징을 이용하여 유사도를 측정하는 방법을 설명하고, 5장에서는 실험 결과를 보여 준다. 마지막으로 6장에서는 결론과 향후 연구 과제를 제시한다.

## 2. 시스템 개요 및 Quadtree를 이용한 색상 정보 추출

본 장에서는 제안된 시스템의 전체 구성도를 간략히 설명하고, Quadtree를 이용하여 입력 이미지의 색상 정보를 추출하는 방법을 서술한다.

### 2.1 시스템 개요

색상 정보와 질감 정보를 추출하는 전체 구성도는 그림 1과 같다. 그림 1에서 이미지의 특징 정보를 추출하는 방법은 색상-공간 특징 정보와 질감 정보로 크게 두 부분으로 나뉜다. 먼저, 색상 정보의 경우는 입력 이미지로부터 DC 이미지를 만든 후, Quadtree를 이용하여 영역을 분할하고, 영역의 분할은 특정 조건을 만족할 때까지 계속해서 이루어지게 된다. 분할이 완료되면 각 분할된 영역으로부터 대표 색상을 추출하게 된다. 또한, 색상 정보 중에 전역 색상을 추출하여 유사도 비교 시, 데이터베이스와의 검색 시간을 줄일 수 있는 필터링 과정에 이용된다. 필터링 과정에서는 질의 이미지와 데이터베이스 내의 이미지들의 전역 색상을 비교하여 임계치 이상이 되는 경우에 유사도를 측정한다.

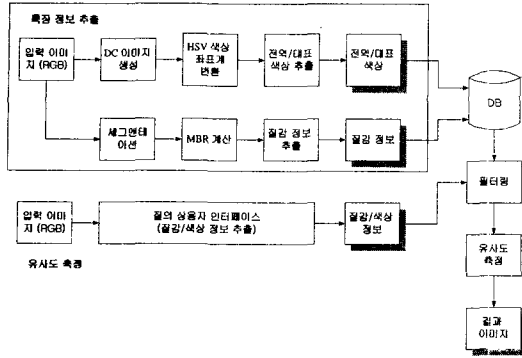


그림 1 제안된 시스템의 전체 구성도

질감 정보를 추출하기 위해서는 원래 이미지를 세그멘테이션한 후, 객체의 MBR을 구하게 된다. MBR을 구할 경우 각 MBR이 다른 MBR에 포함되었는가, MBR이 배경과 같지 않은가를 판단하여 포함되거나 배경과 유사한 MBR을 제거하게 된다. MBR의 제거가 끝나면 각 MBR에 대하여 웨이블릿 변환을 적용하여 질감 정보를 추출하게 된다. 마지막으로, 색상과 질감 정보를 데이터베이스에 저장하게 된다.

그리고, 검색의 경우에는 질의 이미지에서 똑같은 방식으로 질감과 색상 정보를 추출하여 필터링 과정을 거치게 된다. 필터링 과정에서의 조건을 만족하는 이미지는 제안된 유사도의 계산 방법을 통하여 유사도가 계산되고, 질의 이미지와 유사도가 높은 이미지부터 결과 이미지로 보여지게 된다. 각 단계의 자세한 내용은 다음 장과 3장을 통하여 설명한다.

**2.2 색상 정보 추출을 위한 전처리 단계 - DC 이미지 생성, 색상 좌표계 변환**

DC 이미지는 원래의 이미지에 비해 공간적으로 축소시킬 수 있어, Quadtree를 형성할 때 트리의 깊이를 원래의 이미지보다 줄일 수 있다. 원래의 이미지에서 8x8 크기를 가지는 블록에서 블록내의 64개 화소들의 평균값을 가지는 것이 DC값이고 식 (1)과 같이 나타낸다. 식

(1)에서  $d(i, j)$ 는 DC 이미지에서의  $(i, j)$ 위치의 DC값을 나타내고,  $p_k(x, y)$ 는 원래의 이미지에서  $k$ 번째 블록에 있는 화소값이다.  $k$ 는 DC 이미지의 폭을  $w$ 이라고 할 때 식 (1)에서와 같이 나타내는데, 예를 들어  $d(0, 0)$ 는 원 이미지의 1번째 블록내의 화소들의 평균값이 된다.

$$d(i, j) = \frac{1}{64} \sum_{x=1}^8 \sum_{y=1}^8 p_k(x, y) \quad (1)$$

$$k = (w \times i + j) + 1$$

이미지들의 종류에 따라서 식 (1)을 적용할 수도 있고, 적용되지 않는 경우도 있다. BMP와 같은 압축이 되지 않는 이미지는 식 (1)을 적용하여 DC 이미지를 만들게 되고, JPEG[12]과 같은 압축 이미지는 파일에서부터 바로 DC 값을 읽어서 생성하게 된다. 이와 같이 DC 이미지에서 색상 정보를 추출할 경우 이미지의 크기가 원래의 이미지보다 1/64로 줄어들어 Quadtree의 깊이를 줄일 수 있고, 저장할 데이터의 양을 줄일 수 있다.

색상 정보에 사용되는 색상 좌표계는 사람들이 인지하는 색상으로 표현하기 쉬운 HSV 색상 좌표계를 사용한다. H(Hue)값은 색상들의 집합을 나타내고, S(Saturation)는 색상의 세기, V(Value)는 색상의 밝고 어두운 것을 나타낸다. 예를 들어 S가 크면 순도가 높아지고 V가 크면 흰색에 가깝게 된다. 각 성분의 구간 값은 H의 경우 [0,360]이고, S와 V는 [0,1] 사이의 값으로 표현된다.

본 논문에서는 대표 색상(전역 색상과 영역별 대표 색상) 추출을 위하여 [13]에서 제안되었던 비선형(nonlinear) 방식의 대표 구간 나누기를 수행한다. 이 방법은 H, S, V값을 보고 H는 7개의 구간으로 나누고 S와 V는 각각 두 구간으로 나누게 된다. 각 구간의 범위는 표 1과 같다. HSV의 구간 값이 정의되고 나면 식 (2)와 같이 계산되어 레이블이 적용된다. 따라서, 최종적으로 36개의 레이블이 만들어지게 된다. 표 1에서  $v$ 의 구간이 [0,0.2]인 경우는 검은색에 속하는 구간이고,  $s \in [0,0.2]$ 이고  $v \in [0, 8, 1, 0]$ 인 경우는 흰색 구간에 속하게 된다. 따라서, 무채색인 경우에는 특정 값을 주게되고  $s \in [0,0,2]$ 이고  $v \in [0,$

표 1 HSV 색상 좌표에서의 비선형 방식의 대표 구간 나누기

H	구간	(330,22)	(22,45)	(45,70)	(70,155)	(155,186)	(186,278)	(278,330)
	레이블	0	1	2	3	4	5	6
S	구간	(0.2,0.65)	(0.65,1)	$v \in [0,0.2]$ 이면 $l=0$ $s \in [0,0.2]$ 이고 $v \in (0.2,0.8)$ 이면 $l = \lfloor (v-0.2) \times 10 \rfloor + 1$ $s \in (0,0.2)$ 이고 $v \in (0.8,1.0)$ 이면 $l=7$				
	레이블	0	1					
V	구간	(0.2,0.7)	(0.7,1)					
	레이블	0	1					

2,0,8인 구간은 그레이색상에 속한다.

$$l = 4H + 2S + V + 8 \quad (2)$$

**2.3 Quadtree를 이용한 색상 정보 추출**

색상 정보를 추출하기 위한 첫 번째 단계로서 DC 이미지를 4등분하고, 각 1, 2, 3, 4분면의 화소들의 레이블 분포를 보고, 영역을 분할하거나 대표 색상 레이블을 추출하게 된다. 한 영역에서의 색상 추출은 영역 내의 히스토그램을 구하여 레이블의 개수가 많은 4개의 색상 레이블을 선택한다. 식 (3)은 한 영역에서 추출한 4개의 대표 색상 레이블이 그 영역의 대표 색상 레이블이 되기 위한 조건을 나타낸다.

$$\sum_{i=1}^4 C_i \geq 0.9 \quad (3)$$

$$C_i = \frac{R_{i\_pixel}}{R_{total\_pixel}}$$

만약, 식 (3)의 조건을 만족하면 그 영역의 대표 색상 레이블로 추출하고, 조건을 만족하지 않으면 그 영역을 다시 4등분으로 분할한다. 식 (3)에서  $R_{i\_pixel}$ 은  $i$ 번째의 대표 색상 레이블의 화소수를 나타내고,  $R_{total\_pixel}$ 은 영역의 전체 화소수이다. 그리고  $C_i$ 는 영역의 전체 화소수에서 해당되는 색상의 분포값을 나타낸다. 추출된 4개의 대표 색상 레이블이 영역의 90%를 차지하지 않으면 그 영역은 분할하고, 영역의 크기가 전체 이미지의 0.2% 이하가 되면 분할을 멈추게 된다. 영역의 크기를 제한함으로써 화소 단위가 되는 것을 막을 수 있다.

하나의 이미지는 Quadtree 구조로 보면 레벨 0이 되고, 처음 분할된 4개의 영역은 레벨 1이 된다. 레벨 1에 해당하는 각 영역은 식 (3)의 조건에 따라 영역을 분할하거나 대표 색상 레이블을 추출하게 된다. 그림 2는 분할된 이미지와 결과로 나온 Quadtree를 보여 준다. Quadtree의 단말 노드에는 대표 색상 레이블과 레이블의 분포값을 가지며, 하나의 이미지로부터 만들어지는 단말 노드의 개수는  $d$ 를 트리의 깊이라 할 때  $4^l \sim 4^d$ 개가 된다. 예를 들어 그림 2의 경우는 단말 노드의 개수

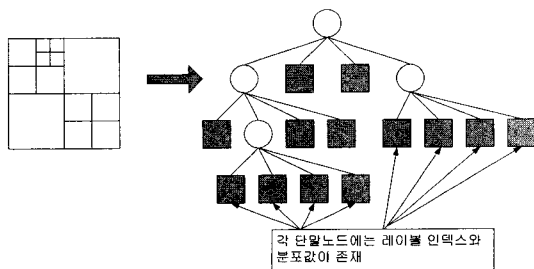
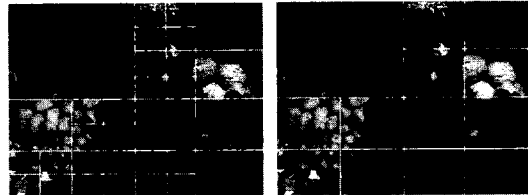


그림 2 이미지로부터 만들어진 Quadtree 구조

가 13개 된다. 즉, Quadtree 구조를 이용하여 영역을 분할한 결과, 13개의 영역으로 나누어진 경우가 된다. 그림 3은 이미지를 Quadtree를 이용하여 분할된 결과로서 (a)는 원래 이미지의 결과이고, (b)는 DC 이미지의 결과를 보여준다.



(a) 실제 이미지 (b) DC 이미지

그림 3 Quadtree를 이용한 영역 분할

**2.4 필터링 과정**

필터링 과정은 질의 이미지와 데이터베이스 내의 이미지간의 유사도를 측정하기 전의 전처리 과정으로 검색 시간을 줄일 수 있는 방법이다. 예를 들어, 파란 하늘과 녹색의 잔디가 있는 이미지를 찾을 경우 빨간색의 분포가 많은 이미지와 비교할 경우 유사도는 거의 0에 가깝게 된다. 이런 경우, 빨간색이 많은 이미지는 유사도를 비교할 필요가 없다. 이와 같이, 질의 이미지의 색상 분포와 유사한 색상 분포를 가진 이미지에 대하여 유사도를 측정할 경우 보다 높은 유사도가 나오게 된다. 비교할 이미지가 많을 경우 필터링 과정을 거치게 되면, 필터링 과정을 거치지 않은 방법보다 검색 시간이 줄어들면서 결과는 유사하게 나온다.

필터링 과정은 먼저 DC 이미지에서 전역 히스토그램(global histogram)을 구하고 화소수가 많은 색상 레이블을  $L$ 개 추출한다. 전역 히스토그램은 색상 정보를 추출하기 이전에 만들어진 DC 이미지의 색상 히스토그램을 구한 것을 말하며, Quadtree에서 추출했던 같은 색상 좌표계와 레이블을 적용한다. 모든 입력 이미지는  $L$ 개의 전역 히스토그램의 정보를 가진 색상 레이블과 Quadtree를 이용하여 추출된 대표 색상 레이블을 가지게 된다. 유사한 이미지를 찾기 위한 질의 이미지가 주어질 경우, 질의 이미지의 전역 색상 레이블과 데이터베이스 내의 전역 색상 레이블이 같은 경우에 대하여 화소수를 비교하여 그 값을  $F_i$ 에 저장한다. 질의 이미지의  $L$ 개 레이블을 모두 비교한 후 같은 레이블이  $e$ 개 존재할 때, 최종적인 필터링 값  $F_{sum}$ 은 식 (4)와 같이 나타난다.

$$F_{sum} = \frac{1}{L} \sum_{i=1}^e F_i \quad (4)$$

$$F_i = \frac{DB\_num_i}{Q\_num_j} \quad \text{if } D\_label_i = Q\_label_j$$

( $i, j$ : 전역 색상 레이블 인덱스)

식 (4)에서  $D\_label_i$ 는 데이터베이스 내의 한 이미지의  $i$ 번째의 전역 색상 레이블이고  $D\_label_j$ 는 질의 이미지의  $j$ 번째의 전역 색상 레이블을 나타낸다. 그리고  $D\_num_i$ 는 데이터베이스 내의 한 이미지의  $i$ 번째 전역 색상 레이블의 화소 수이고  $D\_num_j$ 는 질의 이미지의  $j$ 번째 전역 색상 레이블의 화소 수이다. 질의 이미지와 데이터베이스내의 이미지에서 같은 레이블에 대하여 합을 구한 값이  $F_{sum}$ 이 된다.  $F_{sum}$ 이 주어진 임계치( $\tau$ ) 이상이면  $Filter$ 는 참이 된다. 즉, 아래의 조건과 같이  $Filter$ 값이 참이 되면 유사도를 측정하고, 거짓이 되면 데이터베이스 내의 다음 이미지와 비교하게 된다.

$$Filter = TRUE \quad \text{if } F_{sum} \geq \tau$$

### 3. 객체의 MBR 검출과 질감 정보 추출

질감 정보를 추출하기 위하여 웨이블릿 변환이 많이 사용되고 있음을 앞에서 언급하였으며, 본 논문에서는 질감 정보 추출을 위하여 이 변환을 이용한다. 질감 정보 추출을 위한 전단계로 객체의 MBR을 검출하는 방법을 제안하고, 검출된 MBR로부터 웨이블릿 변환을 적용하게 된다. 먼저, 웨이블릿 변환에 대한 간단한 소개와 제안된 방법을 설명한다.

#### 3.1 웨이블릿 변환

웨이블릿 변환은 신호와 웨이블릿(wavelet)의 내적을 이용하여 계수를 계산한다. 각 웨이블릿은 모웨이블릿(mother wavelet)인  $\Psi(x)$ 에서 스케일(scale)과 쉬프트(shift) 함수를 통하여 만들어지게 된다. 웨이블릿 변환의 수식은 식 (5)와 같이 나타낸다. 모웨이블릿을 만들기 위해서는 스케일 함수  $\Phi(x)$ 와 웨이블릿 커널  $\Psi(x)$ 를 정의해야 하며 (식 (6)), 특히 계수인  $h(x)$ 와  $g(k)$ 는 이산 웨이블릿 변환(discrete wavelet transform)에서 중요한 역할을 한다[14].

$$\Psi_{m,n}(x) = 2^{-m/2} \Psi(2^{-m}x - n) \quad (5)$$

$$\Psi(x) = \sqrt{2} \sum_k g(k) \Phi(2x - k) \quad (6)$$

$$\Phi(x) = \sqrt{2} \sum_k h(k) \Phi(2x - k), \quad g(k) = (-1)^k h(1 - k)$$

웨이블릿 분석에서 신호는 저주파의 성분을 가지는 근사값(approximation)과 고주파의 성분인 상세값(detail)으로 나누어진다. 일반적인 웨이블릿 변환의 경우, 근사값은 다시  $L$  레벨까지 근사값과 상세값으로 세분화할 수 있다. 반면, 웨이블릿 패킷 변환(wavelet packet

transform)인 경우에는 근사값과 함께 상세값까지 세분화 할 수 있기 때문에  $2^L$ 개까지의 주파수 성분을 구할 수 있다. 2차원 웨이블릿 변환에 이용되는 기본 함수(basis function)는 수평, 수직, 대각선의 고주파 성분과 하나의 저주파 성분을 구할 수 있는 필터를 제공한다. 따라서, 2-D 필터 계수는 다음과 같이 표현할 수 있다.

$$h_{LL}(k, l) = h(k)h(l), \quad h_{LH}(k, l) = h(k)g(l)$$

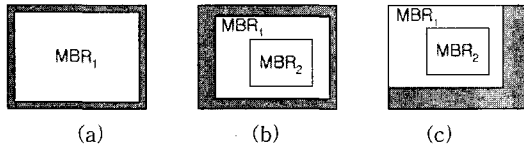
$$h_{HL}(k, l) = h(k)h(l), \quad h_{HH}(k, l) = h(k)h(l)$$

#### 3.2 이미지 내의 객체 MBR 검출

이미지 내에는 여러 개의 객체가 존재할 수 있으며, 이러한 객체에 대한 정보 또한 중요하다. 이미지 내의 객체에 대한 MBR은 [3]에서 제안한 방법을 변형하여 객체, 객체를 포함한 배경 등을 구한다. [3]에서 제안한 방법을 간단히 보면, 입력 이미지의 RGB 좌표계를  $L^*a^*b^*$ 의 색상 좌표계로 변형한 후 이미지 분할을 수행한다. 이미지 분할은 먼저 이미지 상에 있는 임의의 화소와 주위 8방향의 화소간의 색상간의 거리가 임계값 이하가 되면 같은 영역에 포함을 시킨다. 영역에 포함된 화소의 색상값과 영역의 색상값에 대하여 평균 색상 값을 구하고, 영역의 크기가 미리 정의된 크기를 넘지 않는 범위에서 계속 성장(growing)을 하게 된다. 영역의 성장이 멈춰지게 되면 그 영역의 대표 색상과 영역의 MBR 등의 특징 정보를 추출한다. [3]에서 제안한 방법은 영역의 크기가 미리 정의되어 있기 때문에 이미지 내에 존재하는 객체들이 여러 개로 분리되는 문제가 있다. 따라서, 본 논문에서는 이러한 영역의 크기를 제한하지 않고 영역 분할을 수행한다.

영역의 분할을 통하여 나온 결과는 객체의 MBR을 추출할 수 있지만, 몇 가지 문제점을 가지게 된다. 첫 번째는 검출된 MBR을 추출할 수 있지만, 몇 가지 문제점을 가지게 된다. 첫 번째는 검출된 MBR이 배경과 유사하게 되는 경우 객체의 특징은 사라지게 됨으로 검출된 MBR이 배경인지를 판단하여 이를 제거해야 한다. 두 번째는 하나의 MBR이 다른 MBR을 포함하는 경우로서, MBR의 특징 정보가 중복되어 나타나는 문제가 발생한다. 본 논문에서는 이 문제를 해결하기 위해서 내부에 포함되는 객체의 MBR을 제거하게 된다. 마지막으로 MBR의 한 좌표의 값이 0인 경우로, 이 경우에 MBR 내에 또 다른 MBR을 포함하고 있으면 제거를 해야 한다. 좌표의 값이 0인 것은 MBR이 이미지 내의 배경부분인 경우가 많기 때문에 특정 객체를 포함하고 있으면 객체의 특성이 많이 없어지게 된다. 따라서, 내부의 MBR을 포함하는 MBR을 제거함으로써 해서 내부

의 MBR 특징을 더욱 높여 줄 수 있다. 그림 4는 제거되어야 할 MBR의 유형을 보여 주고 있다. 그림 5는 MBR 제거를 위한 의사 코드(pseudo code)이다.



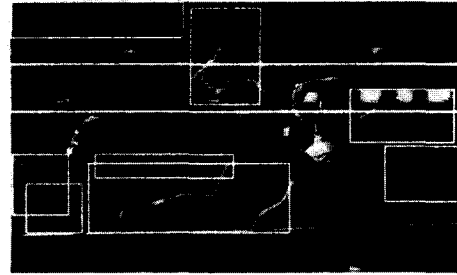
(a) MBR<sub>1</sub>의 크기가 이미지의 크기와 유사함으로 제거  
 (b) MBR<sub>1</sub>의 좌표값이 0이고 내부에 MBR<sub>2</sub>를 포함하므로 제거  
 (c) MBR<sub>2</sub>는 MBR<sub>1</sub>에 포함됨으로 제거

그림 4 MBR 제거 유형

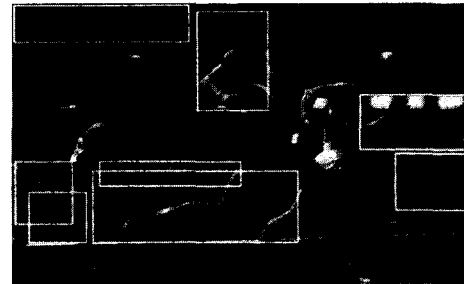
그림 6의 (a)는 영역 분할을 통하여 검출된 MBR의 위치를 나타내고, (b)는 그림 5의 조건을 판단하여 MBR이 제거된 후의 결과를 보여준다. 그림 (a)에서는 전체 배경이 되는 부분이 검출되었지만 그림 5의 조건에 의하여 제거된 것을 알 수 있으며 일부 검출되지 않은 영역은 영역의 크기가 작기 때문에 제거된 것이다. 그림 6에서 (a)는 10개의 MBR을 구하였고, MBR제거를 통하여 최종적으로 8개를 검출하였다. 검출된 각 MBR로부터 웨이블릿 변환을 적용하여 질감 정보를 추출하게 된다.

**3.3 웨이블릿 변환을 이용한 질감 정보 추출**

이미지 내에서 검출된 MBR은 웨이블릿 변환에 적용하기 위하여 그레이 이미지로 바꾼다. 본 논문에서는



(a) MBR를 제거하지 않은 상태



(b) MBR 제거 조건에 의해 제거된 상태

그림 6 MBR 제거 예

2-D 이산 웨이블릿 변환을 이용하며 레벨은 3까지를 적용한다. 입력 이미지는 웨이블릿 변환을 통하여 하나의 근사값과 3개의 상세값이 나타난다. 이 중 근사값은 다시 웨이블릿 변환을 적용하며 레벨이 3이 될 때까지

```

    이미지의 폭 : width, 높이 : height
    MBRi의 좌표 : (lefti, topi, righti, bottomi), 폭 : widthi, 높이 : heighti
    MBRj의 좌표 : (leftj, topj, rightj, bottomj), 폭 : widthj, 높이 : heightj

    경우 1 : MBRi ( 또는 MBRj )의 크기가 이미지의 크기와 유사한 경우 (그림 4 (a)).
    if ((widthi + widthi×0.05 ≥ width) and (heighti + heighti×0.05 ≥ height))
        then remove MBRi
    end if

    경우 2 : MBRi ( 또는 MBRj )의 좌표값이 0이고, 내부에 MBRi( 또는 MBRj )를 포함하고 있는 경우 (그림 4 (b)).
    if ((lefti == 0) or (topi == 0))
        if ((leftj ≥ lefti) and (topj ≥ topi) and
            (rightj ≤ righti) and (bottomj ≤ bottomi))
            then remove MBRi
        end if
    end if

    경우 3 : MBRi( 또는 MBRj )가 MBRj( 또는 MBRi )를 포함하는 경우 (그림 4 (c)).
    if ((leftj ≥ lefti) and (topj ≥ topi) and (rightj ≤ righti) and (bottomj ≤ bottomi))
        then remove MBRj
    end if
    
```

그림 5 MBR 제거를 위한 의사 코드

반복한다. 그림 7은 분할된 이미지의 계수를 구하기 위하여 적용되는 웨이블릿 필터이다. 그림 7에서  $f_1$ 부터  $f_{10}$  까지가 하나의 MBR의 질감 정보가 되며  $F_w = (f_1, f_2, \dots, f_{10})$ 으로 표현할 수 있다.

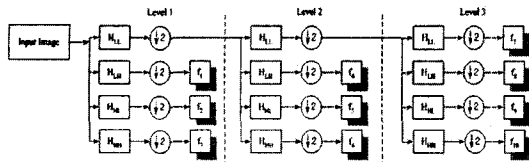
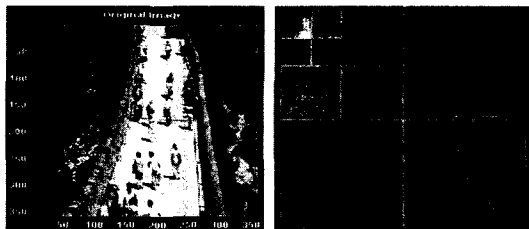


그림 7 질감 정보 추출을 위한 웨이블릿 필터

각각의 질감 정보는 식 (7)과 같이 에너지 값을 얻을 수 있다. 추출된 계수 중  $f_n$ 은 저주파 성분을 가지고 있고 나머지 계수들은 고주파 성분을 가지게 된다. 그림 8은 하나의 MBR에 대하여 웨이블릿 변환을 수행한 결과를 보여주는데, 왼쪽의 가장 윗부분의 채널은 원래의 이미지보다 크기가 작아진 형태로 저주파 부분이 되며 나머지는 에지의 정보를 가지게 된다. 식 (7)에서  $i$  번째의 질감 정보를  $f_i$ 이고, MBR의 폭을  $M$ , 높이를  $N$ 으로 나타내며,  $p_{ki}$ 은 웨이블릿 변환에 의한  $(k, l)$  좌표에서의 웨이블릿 계수값이 된다.

$$E_{f_i} = \frac{1}{MN} \sum_{k=1}^M \sum_{l=1}^N |p_{kl}| \quad (7)$$



(a) 입력 이미지 (b) 웨이블릿 변환 결과

그림 8 웨이블릿 필터를 통한 결과 이미지

#### 4. 유사도 측정

유사도 측정은 색상-공간 정보와 질감 특징 정보가 따로 계산된 후 각각에 가중치 값을 주고 이에 대한 합으로 질의 이미지와 데이터베이스 이미지간의 유사도 값을 측정하게 된다. 질의에 대한 유사도 측정을 위하여 본 논문에서는 색상-공간 정보와 질의 정보의 유사도를 측정하는 방법을 제시한다.

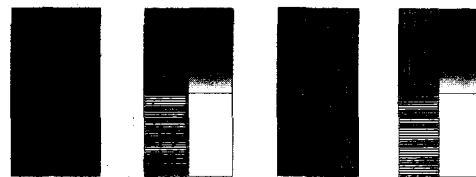
##### 4.1 색상-공간 특징 정보에 대한 유사도 측정

유사도 측정은 Quadtree의 레벨이 1인 한 노드와 그

노드에 포함되어 있는 단말 노드들의 정보(대표 색상 레이블, 대표 색상 레이블이 가지는 분포값)들을 이용한다. 그리고, 유사도 측정에서는 단말 노드의 레벨이 같은 경우와 다른 경우를 고려해야 한다. 전체 이미지의 유사도는 Quadtree의 레벨이 1인 노드들의 유사도를 각각 구한 후, 이 값을 더하게 된다.

##### 4.1.1 비교하는 단말 노드의 레벨이 다른 경우

비교하는 단말 노드의 레벨이 다른 경우는 비교하는 영역의 크기에 관계가 있기 때문에, 레벨이 다른 경우는 비교하는 영역의 크기가 서로 달라서 정규화를 해주어야 한다. 그림 9는 서로 비교하는 단말 노드의 레벨이 다른 경우를 보여 준다. 예를 들어, 그림 9 (b)를 질의 이미지라고 할 때, 4개 영역 중  $R_1$ 영역이 그림 9 (a)와 같다.  $R_1$ 영역과 그림 9 (a)만을 보면 유사도가 1이 되지만,  $R_1$  영역의 크기가 전체의 1/4밖에 되지 않으므로 두 이미지의 전체 유사도는 0.25가 됨을 알 수 있다. 따라서, 유사도 측정 시 비교하는 단말 노드의 레벨이 다른 경우를 고려해야만 정확한 결과를 얻을 수 있다.



(a) 데이터베이스 이미지 (b) 질의 이미지

그림 9 비교하는 단말 노드의 레벨이 다른 경우(질의 이미지의 크기가 작은 경우)

레벨 1의 한 노드에서 비교하는 단말 노드의 레벨이 다른 경우의 유사도는 식 (8)과 같다.

$$SO_m = \sum_{i=0}^{m-1} \frac{\left(\frac{1}{4}\right)^{|level_i - level_j|} \times P_{ij}^{dq}}{(level_i)^2}, \quad (i+j=m) \quad (8)$$

$$level_i = \begin{cases} level_i^d & (level_i^d \geq level_j^q) \\ level_j^q & (level_i^d < level_j^q) \end{cases}$$

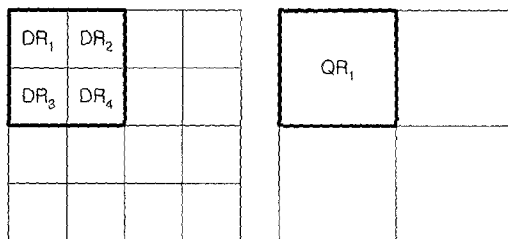
여기에서,  $i, j$ 는 레벨의 다음 위치를 표시하는 것으로 만약,  $level_j^q$ 가 크면  $i$ 를 1씩 증가시키고,  $level_i^d$ 가 크면  $j$ 를 1씩 증가시킨다. 즉, 데이터베이스와 질의 이미지간의 비교하는 단말 노드의 레벨을 배열에 두고, 레벨이 큰 쪽의 배열 인덱스를 증가시키는 의미가 된다.

$$P_{ij}^{dq} = \frac{\sum_{l=0}^{(4-null\_label)-1} p_{h_l}^d}{(4-null\_label)} \quad (9)$$

식 (8)의  $SO_m$ 은 비교하는 단말 노드의 유사도 값을

나타낸다.  $m$ 은 질의 이미지에서 레벨이 1인 한 노드 내에서 비교하는 단말 노드의 레벨이 다른 노드의 개수이다.  $level_i^q$ 는 비교하는 데이터베이스 이미지의  $i$ 번째 단말 노드의 레벨이고  $level_j^q$ 는 질의 이미지의  $j$ 번째 단말 노드의 레벨을 의미한다.  $level_a$ 는 비교하는 질의 이미지와 데이터베이스 내의 단말 노드의 레벨 중 상위 레벨을 의미하며 정규화를 위한 변수이다. 식 (9)에서  $P_h^{dq}$ 는 비교하는 레벨이  $h$ 번째인 두 단말 노드에서 색상 레이블이 같은 값들이 어느 정도 분포하는 지를 나타낸다.  $p_h^q$  또는  $p_h^s$ 는  $h$  레벨에 해당되는 단말 노드의  $j$ 번째 대표 색상 레이블이 가지는 색상 분포값이다.  $null\_level$ 이 발생하는 경우는 질의 이미지의 한 영역에서 추출된 대표 색상 레이블 중 색상 정보가 전혀 없을 때이다. 즉, 한 영역의 대표 색상 레이블이 3개 이하로도 그 영역을 대표(식 (3)의 조건 만족)할 수 있는 경우를 고려한 것이다.

식 (8)에서  $(1/4)^{|level_i^q - level_j^q|}$ 는 서로 비교하는 단말 노드의 크기를 고려하여 정규화 시키기 위한 변수이고,  $1/(level_b)^2$ 은 비교하는 질의 노드와 이 노드를 포함하는 레벨이 1인 한 노드에 대한 크기를 고려한 것이다. 그림 10은 두 변수를 이용한 유사도를 얻는 예를 보여준다. 그림 10의 두 이미지는 전체 이미지의 1/4로써 레벨이 1인 경우이다. 질의 이미지에서  $QR_1$ 은 레벨이 2가 되고 데이터베이스 내의 이미지에서  $DR_1 \sim DR_4$ 은 레벨이 3이 된다. 만약,  $QR_1$ 과  $DR_1 \sim DR_4$ 가 같은 색상 분포값을 가진다고 가정하면 질의 이미지에서  $QR_1$ 과  $DR_1$ 의 유사도는 1이 된다. 그러나, 질의 이미지의 영역이 더 크기 때문에 유사도는 1/4로 떨어지지만  $QR_1$ 과 데이터베이스내의 4개의 영역과 비교하면 유사도는 1이 된다.  $QR_1$ 은 질의 이미지의 1/4에 해당하므로 ( $level_b = 2$ ) 정규화 하면 최종적인  $QR_1$ 의 유사도는 1/4 이 된다.



(a) 데이터베이스 이미지 (b) 질의 이미지  
그림 10 비교하는 단말 노드의 레벨이 다른 경우(질의 이미지의 크기가 큰 경우)

4.1.2 비교하는 단말 노드의 레벨이 같은 경우와 색상-공간 유사도

비교하는 단말 노드의 레벨이 같은 경우는 질의 노드와 이 노드를 포함하는 레벨이 1인 한 노드에 대한 크기만을 고려하면 된다. 질의 이미지에서 레벨이 1인 한 노드 내에서 비교하는 단말 노드의 레벨이 같은 노드의 수가  $n$ 일 때, 유사도는 식 (10)과 같다. 식 (10)의  $SE_n$ 은 비교하는 단말 노드의 레벨이 같은 경우의 유사도값을 나타내고,  $P_{dq}$ 는 식 (9)와 같은, 비교하는 단말에서 같은 색상 레이블의 분포 값이다. 그리고,  $(1/4)^{|level_i^q - 1|}$ 은 비교하는 질의 노드와 이 노드를 포함하는 레벨이 1인 한 노드에 대한 크기를 고려한 정규화 변수이다.

$$SE_n = \sum_{k=0}^{n-1} \left( P_k^{dq} \times \left(\frac{1}{4}\right)^{|level_i^q - 1|} \right) \quad (10)$$

질의 이미지에서 레벨이 1인 한 노드에 포함하는 단말 노드의 유사도 측정을 위하여 데이터베이스 내의 이미지와 비교하는 단말 노드의 개수는  $m+n$ 개이다. 즉, 유사도 측정을 위하여 비교하는 단말 노드들이 레벨이 서로 다를 경우와 같은 경우를 나누어 계산한 후, 이 두 경우를 더하여 유사도를 측정한다. 식 (11)의  $Sim\_node$ 는 질의 이미지의 레벨이 1인 한 노드의 유사도 값이다. 질의 이미지의 레벨이 1인 모든 노드의 유사도를 구한 후, 식 (12)에서처럼 각 노드의 유사도를 합하면 질의 이미지에 대한 색상-공간 유사도( $Color_{sim}$ )를 구할 수 있다.

$$Sim\_node = SO_m + SE_n \quad (11)$$

$$Color_{sim} = \frac{\sum_{i=0}^N Sim\_node_i}{N} \quad (1 \leq N \leq 4) \quad (12)$$

4.2 질감 정보에 대한 유사도 측정

질의 이미지와 데이터베이스 이미지간의 질감 정보를 비교하기 위해 MBR의 크기, 거리를 고려해야 한다. 질의 이미지의 MBR은 데이터베이스의 MBR간의 크기를 비교한 값과 거리의 차이를 구하고 질감 정보의 유사도를 더하고, 구하여진 값들 중에 가장 최소인 값이 유사도가 된다. 그러므로, 하나의 질의 이미지의 MBR에서 최소 유사도 값을 구하기 위해서는 모든 요소들이 최소가 되어야 한다. 질의 이미지의 모든 MBR을 이와 같이 구하여 합을 하게 되면 최종적인 유사도가 나온다.

먼저, MBR의 거리를 비교하기 위하여 MBR의 중앙점(center point)을 구하게 되며, 이 값은 정규화된 값인 0과 1사이의 값으로 값을 비교하는 이미지의 크기에 관계 없도록 해야 한다. 이미지의 크기를 이고, MBR의 좌표값 ( $left, top, right, bottom$ )을 ( $MBR_L, MBR_T, MBR_R, MBR_B$ )로 할 때, 정규화된 중앙점은 식 (13)과



같다. 식 (13)에서 구한 각 MBR의 좌표값들은 유사도 계산 시, 유클리디안 거리(euclidean distance)를 이용하게 된다. 그러나 좌표값들이 정규화 된 값이므로 정확한 값을 유도해 내기 위해 각각의 좌표값에 가중치를 곱하여 1 이상의 값으로 바꾼 후, 거리 계산을 한다(식 (14) 참조).

$$(Center_x, Center_y) = (MBR_L + \frac{(MBR_R - MBR_L)}{2M}, MBR_T + \frac{(MBR_B - MBR_T)}{2N}) \quad (13)$$

크기에 대한 비교는 질의 이미지 내에 있는 MBR을 기준으로 하여 데이터베이스의 MBR의 상대적인 크기를 구한다. 따라서, 질의 이미지의 MBR의 크기를  $m \times n$ 이라 하고, 데이터베이스에 있는 MBR을  $m' \times n'$ 라 할 때, 크기는  $(m'/n, n'/n)$ 이 된다. 그러나, 두 MBR 간의 크기 값은 최소값이 아니므로 식 (14)의  $Sim_s$ 와 같이 1에서 빼주어야 한다. 질감 정보에 대한 유사도는 L1-norm을 이용하며, 계산 후의 에너지값을 정규화된 값으로 하기 위하여 비교하는 채널 중, 최대값을 분모로 하여 나누어주게 된다. 식 (14)의  $Sim_E$ 는 비교하는 두 MBR간의 질감 정보의 유사도를 비교하는 식이다. 식 (14)의 유사도 값을 이용하여 최종 유사도를 비교하는 방법은 알고리즘 1과 같다.

$$Sim_D = \sqrt{u(MBR_{DB_C} - MBR_{Q_C})^2 + u(MBR_{DB_O} - MBR_{Q_O})^2} \quad (u \geq 10)$$

$$Sim_S = \begin{cases} 1 - \frac{(m'/m + n'/n)}{2} & \text{if } m' \leq m \text{ or } n' \leq n \\ 1 & \text{else} \end{cases}$$

$$Sim_E = \sum_{i=1}^L \frac{|MBR_{DB_E} - MBR_{Q_E}|}{\max(MBR_{DB_E}, MBR_{Q_E})} \quad (14)$$

**Algorithm 1** : 질감 정보의 유사도 측정  
 입력 : 질의 이미지의 MBR 개수 X, 데이터베이스 이미지의 MBR 개수 Y  
 출력 : 유사도  $Sim_c$  ( $0 \leq Sim_c \leq 1$ )  
**Begin**  
 for ( $i=1; i \leq X; i++$ )  
   for ( $i=1; j \leq Y; j++$ )  
    $Sim_{dist} = Sim_D(i, j);$  /\* MBR<sub>i</sub>와 MBR<sub>j</sub> 간의 거리 계산 \*/  
    $Sim_{size} = Sim_S(i, j);$  /\* MBR<sub>i</sub>와 MBR<sub>j</sub> 간의 크기 계산 \*/  
    $Sim_{energy} = Sim_E(i, j);$  /\* MBR<sub>i</sub>와 MBR<sub>j</sub> 간의 질감 정보 계산 \*/  
   /\*  $w_1, w_2$ 는 가중치 값 \*/  
    $Sim(j) = (w_1 \times (Sim_{dist} + Sim_{size}) + w_2 \times Sim_{energy}) / (w_1 + w_2);$   
   endfor  
    $Sim(i) = \min\{1 \leq j \leq Y \mid Sim(j), Sim(j) \neq 0\}$   
 endfor  
 $Texture_{sim} = SUM(Sim(i));$   
**End**

최종적인 유사도의 가중치 ( $w_1, w_2$ )를 적용하여 식 (15)와 같이 나타내며, 사용자의 관점에 따라 가중치를 조정하여 색상-공간 특징 정보를 높일 수도 있고, 질감 정보를 높일 수 있다.

$$Similarity = \frac{w_1 \times Color_{sim} + w_2 \times Texture_{sim}}{(w_1 + w_2 = 1)} \quad (15)$$

## 5. 실험 결과

### 5.1 실험 환경

본 장에서는 제안된 방법과 다른 방법들을 비교 분석하고 그 결과를 보여준다. 실험에 사용된 이미지는 640×480, 512×769, 1024×466 등의 크기를 가진 자연, 우주, 비행기, 꽃, 동물 등 다양한 종류의 칼라 이미지 1000장을 사용하였다. 질감 정보를 추출하기 위하여 다우비치 웨이블릿(Daubechies wavelet) 필터를 이용하였고, 3 레벨에 대하여 16개의 탭(tab)을 사용하였다. 실험에 사용된 시스템은 윈도우즈 2000의 운영체제와 Pentium 4의 1.3 GHz의 CPU이다.

본 논문에서 제안한 방법과의 비교를 위하여 먼저, 원래의 이미지와 DC 이미지간의 결과의 차이를 실험하였다. 즉, 입력으로 들어오는 이미지를 원래의 이미지와 DC 이미지로 하여 그 결과 차이를 알아보았다. 그리고, 다른 방법과의 비교를 위하여 히스토그램 방법과 일정 영역으로 나눈 방법을 사용하였다. 두 방법은 본 논문에서 제안한 방법과 같은 환경을 유지하기 위하여, DC 이미지를 만들고 HSV 좌표계로 바꾸었다. 좌표계가 변환된 화소들의 값을 보고 각 화소들은 레이블로 대체하였다. 히스토그램 방법은 DC 이미지에서 해당되는 레이블의 화소수를 구하고, 일정 영역으로 나누는 방법은 DC 이미지를 3×3(9개의 영역)과 8×8(64개의 영역)로 나누어 각 영역별로 4개의 대표 색상 레이블을 추출하였다. 질의에 대한 검색 결과는 사용자가 임의의 임계치( $\lambda$ )를 주고 이 값 이상이 되는 이미지를 보여 주었다. 두 번째는 질감 정보를 추가했을 때의 결과를 알아보았다.

검색에 대한 정확도를 보기 위하여 Precision과 Recall을 실험평가 기준으로 사용하였다. 임의의 질의에 대하여 데이터베이스 내에 질의 이미지와 관련된 총 이미지 수를 I라 하고, 임계치( $\lambda$ )이상으로 검색된 이미지 수를  $R_i$ 라 하며, 사람이 보기에 유사하다고 판단되는 이미지 수를  $R_s$ 이라고 할 때 Precision과 Recall은 다음과 같다.

$$Precision = \frac{R_s}{R_i}$$

$$Recall = \frac{R_s}{I}$$

5.2 실험 결과 및 고찰

필터링 과정을 위하여 본 논문에서는 상위 6개의 레이블을 추출하여 질의 이미지와 데이터베이스 내의 이미지의 임계치를 비교하였다. 필터링 과정에서의 문제점은 추출한 6개의 레이블이 한 이미지에서 비중이 일마나는 것이다. 실험에 사용된 1000개의 이미지에 대하여 3개부터 8개까지의 레이블을 추출한 후 각 이미지별로 전역 히스토그램의 비중을 검색해본 결과 그림 11과 같았다. 그림 11은 x축을 전역 색상이 전체 이미지에서 차지하는 비율, y축을 이미지 수로 나타낸 그림이다. 그림에서 보면 레이블 수가 6개 이상인 경우에 대하여 비슷한 이미지 수를 보여주고 있다. 즉 6개 이상의 레이블 수는 오히려 데이터베이스에 저장되는 데이터의 양이 많아질 뿐이다. 따라서, 6개의 레이블을 추출하여 질의 이미지와 데이터베이스의 이미지를 비교하는데 있어서 큰 문제점이 없음을 알 수 있다.

원래 이미지와 DC 이미지간의 비교와 다른 방법과의 비교를 하기 위해서 필터링 과정에 필요한 임계치( $\tau$ )를 0.3으로 하였고, 검색 결과의 임계치( $\lambda$ )는 이미지에 따라서 0.3에서 0.4로 하였다. 표 2는 실험 결과를 보여준다.

표 2에서 원래 이미지와 DC 이미지의 Precision과 Recall은 큰 차이가 나지 않음을 알 수 있다. 반면, 인덱스의 크기는 원래 이미지에 비해 DC 이미지가 약 54% 감소되었기 때문에, 원래 이미지가 아닌 DC 이미지를 이용하더라도 검색에 큰 영향을 주지 않으면서 저

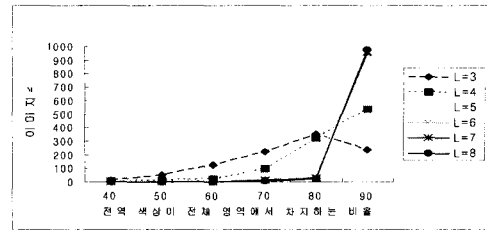


그림 11 전역 색상이 전체 영역을 차지하는 비율

표 2 제안된 방법과 다른 방법들과의 비교 실험 결과

방법	구분	$\lambda$	$\tau$	Precision	Recall	인덱스	인덱스
						크기	시간
제안한 방법	원래 이미지	0.3~0.4	0.3	0.821	0.723	0.289	8분 56초
	DC 이미지	0.3~0.4	0.3	0.805	0.709	0.124	11분 28초
일정 영역으로 나눈 방법	8×8	0.3~0.4	0.0	0.766	0.688	1	3분 54초
	3×3	0.3~0.4	0.0	0.535	0.667	0.141	3분 36초
	히스토그램	0.6~0.7	0.0	0.217	0.240	0.083	2분 44초

장 공간을 줄일 수 있다. 제안된 방법은 다른 방법에 비해 Precision과 Recall이 우수하며 인덱스의 크기는 8×8 방법에 비해 약 88% 감소시켰다. 3×3 방법과 히스토그램 방법은 인덱스의 크기는 제안된 방법보다 적지만 검색 결과가 상당히 나쁜 것을 알 수 있다. 따라서, DC 이미지를 이용한 제안된 방법은 기존의 방법에

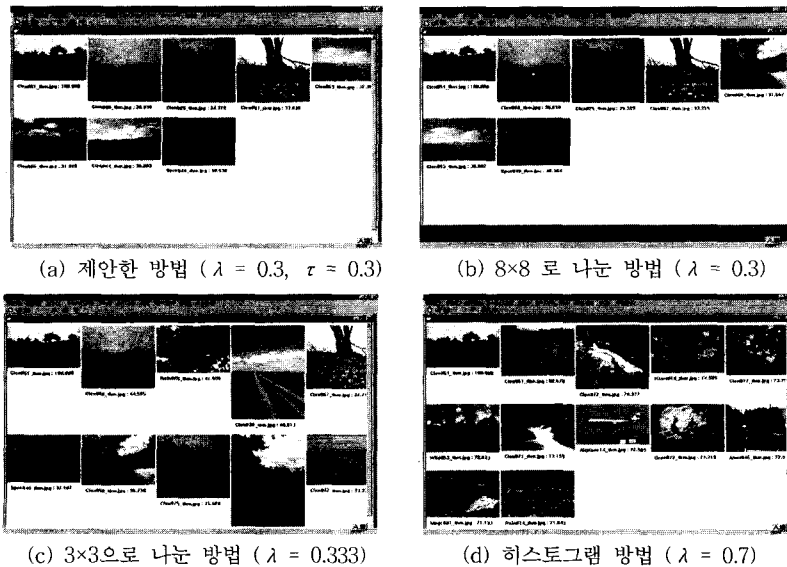


그림 12 실험 결과

비해 저장 공간을 줄이면서 좋은 결과 - 히스토그램 방법에 비해 약 74%, 3×3으로 나눈 방법에 비해 약 54% 성능 향상-를 보임을 알 수 있다. 그림 12는 실험에 의한 결과 이미지를 나타낸다.

그러나, 인덱싱 시간은 다른 방법에 비하여 3배 이상이 소요되는 단점이 있다. 원래 이미지와 비교하면 DC 이미지 생성 시간이 없기 때문이며, 다른 방법은 분할 자체가 고정적이기 때문에 인덱싱 시간이 많이 소요되지 않기 때문이다. 인덱싱 시간은 1000장에 대한 결과 시간으로 나타냈다.

색상-공간 특징 정보에 대한 성능 평가는 필터링 임계치에 따라 차이가 나타난다. 좋은 성능 평가를 얻기 위하여 적절한  $\tau$  값을 결정해야 한다.  $\tau$ 에 대한 성능 차이가 그림 13에 나타난다. 임계치가 0.3까지는 거의 성능 차이가 나타나지 않으며 0.4 이상이 되면 성능이 조금씩 떨어짐을 알 수 있다. 이러한 이유는  $\tau$  값이 커짐에 따라 비교하지 않는 이미지의 수가 있기 때문이다. 그림 13을 통하여 필터링 임계치를 0.3 정도로 하게 되면 우수한 성능 결과를 얻을 수 있다. 검색 시간은 임계치에 대하여 크게 차이가 나지 않음을 알 수 있다. 실험 결과 검색시간은 1~2초 정도가 되었다.

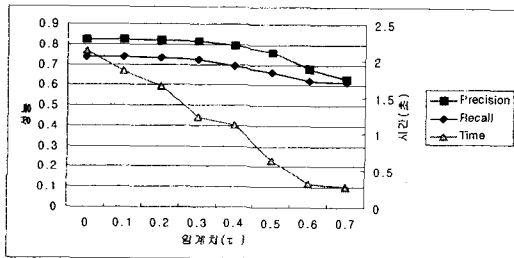
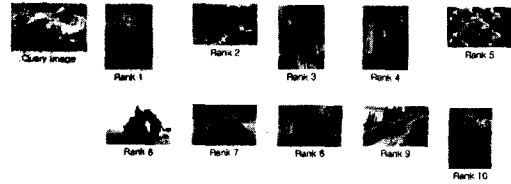


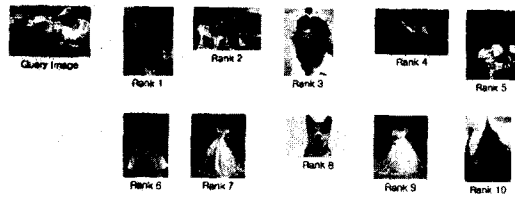
그림 13 필터링 임계치에 따른 성능 평가와 검색 시간

그러나, 색상-공간 특징만을 이용할 경우 유사한 색상에 대해서는 좋은 검출율을 보이지만(표 2 참조), 특정 객체에 대한 결과는 좋지 않다. 만약 같은 공간상에 유사한 질감이 있지만 색상이 다르게 되면 제안한 방법 또한 원하는 결과를 얻지 못하는 경우가 있다. 따라서, 이러한 문제점을 해결하기 위하여 객체의 MBR에 대한 질감 정보를 추출하였으며, 이에 대한 결과는 그림 14에서 볼 수 있다. 그림 13의 (a)는 색상-공간 정보만을 이용하여 검색된 결과이고, 그림 14의 (b)는 질감 정보에 0.6, 색상-공간 정보에 0.4의 가중치를 주었을 때의 결과를 나타낸다. 실험 결과, 객체 MBR의 질감 정보를 같이 이용함으로써 유사한 질감을 가진 객체의 추출이

이루어질 수 있다.



(a) 색상 정보만을 이용한 검색 결과



(b) 색상 정보와 질감 정보를 이용한 검색 결과

그림 14 질감 정보를 이용한 성능 향상

색상-공간 정보와 질감 정보의 가중치에 따라 유사도가 다르게 나타난다. 두 가지 특징 정보와 각각의 가중치에 대한 관계를 알아보기 위하여 그림 15와 같은 실험 결과를 보였다. 성능 평가의 기준치가 되는 Precision과 Recall은 크기에 관계하지 않는 유사한 질감 정보를 가진 MBR(배경의 MBR 포함)이 얼마나 포함되어 있는가와 지역적 색상 정보가 얼마나 유사한가를 판단하여 나온 수치이다. 실험해본 결과, 색상-공간 정보의 가중치가 0.4~0.6일 때(이 경우 질감정보는 0.6~0.4가 된다) 좋은 결과가 나올 수 있다. 즉, 어느 한 쪽에 많은 가중치를 가지게 되면 유사도가 떨어짐을 나타낸다.

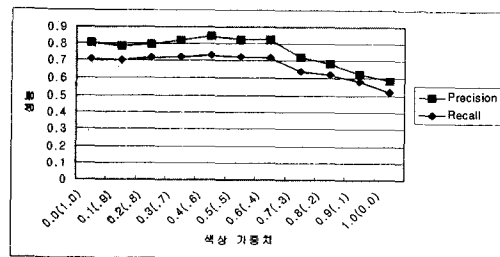


그림 15 색상-공간(질감) 가중치에 따른 성능 평가

## 6. 결론

본 논문에서는 Quadtree를 이용하여 영역을 가변적으로 나누어 비교 검색할 영역의 수를 줄이고, 필터링 과정을 통하여 전체 이미지를 검색하지 않음으로써 검색 시간을 빠르게 하는 색상 공간 특징과 객체의 MBR

의 질감 정보를 이용한 내용 기반 검색 시스템을 제안하였다. 제안한 시스템은 HSV 좌표계로 변환된 DC 이미지를 Quadtree를 기반으로 하여 영역을 분할한 후 색상 정보를 추출하였고, 추출된 색상 정보는 영역의 위치에 따라 Quadtree의 단말 노드에 저장하였다. 그리고, 필터링 과정을 통하여 데이터베이스 내의 모든 이미지의 유사도를 측정하지 않으므로써 질의에 대한 검색 시간을 줄였다. 또한 이미지 내에 객체 MBR을 구하고, 각 MBR로부터 질감 정보를 추출하여 이를 검색 시 이용함으로써 색상-공간 특징만을 이용한 경우의 단점을 보완하였다. 색상과 질감 정보를 같이 사용하여 원하는 색상과 질감을 가진 이미지를 검색할 수 있었다.

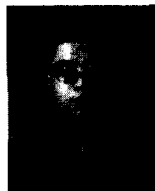
본 논문에서 제안한 방법은 질의 검색 결과, 히스토그램 방법에 비해 약 74% 정도 향상되었으며, 일정 개수의 영역으로 나누는 방법(3×3)보다 약 45%의 성능이 향상되었다. 인덱스의 저장 크기도 원래의 이미지에 비해 53%를 줄였고 8×8 방법보다 88%를 줄였으나, 검색 결과는 유사하게 나타났다. 이는 Quadtree를 이용한 영역 분할은 색상이 많이 분포된 곳은 많은 영역으로 나누고, 색상이 적게 분포된 곳은 영역의 수를 적게 함으로, 일정 크기로 나누는 방법에 비해 유사도 측정 시 보다 정확한 결과를 낼 수 있는 장점이 있다. 그리고, 필터링 과정을 통하여 전체 이미지를 검색하지 않고 질의에 대한 유사한 색상만을 가진 이미지와 유사도를 측정함으로써 검색 시간을 줄일 수 있다. 또한, 질감 정보를 추출하여 색상-공간 특징과 함께 이용할 경우 유사한 색상과 질감을 가진 이미지를 검색함으로써 보다 정확한 검색 결과를 유도할 수 있었다.

그러나, 본 논문에서 제안된 객체의 MBR은 정확하게 특정 객체의 테두리를 검출하지 못하며, 하나의 객체에 대하여 여러 개의 MBR이 발생한다. 이는 객체를 검출하는 영역 분할 방법이 보다 정확한 것이 아니기 때문이며, 이 문제의 개선을 위해서는 객체를 정확하게 검출하는 알고리즘 개발이 필요하다.

**참 고 문 헌**

[1] M. Flickner, "Query by Image and Video Content: The QBIC System," *IEEE Computer*, pp. 23-32, September, 1995.  
 [2] X. Wan and C. C. J. Kuo, "A New Approach to Image Retrieval with Hierarchical Color Clustering," *IEEE Trans. on Circuits and System for Video Technology*, Vol. 8, No. 5, pp. 628-643, September, 1998.  
 [3] 안철용, 김승호, "색상-공간 특징을 사용한 내용 기반

칼라 이미지 검색 시스템의 설계 및 구현", 정보과학회(C), 제 5권, 제 5호, pp.629-638, 1999.  
 [4] 김철원, 최기호, "칼라 지정을 이용한 내용기반 화상검색 시스템 구현", 한국지리정보학회 논문지, 제 4권, 제 4호, pp. 933-943, 1997.  
 [5] H.C Lin, L. L. Wang and S. N. Yang, "Regular-texture image retrieval based on texture-primitive extraction," *Image and Vision Computing* 17, pp. 51-63, 1999.  
 [6] M. Tuceryan and A. K. Jain, *Texture Analysis, Handbook of Pattern Recognition and Computer Vision*, World Scientific, 1993.  
 [7] B. M. Mehtre, M. S. Kankanhalli and W. F. Lee, "Shape Measures for Content Based Image Retrieval: A Comparison," *Information Processing & Management*, Vol. 33, No. 3, pp. 319-337, 1997.  
 [8] T. Chang and C. C. J. Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Trans. on Image Processing*, Vol. 2, pp. 4298-441, April, 1993.  
 [9] M. C. Lee and C. M. Pun, "Texture Classification Using Dominant Wavelet Packet Energy Features," *Proc. Image Analysis and Interpretation, 4th IEEE Southwest Symposium*, pp. 301-304, 2000.  
 [10] J. Guo, A. Zhang, E. Remias and G. Sheikholeslami, "Image Decomposition and Representation in Large Image Database Systems," *Journal of Visual Communication and Image Representation*, Vol. 8, No. 2, pp. 167-181, June, 1997.  
 [11] C. K. Li and H. Yuen, "A High Performance Image Compression Technique for Multimedia Applications," *IEEE Trans. on Consumer Electronics*, Vol. 42, No. 2, pp. 239-243, May, 1996.  
 [12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 2nd Ed.*, Addison-Wesley Publishing Co., 1992.  
 [13] Z. Lei, L. Fuzong and Z. Bo, "A CBIR Method Based on Color-Spatial Feature," *TENCON99, Proc. of the IEEE Region 10 Conference*, Vol. 1, pp. 166-169, 1999.  
 [14] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.



**최 창 규**

1998년 2월 대구대학교 전자계산학과 졸업(공학사). 2000년 2월 경북대학교 컴퓨터공학과 졸업(공학석사). 2000년 ~ 현재 경북대학교 컴퓨터공학과 박사과정 관심분야는 이미지/비디오 처리, MPEG 표준, 멀티미디어 데이터베이스



류 상 루

1988년 경북대학교 전자공학과 졸업(공학사). 1990년 경북대학교 대학원 컴퓨터공학과 졸업(공학석사). 1997년 경북대학교 대학원 컴퓨터공학과 졸업(공학박사)  
1998년 ~ 현재 청운대학교 컴퓨터과학과 조교수. 관심분야는 알고리즘 및 계산

이론, 멀티미디어 등

김 승 호

정보과학회논문지 : 컴퓨팅의 실제  
제 8 권 제 1호 참조