사분트리 분할 인덱스를 이용한 컬러이미지 검색

오석영°, 홍성용, 나연묵
단국대학교 전자컴퓨터공학과
osysy@dankook.ac.kr, {syhong, ymnah}@dku.edu

Color Image Retrieval using Quad-tree Segmentation Index

Sukyoung Oh°, Sungyong Hong, Yunmook Nah
Department of Computer Engineering, Dankook University

요 약

최근, 이미지 검색기법에서는 객체추출 방법이나 관심영역 추출방법에 관한 연구가 활발히 이루어지고 있다. 그러나, 컬러 이미지의 경우 색상을 고려한 관심영역 특징추출 방법이나 인덱스 기법은 많이 연구되지 못하고 있다. 따라서, 본 논문에서는 컬러 이미지의 색상을 기반으로 하는 사분트리 분할 인덱스 기법을 제안한다. 사분트리 분할 인덱스 구조는 컬러 이미지의 공간 영역을 계층적인 영역으로 분할하여 각 공간 영역의 평균 색상 값을 데이터베이스에 저장한다. 저장되어진 각 영역의 평균 색상은 검색의 효율성을 높이기 위해 사분트리 디스턴스(*Quad-tree distance*)를 퍼지 값으로 계산하여 인덱스를 생성한다. 생성된 사분트리 분할 인덱스는 컬러 이미지의 관심영역(*Region of Interest*)의 색상을 검색할 때 유용하게 사용되며, 검색속도의 향상에 도움을 준다.

1. 서 론

이미지 검색기법은 다양한 이미지를 효율적으로 저장하여, 저장된 이미지 또는 메타데이터를 최소화하고, 이 범위 내에서 정확성을 보장하고, 고속화하기 위해 다양한 인덱스 기법이 연구되고 있다.

의미기반 또는 내용기반 검색을 위한 이미지의 색상정보를 데이터베이스에 저장하기 위해서는 우선 이미지의 색상(Color)이나 질감(Texture), 모양(Shape) 등을 추출하고, 이 정보의 전체 또는 압축된 정보를 데이터베이스에 저장한다. 저장된 정보를 이용하여 이미지의 전체 영역 및 관심 영역 비교 등의 다양한 이미지의 유사도를 측정할 수 있다. 그러나, 데이터베이스에 저장되기 위한 색상 정보의 공간적 영역 비교를 위해 고정된 해상도로 양자화 하여 저장하며, 데이터베이스에 저장되기 위한 칼라 히스토그램은 위치 또는 크기에 안정적이지 못하고, 정보를 축소 할수록 원래의 정보와는 큰 차이를 보이므로, 영역 단위로 세분화할 필요가 있다. 그래서, 사분트리를 사용한 이미지 검색방법이 인덱스방법 중 가장 활발하게 연구되어 왔다[1].

따라서, 본 논문에서는 컬러 이미지의 색상을 기반으로 한 사분트리 분할 인덱스 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 이미지 검색방법에 관한 관련 연구를 알아보고, 3장에서는 본 논문에서 제안하는 색상추출 방법에 대해 설명한다. 4장에서는 특징벡터를 추출하기 위한 쿼드트리 인덱스 기법 및 질의처리 방식을 설명하고, 마지막으로 5장에서는 결론과 향후 연구 방향을 제시한다.

2. 이미지 검색 기법

2.1 이미지 인덱스 구조 및 검색도구

시스템 또는 웹상의 수많은 이미지와 정보를 기술하기 위한 툴과 데이터베이스 저장을 위한 인덱스 기법의 연구가 활발하다.

*IBM*의 *QBIC*(*Query By Image Content*) 시스템은 R*트리 인덱스로 구성하였고[2], *UC Berkeley* 대학에서 Michael Stonbraker 교수의 주도로 개발된 *Chabot* 시스템의 데이터베이스 시스템 *POSTGRES*에서는 B트리와 R트리 인덱스 구조를 사용하였다[3]. 이미지에 대한 인덱스 구조는 검색속도 뿐만 아니라, 정확성에도 영향을 미치므로, 많은 연구가 활발히 이루어지고 있다.

2.2 이미지 영역 분할 및 관심영역 추출방법

수많은 이미지를 효율적으로 저장하기 위해 이미지의 특징을 공간적 구성, 유사성 등에 따라 분할하여 저장하고 처리하는 연구가 진행되어 왔다.

보스턴 대학에 의해 연구된 *ImageRover* 시스템에서 이미지 고정 분할기법을 사용하였고[4], 코넬 대학에서 연구한 *color correlogram*에서는 임의의 영역 내에서 공간적 특성치의 상관관계를 나타내는 확률정보로 특징을 추출하였다[5].

3. 이미지 색상 추출 방법

컬러 이미지를 구성하는 색상(Color)은 물체를 식별하기 위한 가장 단순하고, 중요한 특성이다. 일반적으로 색상을 표현하는 방법에는 RGB, CMY 그리고 HSI공간의 표현 방법이 있다.

이미지를 구성하는 색상 공간을 구성하는 픽셀은 R

(red, 적색), G(green, 녹색), B(blue, 청색) 채널로 구성된다. 이때, 컴퓨터에서 디스플레이 장비가 이 세 가지 색상의 주사선을 쏘아 다양한 색상의 픽셀로 출력하기 위한 기법으로 각 채널은 최소 0에서 최대 255까지의 값을 가지며, 세 채널의 최대치인 255의 값으로 조합하면 흰색, 0으로 조합하면 검은색이 되므로, 가산모델이라고 한다. RGB와 상대되는 모델인 CMY는 종이 등에 C(cyan, 청녹색), M(Magenta, 자홍색), Y(yellow, 노랑색) 등의 색을 혼합하여 인쇄하기 위한 모델로, 감산모델이라고 한다. HSI 모델은 인간의 시각체계와 유사한 모델로 색을 표현하기 위한 모델로, 색의 순수성을 나타내는 0도에서 360도까지의 범위를 가지는 H(Hue, 색상), 색의 농도를 나타내는 S(Saturation, 채도), 밝기를 나타내는 I(Intensity, 명도)로 구성된다. 표 1에서 이 세 모델에 대한 값을 비교 분석하여 보여주고 있다.

	RGB	CMY	HSI
Red	255,0,0	0,255,255	0°,255,85
Green	0,255,0	255,0,255	120°,255,85
Blue	0,0,255	255,255,0	240°,255,85
Cyan	0,255,255	255,0,0	180°,255,383
Magenta	255,0,255	0,255,0	300°,255,383
Yellow	255,255,0	0,0,255	60°,255,383

표 1. 색상 공간에서의 각 모델의 비교

그림 1에서는 공간적 영역 분할 방법을 선택하고, 이 방법에 따라 평균 RGB를 추출하여 CMY나 HSI 등의 색상모델로 변환하고, 이 모델을 데이터베이스에 저장하여, 쿼리를 수행한다. 그러나, 공간 분할방법과 데이터베이스 스키마 설계의 연관성이 적절하지 못한 경우 효율적인 검색을 기대하기 어렵고, 검색속도가 지연되며, 잘못된 결과가 수행되게 된다. 따라서, 본 논문에서 이미지를 구성하는 모든 픽셀은 RGB 모델을 사용하며, 각 이미지를 64 X 64 픽셀로 조정하여 균등하게 나누고, 이에 대한 사분트리로 분할하여 상향식(*Bottom up*)으로 레벨 단위의 계층적 테이블을 구성하고, 각 사분 노드 단위로 평균 RGB를 저장한다.

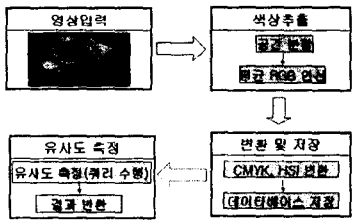

그림 1. 색상 추출 및 유사도 측정과정

그림 2에서 사분트리의 인덱스 구조를 설명하고 있다. 사분트리의 인덱스는 가장 좌측을 레벨1로 하고, 우측으로 갈수록 각각의 하위 레벨의 인덱스가 된다. 각 영역을 그림과 같이 4개로 분할하고 인덱스를 각각 *"A"*, *"B"*, *"C"*, *"D"* 로 정의하고, 이 과정을 주어진 최하 레벨까지 반복하며, 최상위 레벨 이후의 인덱스를 함께 사용한다. 예를 들어, 레벨1의 "A"를 레벨2로 분할하면, 각각 *"AA"*, *"AB"*, "AC", *"AD"*가 되는 것이다.

따라서, 알고리즘이 복잡하지 않은 장점이 있으나, 사분트리의 단점으로 관심영역을 사분트리로 나누어 최대 레벨의 사분트리로 대치할 경우, 해당하는 모든 레벨을 검색해야 하는 단점을 해결하기 위해, RGB 정보와 함께, 사분트리 내의 노드 중 평균값에 유사한 노드를 찾기 위해 동일 사분트리 내의 형제(*Sibling*) 노드와의 RGB 값의 차를 계산한 사분트리 디스턴스(*distance*)를 계산하여 퍼지(*fuzzy*)값으로 저장한다.

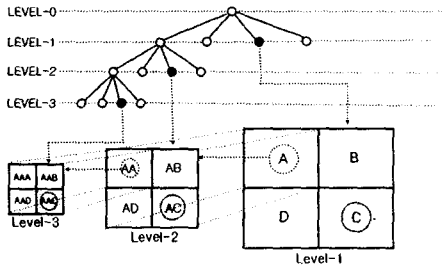

그림 2. 사분트리의 인덱스구조

4. 사분트리 분할 인덱스 기법

4.1 인덱스 구조

그림 3.(a)에서 각 사분트리 노드에서 해당 노드를 *"C_0"*, 시계방향으로 각각 *"C_1"*, *"C_2"*, *"C_3"*라 한다. 그림에서 변수 *"fstrall"*은 데이타베이스 내에서 레벨1 부터 현재 레벨까지의 사분트리의 인덱스이고, *"fstr"*은 현재 레벨의 인덱스, *"nfstr"*은 *"fstrall"*에서 *"fstr"*을 제외한 나머지 인덱스이다. 그림 3.(b)와 같이 형제 노드와의 색상 차에 해당하는 *"$D_1(C_0)$"*, *"$D_2(C_0)$"*, *"$D_3(C_0)$"*를 구하고, 평균 *"$D_{avg}(C_0)$"*를 구하며, 그림 3.(c)의 구조로 데이터베이스에 저장한다. 이때, *"OID"*는 이미지의 고유 식별자이고, *"F_N"*은 파일이름, *"Qi"*는 사분트리 인덱스, *"D_1"~"D_3"*과 *"D_{avg}"*는 각각의 사분트리 디스턴스이다.

```
fstrall:=right(Qi, length(Qi));     //현 레벨의 사분트리 인덱스,
                                    "A"-"D"
if length(Qi)=1 then nfstr:=''
   else nfstr:=copy(Qi ,1, length(Qi)-1);   // 전 레벨 사분 인덱스
if fstr='A' then begin
   C0:=Qi; C1:=nfstr+'B'; C2:=nfstr+'C'; C3:=nfstr+'D';
end else if fstr='B' then begin
   C0:=Qi; C1:=nfstr+'C'; C2:=nfstr+'D'; C3:=nfstr+'A';
end else if fstr='C' then begin
   C0:=Qi; C1:=nfstr+'D'; C2:=nfstr+'A'; C3:=nfstr+'B';
end else if fstr='D' then begin
   C0:=Qi; C1:=nfstr+'A'; C2:=nfstr+'B'; C3:=nfstr+'C';
end else begin end;
```

(a) 계산 영역 *"C_0"~"C_3"* 의 설정

$$D_1(C_0) = \frac{|R(C_1-C_0)| + |G(C_1-C_0)| + |B(C_1-C_0)|}{max(R)+max(G)+max(B)}$$

$$D_2(C_0) = \frac{|R(C_2-C_0)| + |G(C_2-C_0)| + |B(C_2-C_0)|}{max(R)+max(G)+max(B)}$$

$$D_3(C_0) = \frac{|R(C_3-C_0)| + |G(C_3-C_0)| + |B(C_3-C_0)|}{max(R)+max(G)+max(B)}$$

$$D_{avg}(C_0) = \frac{D_1(C_0)+D_2(C_0)+D_3(C_0)}{3}$$

(b) 사분 디스턴스 계산

레벨	데이터베이스 스키마									
L_0	Oid	F_N	R	G	B					
	000001	628.jpg	101	74	16					
L_1	Oid	F_N	Qi	R	G	B	D_{avg}	D_1	D_2	D_3
	000001	628.jpg	A	116	84	14	0.04532	0.00392	0.06667	0.06536
L_2	Oid	F_N	Qi	R	G	B	D_{avg}	D_1	D_2	D_3
	000003	628.jpg	AC	210	139	9	0.26928	0.37256	0.31503	0.12026
L_3	Oid	F_N	Qi	R	G	B	D_{avg}	D_1	D_2	D_3
	000009	628.jpg	ACA	210	150	13	0.26405	0.02745	0.38824	0.08758
	000010	628.jpg	ACB	222	143	7	0.27930	0.40523	0.38301	0.04967
	000011	628.jpg	ACC	213	134	8	0.15163	0.35317	0.04837	0.04837
	000012	628.jpg	ACD	194	129	9	0.15686	0.04444	0.07843	0.34771
	000029	628.jpg	BDA	220	146	8	0.18519	0.41176	0.11373	0.03007
	000029	628.jpg	BDA	220	146	8	0.18519	0.41176	0.11373	0.03007
	000030	628.jpg	BDB	218	162	12	0.15556	0.12680	0.01961	0.32026
	000031	628.jpg	BDC	210	146	8	0.23747	0.01961	0.29412	0.39869
	000032	628.jpg	BDD	209	132	8	0.24488	0.27451	0.37908	0.08105

(c) 데이터베이스 스키마 및 예제

그림 3. 사분 디스턴스 및 데이터베이스 스키마

4.2 칼라 이미지 검색기법

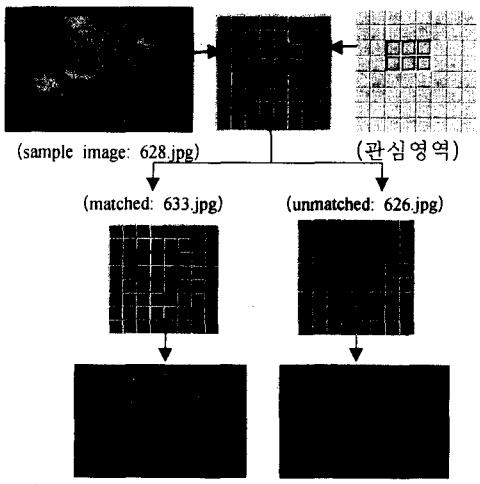

그림 4. 샘플 이미지에 의한 질의에 의한 관심영역 비교

그림 4에서 샘플 이미지의 관심영역에 대한 사분트리 인덱스 노드는 "ACA", "ACB", "ACC", "ACD", "BDA", "BDD"이고, "ACA", "ACB", "ACC", "ACD"는 레벨2의 "AC"로 압축되므로, 검색할 노드는 "AC", "BDA", "BDD"가 된다. 이때, 샘플 이미지인 "626.jpg"의 RGB는 (210, 139,9), (220,146,8), (209,132,8)이고, "633.jpg"는 (197,110,14), (213,111,8), (211,113,7), "626.jpg"는 (202,26,25), (198,16,16), (192,20,21)이지만, 레벨 2와 레벨3의 사분트리 테이블을 검색하게 되므로 비효율적인 수행시간을 보이게 된다. 사분트리 디스턴스를 사용하면, "AC"의 하위 레벨에서 사분트리 디스턴스가 가장 낮은 사분트리 노드의 인덱스는 모두 "ACC"가 되고, 질의 결과는 (213,134,8), "633.jpg"는 (196,90,10), "626.jpg"는 (196,21,23)이 되므로, "633.jpg"가 "626.jpg" 보다 유사도가 크다.

5. 결론 및 향후 연구방향

시스템 또는 웹상의 컬러이미지양이 다량화되고, 검색기법 또한 내용기반, 의미기반 중심으로 연구가 되면서, 특징추출을 위한 인덱스기법의 연구가 함께 병행되어야 하며, 특징추출 알고리즘 이상의 연구 활동이 필요하게 되었다.

본 논문에서는 사분트리에 대한 각 노드의 RGB를 계산한 색상추출방법 및 사분트리 인덱스 기법과, 사분트리 디스턴스에 의해 검색 테이블의 범위를 축소하여 컬러이미지 검색의 효율성을 제시하였다. 본 논문에서는 레벨0에서 레벨3까지 테이블을 생성하여 유사도를 비교하였다.

향후 연구로 객체에 대한 가변영역 추출을 위해 색상 유사도에 따라 사분트리 디스턴스와 유사한 픽셀의 8방향성 색상 유사도에 따라 외곽선을 추출하고, 이를 체인코드 스트링으로 바꾸어 R트리, R*트리 등의 더욱 향상된 인덱스방법에 적용하기 위한 연구가 필요하다.

6. 참고문헌

[1] A.Cassinelli, M.Naruse, M.Ishikawa "Quad-tree image compression using reconfigurable free-space optical interconnections and pipelined parallel processors," Proc. of the Options in Computing conference, Grand Hotel Taipai (Taiwan) , pp.23-25, April 8-11 2002.

[2] W.Niblack, R.Barber, W.Equitz, M.Flickner, E.Glasman, D.Petkonovic, P.anker and C.Faloutsos, "The QBIC Project : Querying images by content using color, texture, and shape," in Proc. SPIE Storage and Retrieval for Image and video Database, pp.177-187, February 1993.

[3] Virgrnia E.Ogle and Michael Stonebraker, "Chabot : Retrieval from a Relational Database of Images," IEEE Computer, vol.28, No.9, pp.40-48, September 1995.

[4] Stan Sclaroff, Leonid Taycher, Marco La Cascia, "ImageRover : A Content-Based Image Browser for the World Wide Web," IEEE Computer, May 1997.

[5] Jing Hwang, S.Ravi kumar, Mandar Mitra, Wei-Jing Zhu, Ramin Zabih, "Image indexing using color correlogram," IEEE Computer, June 1997.