

# 확장된 개념 기반 이미지 검색 시스템

## (An Extended COnccept-based Image Retrieval System : E-COIRS)

김 용 일<sup>†</sup> 양 재 동<sup>\*\*</sup> 양 형 정<sup>\*\*\*</sup>  
(Yong-Il Kim) (Jae-Dong Yang) (Hyoung-Jeong Yang)

**요 약** 본 논문에서는 사용자가 개념과 개념을 보다 정제된 이미지의 특징 정보를 이용하여 질의 할 수 있는 확장된 개념 기반 이미지 검색 시스템(E-COIRS)을 설계하고 구현하였다. 예를 들어, E-COIRS는 "응접세트 북쪽에 검은색 가전제품이 있는 이미지를 검색하라"와 같은 질의를 지원한다. 질의에는 '가전제품'과 같은 IS-A개념과 '응접세트'와 같은 복합 개념이 포함된다. 이러한 질의 처리를 위해 E-COIRS는 3가지 구성 요소, 가시적 이미지 색인기, 시소러스, 그리고 질의처리기를 포함한다. 가시적 이미지 색인기에 의해 인식된 각 객체쌍은 두 객체의 식별자(oid)와 그들간의 공간 관계로 구성되는 트리플로 변환된다. 각 객체에 대한 이미지의 특징 정보는 oid에 의해 참조 가능하다. IS-A 개념은 질의 처리기에서 퍼지 용어 시소러스를 사용하여 인식하며, 복합 개념은 트리플 시소러스에 의해 인식된다. 질의처리기는 역화일과 CS-Tree를 조회하여 사용자 질의의 각 트리플과 관련이 있는 이미지 집합을 얻는다. 색상과 같은 고차원 특징 벡터를 효율적으로 저장하고 빠르게 검색하기 위해서는 셀 기반 시그니처 트리(CS-Tree)를 사용하였다. E-COIRS는 개념이나 이미지의 특징만을 지원하는 다른 검색 시스템에 비해 한 단계 진보된 이미지 검색 시스템이다.

**키워드** : 개념기반이미지검색, 내용기반이미지검색, 시소러스, 퍼지공간관계

**Abstract** In this paper, we design and implement E-COIRS enabling users to query with concepts and image features used for further refining the concepts. For example, E-COIRS supports the query "retrieve images containing black home appliance to north of reception set." The query includes two types of concepts: IS-A and composite. "home appliance" is an IS-A concept, and "reception set" is a composite concept. For evaluating such a query, E-COIRS includes three important components: a visual image indexer, thesauri and a query processor. Each pair of objects in an image captured by the visual image indexer is converted into a triple. The triple consists of the two object identifiers (oids) and their spatial relationship. All the features of an object is referenced by its oid. A composite concept is detected by the triple thesaurus and IS-A concept is recognized by the fuzzy term thesaurus. The query processor obtains an image set by matching each triple in a user query with an inverted file and CS-Tree. To support efficient storage use and fast retrieval on high-dimensional feature vectors, E-COIRS uses Cell-based Signature tree(CS-Tree). E-COIRS is a more advanced content-based image retrieval system than other systems which support only concepts or image features.

**Key words** : Concept-Based Image Retrieval, Content-Based Image Retrieval, Thesaurus, Fuzzy Spatial Relationship

### 1. 서 론

인터넷의 보급과 컴퓨터 저장매체 및 하드웨어의 발달은 이미지 정보를 폭발적으로 증가시키고 있으며, 이러한 이미지 정보는 과학, 산업, 의학, 오락 등 다양한 응용 분야에서 폭 넓게 이용되고 있다. 이에 따라, 방대한 이미지를 효율적으로 저장하고 검색하기 위한 많은 관련 연구가 대학, 연구소 그리고 기업체에서 진행되어 왔다[1].

· 본 논문은 학술진흥재단 과제인 2000-042-E00045 '개념기반 이미지 정보검색 시스템의 구현'의 지원으로 이루어졌습니다.

† 종신회원 : 호남대학교 인터넷학부 교수  
yikim@honam.ac.kr

\*\* 정 회 원 : 전북대학교 컴퓨터과학과 교수  
jdyang@cs.chonbuk.ac.kr

\*\*\* 비 회 원 : 동신대학교 컴퓨터학과 교수  
hjiyang@blue.dongshinu.ac.kr

논문접수 : 2001년 12월 17일

심사완료 : 2002년 2월 19일

이미지 정보 검색을 위한 가장 간단한 방법은 이미지의 내용을 수동으로 기술한 주석으로 찾는 것이다. 그러나 이 방법을 적용하기에는 이미지들의 양이 너무 방대하며, 또 한 이미지조차도 수천 단어에 해당하는 의미를 내포하고 있기 때문에 이를 모두 기술한다는 것은 너무 힘든 일이다. 또한, 검색 결과가 색인자의 주관에 좌우되는 문제점이 있다[2].

이를 해결하기 위해 이미지에 포함되어 있는 특징을 자동으로 추출하여 이를 색인과 검색의 속성으로 이용하는 내용기반 이미지 검색에 대한 연구가 필요하게 되었다. 이미지 색인 속성으로 색, 질감, 모형, 윤곽선, 공간 관계 등 이미지의 주요 특징을 사용하여, 이들을 이용하는 내용기반 이미지 검색 연구가 다각도로 수행되었다[2]. 예를 들어, Photobook[3], QBIC[4], 그리고 Netra[5] 시스템은 이미지로부터 특징을 추출하여 질의에서 요구하는 속성과의 부합에 의한 검색을 지원한다. 이 연구들의 단점은 사용자의 질의와 외향적으로 유사한 이미지 정보만을 검색한다는 데 있다.

본 논문에서는 개념기반 이미지 검색 시스템(COIRS)[6]의 개념기반 질의에 퍼지 공간 정보와 색상 정보도 명시할 수 있도록 기능을 확장한 이미지 정보 검색 시스템(E-COIRS)을 설계하고 구현하였다. E-COIRS는 개념간의 공간 관계에 기반한 질의와 각 개념에 특징 정보를 함께 혼합하여 질의할 수 있다는 점에서 다른 이미지 검색 시스템들과 다르다. 예를 들어, "붉은색 응접세트 북쪽에 가전제품이 있는 그림을 검색"하라는 질의에 대해 붉은색 소파와 탁자를 포함하고 가전제품의 모든 하위 개념들이 북쪽에 위치하고 있는 이미지들이 검색된다. E-COIRS는 가시적 이미지 색인기, 시소러스, 그리고 질의처리를 포함한다. 가시적 이미지 색인기는 이미지 객체들을 인식하는데 사용되며, 인식되는 두 객체는 각 객체의 식별자(oid)와 그 들간의 공간 관계가 명시된 트리플로 표현된다. 각 객체에 대한 특징 정보는 oid에 의해 참조 가능하다. 시소러스는 이미지 인덱싱과 질의 처리과정에서 개념을 처리하기 위해서 사용된다. 이 시소러스는 트리플 시소러스와 퍼지 용어 시소러스 두가지 종류가 있는데, IS-A 개념은 퍼지 용어 시소러스를 사용하여 인식하며, 복합 개념은 트리플 시소러스에 의해 인식된다. 질의처리기는 질의내 각 트리플과 역화일에 있는 트리플간의 정합에 의해 관련 후보 이미지 집합을 얻은 후, 질의에 포함된 특징 정보와 역화일에서 검색된 후보 이미지 집합의 특징 정보 사이의 유사도를 계산하여 결과 이미지 집합을 반환한다. 한편, 색상과 같은 고차원 특징 벡터를 저장하고 효

율적으로 검색하기 위해서는 셀 기반 시그니처 트리(CS-Tree)[19]가 사용된다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 내용기반 이미지 시스템과 공간 관계 및 색상 정보 처리에 대한 관련 연구를 기술하였다. 3장에서는 E-COIRS에서 사용되는 트리플 구조를 제안하였으며, 개념 처리를 위해서 사용된 퍼지 용어 시소러스와 트리플 시소러스에 대해서 4장에서 기술하였다. 5장에서는 퍼지 기반 공간 관계와 색상 특징 정보의 추출 및 저장에 대해서 기술하였다. 질의 처리 과정에 대해서 6장에서 서술하였으며, 7장에서는 E-COIRS 시스템의 구현 결과 및 실험 결과를 보였다. 끝으로 8장에서는 결론 및 향후 연구 방향을 제시하였다.

## 2. 관련 연구

### 2.1 통합 내용기반 이미지 검색시스템

QBIC(Query By Image Content)은 IBM사에서 개발한 내용기반 이미지 검색 시스템 중의 하나이며, 사용자 질의는 색상, 질감, 모형, 사용자 스케치, 그리고 시공간(spatiotemporal) 관계를 기반으로 하고 있다. Photobook은 QBIC과 많은 부분이 비슷하며, 한 이미지 또는 비디오 DB로부터 이미지를 검색 및 브라우징 할 수 있도록 MIT 대학의 Media Lab.에서 개발된 대화형 도구이다. Netra는 Santa Barbara의 California 대학에서 개발한 이미지 검색 시스템으로 이미지를 인덱싱하기 위해 분할된 지역적인 영역을 사용하였다. 이러한 내용기반 이미지 검색시스템들의 공통적인 문제점은 사용자의 질의와 외향적으로 유사한 이미지 정보만을 검색한다는 데 있다. 한편, 일본 NEC C&C 연구실에서 개발한 SEMCOG[7]는 객체기반 멀티미디어 시스템으로 이미지를 계층적으로 표현하며, 이미지내의 객체를 컴포넌트라 정의하였다. 컴포넌트는 이미지의 특징과 의미(semantics)를 포함하는 2개의 구성요소로 구분하였다. 사용자의 개념 질의를 확장하기 위해 용어 관리자를 이용하였으며, 공간 관계는 2D String으로 나타내었다. 그러나 공간 관계나 개념 처리에 퍼지 개념이 이용되지는 않았다는 점에서 E-COIRS와는 다르다.

### 2.2 공간 관계를 사용하는 이미지 검색

이미지내 객체들 사이의 공간 관계를 이용한 이미지 정보 검색 연구로는 2D 스트링[8], 순서 트리플(ordered triple)[9], 기호 배열(symbolic array)[10] 등이 있다. S. K. Chang에 의해 고안된 2D 스트링은 심볼화된 이미지 내의 공간 관계를 두개의 일차원 스트링 형태로 표현하며, 검색시 질의 2D 스트링과 이미지로부터 생성

된 모든 2D 스트링을 일일이 비교해야 한다는 단점이 있다. C. C. Chang은 2D 스트링의 단점을 해결하기 위해 두 개의 객체와 공간 관계를 명시하는 순서 트리플을 색인 방법으로 제안하였다. 공간 관계는 북쪽에서 동북쪽까지 시계 반대 방향으로 1부터 8의 정수로 표현하고, 같은 위치에 있을 경우 9로 표현한다. 예를 들어, A라는 객체의 동남쪽에 위치한 B는 <A, B, 6>의 트리플로 표현된다. 그러나, 두 방법 모두 객체를 심볼화하여 눈금 위의 한 셀에 할당하고 이를 기반으로 공간 관계를 정하기 때문에 각도에 의한 정확한 검색을 지원하지 못하고 또한 객체와 의미적으로 유사한 객체를 고려한 검색을 지원하지 못한다는 단점을 가지고 있다. D. Papadias는 이미지 정보내의 공간 관계성을 기호 배열(symbolic array)을 이용하여 표현하고 있다. 기호 배열에 나타나는 심볼은 상대적인 크기나 거리 정보를 배제한 공간 위치 정보만을 표현하며, 하나의 셀이 다른 기호 배열로 매핑되는 다단계 계층 구조를 갖는다. 따라서, 공간 개념들의 상대적인 위치를 가지고 이미지를 검색하는 응용에는 부적합하다.

**2.3 색상에 기반한 이미지 검색**

이미지나 비디오 데이터베이스로부터 적절한 정보를 검색하기 위해 특징 정보에 대한 많은 연구가 진행되어 왔다. 특히, 색상 정보는 가장 대표적이면서 사람이 쉽게 분별할 수 있는 특징 정보 중 하나로, 이를 나타내기 위해 색상 히스토그램(color histogram)[11], 색상 모멘트(color moments)[12], 그리고 색상 상관도표(correlogram)[13] 등 다양한 기술자가 제안되었다.

색상 히스토그램은 내용 기반 이미지 검색 시스템에서 색상을 표현하는 방법으로 가장 광범위하게 사용된다[11]. 그러나 색상 히스토그램이 갖는 고 차원(high-dimensionality) 정보는 효율적인 색인과 검색시 장애가 된다. 따라서, 색상 내용에 최소한의 손실을 가지고 히스토그램의 차원을 줄이기 위해서는 색상 공간 양자화(quantization)가 필수적이다. E-COIRS에서는 색상 특징 정보를 색인하기 위해 색상 히스토그램을 사용하였으며, 전체 색상 공간을 64개의 bin(bin)으로 양자화하여 색상 특징 벡터로 사용하였다.

또한, 색상과 같은 다차원 특징 벡터를 색인하는 방법도 활발히 연구되고 있다. 다차원 특징 벡터를 색인하는 방법으로 SS-트리(Similarity Search tree)[14], SR-트리[15], TV-트리[16], X-트리[17]가 있다. E-COIRS의 색인 방법은 기존의 트리-기반 색인 구조에 시그너처를 적용한 방법으로, 다차원 특징 벡터 공간을 셀로써 분할하여 하나의 특징 벡터를 그에 해당되는 셀의 시그

니처로 표현한다.

**3. 퍼지 트리플 기반 이미지 인덱싱**

**3.1 퍼지 트리플 기반 이미지 색인**

본 논문에서는 이미지내 객체들 사이의 공간 관계를 표현하는 방법으로 트리플을 이용하였다. 트리플들은 해쉬 함수를 이용하는 역화일로 구성되고, 트리플의 형태로 주어진 질의와 역화일내 트리플간의 정합을 통해 질의와 유사한 공간관계를 갖는 이미지들을 검색하게 된다. 트리플 구조를 이용하여 객체간의 공간 관계를 표현하는 방식은 자료구조가 간단하면서도 빠른 검색을 지원한다는 장점이 있다. 다음은 트리플 구조에 대한 정의이다.

**정의 1.** IDB(Image Database)를 이미지 검색 시스템에 있는 모든 이미지들의 집합이라 하고,  $O_p(p \in IDB)$ 는 한 이미지  $p$ 에 나타나는 모든 객체들의 집합이라 하면,  $p$ 에 대한 트리플 집합  $T_p$ 는 다음과 같이 정의된다.

$$T_p = \{t = \langle o_i, o_j, a_{ij} \rangle \mid o_i, o_j \in O_p, a_{ij} \in [0^\circ, 359^\circ]\}$$

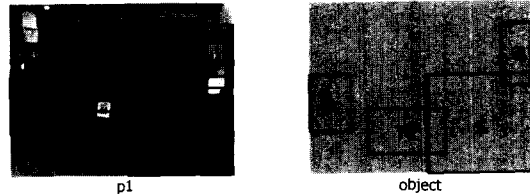


그림 1 이미지 p1

예를 들어, 그림 1의 이미지  $p_1$ 은 응접 탁자(rt), 소파(s), 에어컨(a), 그리고 벤치(b)를 포함하고 있다. 이미지  $p_1$ 을 색인한 트리플 집합  $T_{p1}$ 는  $\{\langle rt, s, 10 \rangle, \langle a, s, 240 \rangle, \langle b, s, 350 \rangle, \langle a, rt, 225 \rangle, \langle a, b, 205 \rangle, \langle b, rt, 340 \rangle\}$ 으로 표현된다.  $a_n$ 는  $a_{ij}$ 의 역을 나타내므로  $n$ 개의 객체로 구성된 이미지  $p$ 의 트리플 개수는  $n * (n - 1) / 2$ 이다.

객체 동치(equality)는 구현하기에 매우 어렵다고 알려져 있기 때문에, 객체 식별자를 용어로 사상하기 위한 함수가 필요하다. 다음 정의는 객체 식별자와 용어간의 관계를 제공하고 있다.

**정의 2.**  $O$ 는 객체들의 집합이고,  $N$ 이 용어들의 집합이면, 이름 함수( $f_{name}$ )는 다음과 같이 나타낸다.

$$f_{name}: O \rightarrow N$$

예를 들어, 이미지  $p_1$ 의 객체  $s$ 에 대해  $f_{name}(s) = '소$

파'이다. 본 논문에서는 표기를 간단하게 하기 위해  $f_{name}(o), o \in O$ 과  $o.name$ 을 동일하게 표현하도록 한다.

정의 1에서 객체간의 공간 관계는 각도로 표현하였다. 그러나, 공간관계가 각도로 표현된 트리플들을 역화일에 저장할 경우 서로 다른 이미지 내에 동일한 객체가 동일한 공간 관계를 갖는 이미지는 거의 없기 때문에 많은 저장 공간을 낭비하게 된다. 따라서 각도로 주어진 공간 관계를 8방위로 클러스터링하기 위한 함수가 필요하다.

**정의 3.**  $b_i = 45 * i, i = 1, \dots, 7$ 이고, 방향 공간 관계  $d_i \in D$ 는 동쪽(east)에서 반시계 방향으로 45도씩 증가한 각도에 대해 각각을 8방위로 정의하면, 객체간의 각도를 8방위로 클러스터링하는 함수( $f_{angToDir}$ )는 다음과 같이 정의할 수 있다.

$$f_{angToDir}(x) = \begin{cases} d_i, & b_i - 23 \leq x < b_i + 23 \text{ 일 때,} \\ d_0, & 338 \leq x \leq 360 \text{ or } 0 \leq x < 23 \text{ 일 때} \end{cases}$$

여기서 사용된 공간 관계의 클러스터링의 범위 설정을 23°로 한 이유는 5장의 K-weight 함수에서 자세히 설명하기로 한다.

예를 들어, 두 객체간의 공간 관계가 30° 인 경우  $d_1$ 이 되며, 따라서  $f_{angToDir}(30^\circ) = northeast$ 가 된다.

퍼지 트리플은 정의 2,3를 이용하여 트리플을 퍼지 용어와 퍼지 공간 관계로 표현한 것으로 역화일의 인덱스로 사용되며, 질의 처리 과정에서 의미적으로 유사한 이미지 검색을 지원한다. 퍼지 트리플과 퍼지 트리플 집합을 정의하면 다음과 같다.

**정의 4.**  $T_p$ 를 정의 1에 의한 트리플 집합이라 하자. 이때 한 이미지  $p$ 에 대한 퍼지 트리플 집합  $T'_p$ 는 다음과 같이 정의할 수 있다.

$$T'_p = \{ t' = \langle f_{name}(o_i), f_{name}(o_j), d_{ij} \rangle \mid d_{ij} = f_{angToDir}(a_{ij}), d_{ij} \in D, o_i, o_j \in O_p \}$$

예를 들어, 그림 1에서의 트리플  $\langle rt, s, 10^\circ \rangle$ 은 정의 4에 의해 퍼지 트리플  $\langle f_{name}(rt)=\text{웅접탁자}, f_{name}(s)=\text{소파}, east \rangle$ 가 된다. 따라서, 이미지  $p_1$ 의 퍼지 트리플 집합은  $T'_p = \{ \langle \text{웅접탁자}, \text{소파}, east \rangle, \langle \text{에어콘}, \text{소파}, southwest \rangle, \langle \text{벤치}, \text{소파}, east \rangle, \langle \text{에어콘}, \text{웅접탁자}, southwest \rangle, \langle \text{에어콘}, \text{벤치}, southwest \rangle, \langle \text{벤치}, \text{웅접탁자}, east \rangle \}$ 가 된다.

**3.2 역화일(Inverted File)**

이미지로부터 얻는 퍼지 트리플들은 해쉬 함수를 이용해서 역화일에 저장된다. 역화일은 공간 관계로 8방위를 갖는 트리플과 트리플에 의해 색인된 객체 정보로 구성된다. 객체 정보(s)는 색인된 트리플의 객체 정보와

객체간의 공간관계를 포함하며, 다음과 같이 정의한다.

**정의 5.** 객체  $o_i, o_j$ 가 이미지  $p \in IDB$ 의 객체이고,  $a_{ij} \in [0^\circ, 359^\circ]$ 가 두 객체간의 공간 관계이면, 객체 정보(s)는 다음과 같이 정의한다.

$$s = \langle p, o_i, o_j, a_{ij} \rangle$$

**정의 6.** 트리플  $t'$ 에 k개의 객체 정보가 삽입된 역화일(I)은 다음과 같이 표현된다.

$$I = \{ \langle t', (s_k) \rangle \mid t' \in T' \}$$

예를 들어, 이미지  $p_1$ 에 두개의 객체 '소파'와 '웅접탁자'가 190°의 방향으로 위치하고 있고, '소파'와 '웅접탁자'의 객체 식별자가 가시적 이미지 색인기에 의해 각각 s, rt로 할당되었다고 가정하면, 트리플  $t = \langle s, rt, 190^\circ \rangle$ 와 퍼지 트리플  $t' = \langle \text{소파}, \text{웅접탁자}, west \rangle$ 이 생성되고, 트리플 t를 기준으로 객체 정보  $\langle p_1, s, rt, 190^\circ \rangle$ 는  $\langle \text{소파}, \text{웅접탁자}, west \rangle$ 의 트리플에 추가된다. 이렇게 생성된 역화일은 사용자의 질의와의 정합을 통해 원하는 이미지를 검색하기 위해 질의 처리기에서 사용되며 구조는 그림 2와 같다.

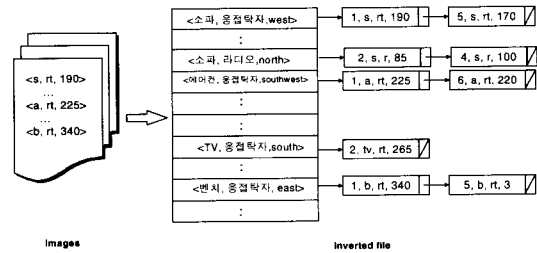


그림 2 역화일의 구조

**4. 개념기반 검색을 위한 시소러스**

본 논문에서는 한 이미지 내에 나타날 수 있는 객체들을 '라디오', '찬장'과 같은 단순 객체(primitive object)와 '부엌', '가전제품' 같은 개념 객체(concept object)로 구분하였다. 개념 객체는 복합 객체와 IS-A 객체로 분류된다. 복합 객체는 하나 이상의 단순 객체나 개념 객체들로 구성된다. 예를 들어, '부엌(Kitchen)'과 같은 복합 객체는 '주방세트(Kitchen Unit)'와 '식탁세트(Dining Set)'라는 두 개념 객체로 구성되며, 주방세트는 다시 '개수대(Sink)'와 '찬장(Cupboard)'이라는 두 단순 객체들로 구성될 수 있다. 한편, IS-A 객체는 여러 개의 단순 객체 또는 IS-A 객체를 하위 노드로 가진다. 예를 들어, '가전제품'은 'TV', '라디오', '에어컨' 같은 단순 객체들의 일반화된 IS-A 객체이다. 단순 객

체와 퍼지 시소러스의 트리플 규칙을 이용한 복합 객체는 인덱싱 과정에서 인식되어 역화일에 저장되며, IS-A 객체는 질의 처리 과정에서 퍼지 시소러스를 이용하여 처리한다.

4.1 퍼지 시소러스

그림 3은 소속성 함수 값으로 명시된 개념들을 퍼지 그래프로 표현한 퍼지 시소러스이다.

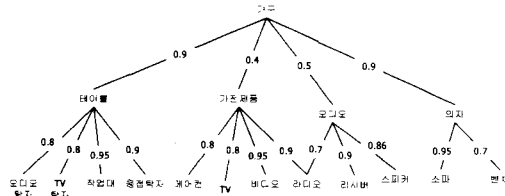


그림 3 퍼지 시소러스

여기서, 그래프의 단말노드는 단순 객체들이고, 중간 노드는 개념 객체를 나타낸다. 또한, 각 노드사이의 간선(edge)은 명시된 정도로 개념사이의 관련성 정도를 나타내고 있다. 예를 들어, '가구'는 '테이블', '가전제품', '오디오', '의자'와 같은 개념 객체를 멤버로 가지며, '가전제품'은 단순 객체인 'TV', '에어컨', '라디오' 등을 멤버로 가지는 개념 객체이다. 따라서 대응되는 멤버 함수는  $\mu_{가구} = \{테이블/0.9, 가전제품/0.4, 오디오/0.5, 의자/0.9\}$ ,  $\mu_{가전제품} = \{에어컨/0.8, TV/0.8, 비디오/0.95, 라디오/0.9\}$ 이다.

정의 7. F가 퍼지 용어이고, N이 용어들의 집합이면, 퍼지 멤버 함수( $\mu_F$ )는 다음과 같이 정의한다.

$$\mu_F : N \rightarrow [0, 1]$$

$$n, n' \in F \text{ 일 때 } \begin{cases} n = n' \text{ 이면, } \mu_n(n') = 1 \\ n \neq n' \text{ 이면, } \mu_n(n') = 0 \end{cases}$$

예를 들어,  $\mu_{가구}(가전제품)$ 는 0.4,  $\mu_{가전제품}(에어컨)$ 는 0.8, 그리고  $\mu_{에어컨}(에어컨)$ 는 1.0의 멤버 함수 값을 갖는다.

단순 객체가 하나 이상의 상위 개념 객체와 관련이 있는 퍼지 시소러스에서 두 객체 사이의 개념 관련 정도는 하나 이상의 값이 된다. 그림 3의 퍼지 시소러스에서 '가구' 개념 객체와 '라디오' 단순 객체 사이의 개념 관련 정도 값은 하나 이상이 된다. 따라서, 개념 사이의 개념적 관련 정도를 구하는 함수가 필요하며 다음과 같이 정의된다.

정의 8.  $a_i$ 는 퍼지 용어,  $n \in N$ 은 용어들의 집합이고,  $F \subset N$ 이면, 개념 사이의 개념적 관련 정도를 구하는 함

수 CR(Concept Relevance)은 다음과 같이 정의한다.

$$CR(F, n) = \begin{cases} \mu_F(n) & \mu_F(n) > 0 \text{ 일 때} \\ \max(\min(CR(F, a_i), CR(a_i, n)), \mu_F(a_i)) & \mu_F(a_i) > 0 \text{ 이고, } i \geq 1 \text{ 일 때} \end{cases}$$

예 1. 그림 3에서 '가구'와 '라디오'의 개념적 관련 정도는 다음과 같이 계산된다.

$$\begin{aligned} CR(\text{가구}, \text{라디오}) &= \max(\min(CR(\text{가구}, \text{가전제품}), CR(\text{가전제품}, \text{라디오})), \min(CR(\text{가구}, \text{오디오}), CR(\text{오디오}, \text{라디오}))) \\ &= \max(\min(\mu_{가구}(\text{가전제품}), \mu_{가전제품}(\text{라디오})), \min(\mu_{가구}(\text{오디오}), \mu_{오디오}(\text{라디오}))) \\ &= \max(\min(0.4, 0.9), \min(0.5, 0.7)) = 0.5 \end{aligned}$$

퍼지 시소러스는 질의 처리 과정에서 IS-A 객체를 확장하는데 사용된다. IS-A 객체의 확장과정은 IS-A 객체를 구성하고 있는 단순 객체들을 구한 후, 단순 객체들의 논리합으로 확장한다. 이를 위해서는 먼저 IS-A 객체로 일반화 될 수 있는 단순 객체들의 정보가 필요하다.

정의 9. X가 IS-A 객체라면, 객체 X가 될 수 있는 단순 객체들은 다음과 같이 정의한다.

$$X.\text{prim} = \{x \mid x \text{는 단순객체이고 } x \in N, \mu_X(x) > 0\}$$

예를 들어, 객체 h(h.name = '가전제품')는 그림 3 퍼지 시소러스의 IS-A 객체이므로 객체 h가 될 수 있는 단순객체 (h.name).prim는 {에어컨, TV, 비디오, 라디오}이다.

정의 10. X가 IS-A 객체이고, 객체 X로 일반화 될 수 있는 모든 단순 객체들의 집합을 X.(recurse)prim이라 하면, 이 집합은 다음과 같이 정의된다.

$$\begin{aligned} &\text{임의의 } c \in X.\text{prim} \text{에 대해서} \\ &c \in X.(recurse)\text{prim}, \quad c.\text{prim} = \emptyset \text{ 일 때} \\ &c.(recurse)\text{prim} \subseteq X.(recurse)\text{prim}, \text{ 그의} \end{aligned}$$

예 2. IS-A 객체 f.name = '가구'에 대한 f.(recurse)prim를 구하는 과정은 다음과 같다.

가구.prim={테이블,가전제품,오디오,의자}이고, 가구.prim에 속한 객체 '테이블'에 대해 테이블.prim≠∅이므로 테이블.(recurse)prim⊆가구.(recurse)prim이다. 따라서 테이블.prim={오디오탁자,TV탁자,작업대,응접탁자}이고, 테이블.prim에 속한 객체 '오디오탁자'에 대해 오디오탁자.prim=∅이므로, 테이블.(recurse)prim = {'오디오탁자'}가 된다. 테이블의 각 객체에 대해 반복하면 테이블.(recurse)prim = {'오디오탁자',TV탁자,작업대,응접탁자'}된다. 비슷한 방법으로 가구.prim에 속한 객체를 처리하면 최종적으로 그림 3의 최하위 노드들인 단순 객체들이 된다.

정의 11. X가 IS-A 객체이고,  $x_i \in X.\text{prim}$ 일 때, IS-A 객체를 포함하고 있는 트리플( $t = \langle b, X, d \rangle \in T$ )를

확장하는 함수는 다음과 같다.

$$f_{extISACon} : T \rightarrow T^+, T^+ = \bigvee_{i=1}^k \langle b, x_i, d \rangle$$

여기서  $k$ 는  $X$ 를 구성하는 단순 객체들의 개수를 나타낸다.

예를 들어, 트리플  $t = \langle h, s, east \rangle$ 은 정의 11과 그림 3의 퍼지 시소러스를 사용하여 IS-A 객체인 '가전제품 ( $h.name = \text{가전제품}$ )'에 대해  $\langle \text{에어콘, 소파, east} \rangle \vee \langle \text{TV, 소파, east} \rangle \vee \langle \text{비디오, 소파, east} \rangle \vee \langle \text{라디오, 소파, east} \rangle$ 로 확장된다.

**4.2 트리플 시소러스**

트리플 시소러스는 인덱싱 과정에서 도메인 전문가에 의해 색인된 객체들로부터 생성된 트리플에서 복합 객체를 탐지하는데 사용된다. 트리플 시소러스내 트리플 규칙은 하나 이상의 단순 객체 또는 개념 객체로 구성되는 복합 객체를 표현하는 방법으로, 복합 객체를 구성하는 객체들과 그들 간의 공간관계를 나타내는 트리플들의 논리연산자로 표현한다.

예를 들어, '개수대'와 '찬장'을 포함하고 있는 트리플 ( $\langle \text{개수대, 찬장, east} \rangle$ )은 그림4의 트리플 규칙(주방세트  $\leftarrow \langle \text{개수대, 찬장, east} \rangle$ )에 의해 '주방세트'라는 복합 객체를 생성하게 된다.

본 논문에서는 트리플 규칙에 사용되는 공간 관계를 2차원만을 고려하였다. 그러나 VRML과 같은 3차원 이미지를 검색하는 경우에는 위상 공간 관계가 필요하며, 이에 관련된 내용은 향후 연구로 넘기도록 한다.

**정의 12.**  $T_P$ 를 이미지 p에 대한 퍼지 트리플들의 집합이라 하면, 트리플( $t_i \in T_P$ )들의 논리곱인  $t_c$ 는 다음과 같이 정의한다.

$$t_c = t_1 \text{ AND } t_2 \text{ AND } \dots \text{ AND } t_s$$

**정의 13.**  $n \in N$ 을 복합 객체라 하고,  $T_C$ 를 트리플들의 논리곱의 집합이라 할 때, 트리플 규칙은 다음과 같이 정의한다.

$$n \leftarrow t_{c_1} \mid t_{c_2} \mid \dots \mid t_{c_m}$$

여기서  $t_{c_i} \in T_C, i = 1, \dots, m$ 이다.

그림 4의 '부엌'이라는 복합 객체는 '주방세트'와 '식탁세트'라는 개념 객체로 구성되며, 다시 주방세트는 '개수대'와 '찬장'이라는 단순 객체로 구성된다. 이를 방향 공간 관계를 고려하여 트리플 규칙으로 표현하면 다음과 같다.

부엌  $\leftarrow \langle \text{주방세트, 식탁세트, south} \rangle$

주방세트  $\leftarrow \langle \text{개수대, 찬장, east} \rangle \mid \langle \text{개수대, 찬장, north} \rangle$

식탁세트  $\leftarrow \langle \text{의자, 식탁, *} \rangle$

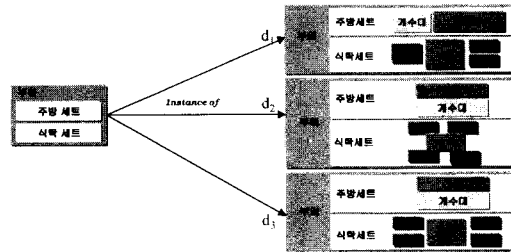


그림 4 복합 객체 확장

**정의 14.**  $T_C$ 를 트리플들의 논리곱의 집합이라 하면, 트리플 규칙들에서 복합 객체를 구하는 함수는 다음과 같다.

$$C_R : T_C \rightarrow N, \text{ 즉, } C_R(t_c) = n \in N$$

예를 들어, 트리플  $\langle \text{개수대, 찬장, north} \rangle$ 는 정의 14의 함수에 따라 '주방 세트'로 해석되며,  $C_R(\langle \text{개수대, 찬장, north} \rangle) = \text{'주방세트'}$ 로 표현한다.

정의 14에서 얻어진 복합 객체들을 검색에 이용하기 위해서는 이 객체와 이미지 내의 다른 객체간의 공간 관계를 고려한 새로운 트리플이 생성되어야 한다. 따라서, 이 공간 관계를 구하기 위해 복합 객체를 구성하고 있는 객체들의 정보가 필요하다.

**정의 15.**  $c$ 를 복합 객체라 하면 복합 객체를 구성하는 객체들은 다음과 같이 정의한다.

$$c.comp = \{ o_i, o_j \mid t_c \in T_C \text{에 대해 } C_R(t_c) = c.name \text{ 이고 } t_c = \langle o_i, name, o_j, name, d_{ij} \rangle \}$$

**정의 16.**  $c$ 가 복합 객체이고, 객체  $c$ 를 구성하는 모든 객체들의 집합을  $c.(recurse)comp \in 2^O$ 라 하면, 이 집합은 다음과 같이 정의된다.

- 1)  $c.comp \subseteq c.(recurse)comp$
- 2) 임의의  $o \in c.comp$ 에 대해서  $o \in c.(recurse)comp, C_R^{-1}(o.name) = \emptyset$ 일 때  $o.(recurse)comp \subseteq c.(recurse)comp$ , 그외
- 3)  $c.(recurse)comp$ 는 다른 어떤 원소도 포함하지 않는다.

**예 3.** 복합 객체  $k.name = \text{'부엌'}$ 에 대한  $k.(recurse)comp$ 를 구하는 과정은 다음과 같다.

$k.comp = \{ ku, dr \mid ku.name = \text{'주방세트'}, dr.name = \text{'식탁세트'} \}$ 이므로 1)에 의해  $k.(recurse)comp = \{ ku, dr \}$ 이 된다.

2)에서  $C_R^{-1}(ku.name) \neq \emptyset$ 이므로  $ku.(recurse)comp \subseteq k.(recurse)comp$ 이다. 따라서  $ku.comp = \{ s, cb \mid s.name = \text{'개수대'}, cb.name = \text{'찬장'} \}$ 이고,  $C_R^{-1}(s.name) = \emptyset$ ,

$C_R^{-1}(cb.name) = \emptyset$  이므로  $ku.(recurse)comp = \{s, cb\}$  이다. 비슷한 방법으로  $dr$  객체를 처리하면 최종적으로 객체  $k$ 에 대한  $k.(recurse)comp$ 는  $\{ku, dr, s, cb, t, c\}$ 이다.

복합 객체(X)와 다른 객체(b)간의 공간 관계를 계산하는 함수  $GenAvgDeg(b, X)$ 는 다음과 같다.

```

Function GenAvgDeg(X, b)
  Input : X (복합 객체),
         b (임의의 객체)
  Output : avgDeg (복합 객체 X와 객체 b 사이의 각도)

  O = X.(recurse)comp;
  convFlag = false;
  deg = FindDegBetween180And360(O, b, convFlag);
  ConvertDegree(deg, O, b, convFlag);
  minDeg = findMinDeg(O, b);
  maxDeg = findMaxDeg(O, b);
  avgDeg = CalculateAvgDeg(O, b);
  return avgDeg;
end.
    
```

FindDegBetween180And360 함수는 복합 객체를 구성하고 있는 객체들( $x_i$ )과 다른 객체(b)와의 각도가  $180^\circ$ 에서  $360^\circ$  사이에 존재하면 그중 가장 작은 값(deg)을 반환하며 convFlag 변수에 참(true)을 할당하는 함수이다. ConvertDegree 함수는 FindDegBetween180And360에서 반환된 convFlag 값이 참인 경우 복합 객체를 구성하고 있는 객체들( $x_i$ )과 객체(b)사이의 각도를 계산하는 함수이다. FindDegBetween180And360 함수에서 반환된 각도(deg)가 기준 각이 되므로 복합 객체를 구성하는 나머지 객체들( $x_i$ )과 다른 객체(b)사이의 각도는 기준 각을 중심으로 반 시계 방향으로 증가한다. 따라서,  $0^\circ$ 에서  $(deg - 180)^\circ$ 도 사이에 존재하는 각도는  $360$ 을 더해 각도를 변환한다. 또한, CalculateAvgDeg 함수는 변환된 각도에 대해 평균 값( $\frac{\sum_{i=1}^l a_{b,x_i}}{l} \bmod 360$ )을 계산하는 함수이다. 여기서  $a_{b,x_i}$ 는 객체  $x_i$ 와 객체 b가 이루는 각도이며,  $l$ 은 객체 X를 구성하는 객체의 개수를 나타낸다.

**예 4.** 그림 5의 객체 b와 복합 객체 X에 대해 생성되는 새로운 트리플을 구해보자.

먼저 단순 객체들에 대한 트리플 집합( $T_p$ )는  $\{ \langle b, x_1, 45 \rangle, \langle b, x_2, 5 \rangle, \langle b, x_3, 330 \rangle, \langle x_1, x_2, 90 \rangle, \langle x_2, x_3, 90 \rangle, \langle x_1, x_3, 90 \rangle \}$ 이다. 트리플 규칙( $X \leftarrow \langle x_1, x_2, north \rangle$  AND  $\langle x_2, x_3, north \rangle$  AND  $\langle x_1, x_3, north \rangle$ )에 의해 생성된 복합 객체 X와  $b \in X.(recurse)comp$ 사이의 공간 관계를 계산한 후 새로운 트리플을 생성한다. 복합 객체 X와 b사이의 트리플 중  $\langle b, x_3, 330 \rangle$ 이 조건( $180^\circ < a_{x_i, b} < 360^\circ$ )

을 만족하므로, 나머지 객체( $x_1, x_2$ )와 b로 구성된 트리플의 공간 관계에 대해 각도를 변환하면  $\{ \langle b, x_1, 405 \rangle, \langle b, x_2, 365 \rangle, \langle b, x_3, 330 \rangle \}$ 이 된다. 따라서 새로운 트리플의 공간 관계는  $(\frac{405 + 365 + 330}{3}) \bmod 360 = 6$ 이 된다.

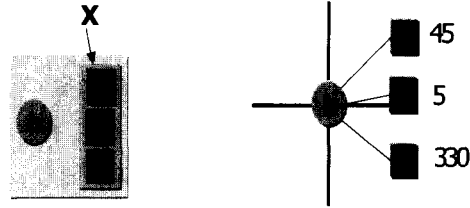


그림 5 새로운 복합 객체의 공간 관계

복합 객체와 다른 객체간의 평균 각도를 이용한 새로운 공간 관계는 기존의 LUB와 GLB를 이용한 방법[6]보다 개선된 공간 관계를 계산할 수 있다. 예를 들어, 그림 6과 같이 복합 객체(X)와 객체 b사이의 공간관계는 LUB와 GLB를 이용할 경우 새로운 공간 관계를 생성할 수 없으나 평균 각도를 이용하면  $15^\circ$ 가 된다.

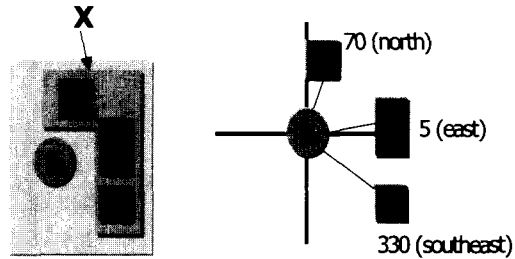


그림 6 복합 객체와 다른 객체간의 공간 관계

트리플 집합( $T_p$ )과 정의 13의 트리플 규칙을 이용하여 얻어진 복합 객체는 GenAvgDeg 함수를 이용하여 이미지내의 다른 객체간의 공간 관계를 계산하여 새로운 트리플을 생성한다.

트리플 집합과 트리플 규칙을 이용하여 복합 객체를 포함하는 트리플을 생성하는 함수 GenTriplePlus는 다음과 같다.

$$f_{GenTriplePlus} : T_p \rightarrow T_p^*, T_p^* \\
 = \{ \langle c, o, a \rangle \mid c, o \in O_p \wedge o \notin c.(recurse)comp, \\
 c = C_R(t_c), t_c \in T_C, a \text{는 } c \text{와 } o \text{의 공간관계,} \\
 a \in [0, 359] \}$$

```

Function GenTriplePlus( $T_P$ )
  Input :  $T_P$  (이미지  $p$ 의 트리플 집합)
  Output :  $T_P^*$ 
   $T_P^* = \{ \};$ 
   $O_P = \text{GetObject}(T_P);$ 
  while( $c = C_R(t_c)$ )
     $T_P^* = T_P^* \cup \text{GenConceptTriple}(c, O_P);$ 
     $O_P = O_P \cup \{c\};$ 
  end;
  return  $T_P^*$ ;
end;
    
```

```

Function GenConceptTriple( $c, O_P$ )
  Input :  $c, O_P$ 
  Output : tripleList

  tripleList = { };
  while( $o_i \in O_P \wedge o_i \notin c.(recurse)comp$ )
     $a = \text{GenAvgDeg}(o_i, c);$ 
    tripleList = tripleList  $\cup \{ \langle c, o, a \rangle \};$ 
  end;
  return tripleList;
end;
    
```

예 5. 예 4에서 GenTriplePlus(Tp)에 의해 생성되는 트리플 리스트는 다음과 같다.

TripleList = { <b, x<sub>1</sub>, 45>, <b, x<sub>2</sub>, 5>, <b, x<sub>3</sub>, 330>, <x<sub>1</sub>, x<sub>2</sub>, 90>, <x<sub>2</sub>, x<sub>3</sub>, 90>, <x<sub>1</sub>, x<sub>3</sub>, 90>, <b, X, 6> } 이 된다.

### 5. 특징 정보

내용 기반 이미지 검색 시스템은 색상(color), 질감(texture), 공간(spatial), 그리고 모형(shape)과 같은 특징 정보중 하나를 이용하거나 여러 개를 혼용한 형태이다. 본 논문에서는 공간과 색상 특징 정보만을 고려하였다. 공간 특징 정보는 퍼지 개념을 이용하여 처리하였으며, 색상 정보는 색상 특징 벡터의 차원을 최소화하기 위해 클러스터링을 수행 하였다.

#### 5.1 퍼지 기반 공간 관계

두 객체간의 공간 관계는 사람의 해석에 상당히 의존하는 애매한 개념이며, 사용자의 질의에 표현되는 공간관계 역시 매우 애매한 퍼지 용어라 할 수 있다. 따라서, 객체간의 공간관계를 나타내기 위해 K-weight 퍼지 멤버 함수를 사용하였다[18]. 퍼지 멤버 함수는 질의에 8방위나 각도로 표현된 공간 관계 정보에 대해 역화일의 객체 정보의 각도와의 퍼지 정합을 지원하기 위한 소속성 함수로 사용된다. 따라서 본 논문에서는 8방위에 대한 퍼지 멤버 함수와 각도에 대한 퍼지 멤버 함수를 정의하였다.

##### 5.1.1 방위에 대한 퍼지 멤버 함수

K-weight 함수를 이용한 각 방위에 대한 소속성 함수는 그림 7과 같다. K-weight 함수는 동쪽이 0° (즉,  $\mu_{east}(0^\circ) = 1$ ) 이고 시계 반대 방향으로 각도가 증가하는 식으로 정규화 된다. 예를 들어, 북동쪽은 45°에 북쪽

은 90°에 1이 되도록 정규화 되어 있다. 또한, 우리의 직관과 일치하려면 동쪽과 북동쪽의 중간인 22.5°를 기준으로 22.5°에 가까워질수록 동쪽에 대한 멤버 함수 값이 서서히 감소하여 22.5°일 때 0.5 값을 가지고 45°에 가까워짐에 따라 북동쪽에 대한 멤버 함수의 값이 증가하게 된다.

정의 17.  $b_0=0$  또는 360이고  $b_i=45*i, i=1, \dots, 7$ 이라 하면  $d_i \in D$ 를 위한 멤버 함수  $\mu_{d_i}, i=0, \dots, 7$ 는 다음과 같다.

$$\mu_{d_i}(x) = \begin{cases} C \times \{45^2 - (x - b_i)^2\}^k, & \text{이때 } i=1, \dots, 7, 45*(i-1) \leq x \leq 45*(i+1) \\ 0 & \text{그외} \end{cases}$$

$$\mu_{d_0}(x) = \begin{cases} C \times \{45^2 - (x - b_0)^2\}^k, & \text{이때 } b_0=0, 0 \leq x \leq 45 \\ C \times \{45^2 - (x - b_0)^2\}^k, & \text{이때 } b_0=360, 45*7 \leq x \leq 45*8 \\ 0 & \text{그외} \end{cases}$$

정규화를 위해서 상수 C값은  $1/2025^k$ 로 설정했으며, k값은 분석 결과 2.5를 선택하였다.

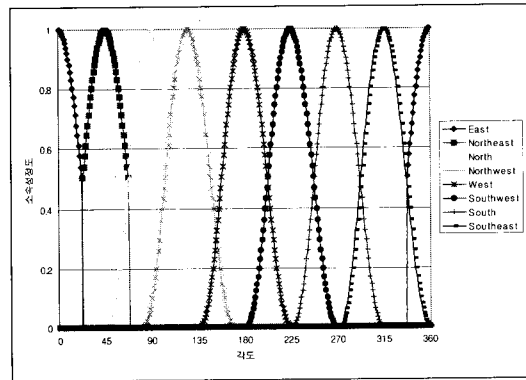


그림 7 8방위에 대한 멤버 함수

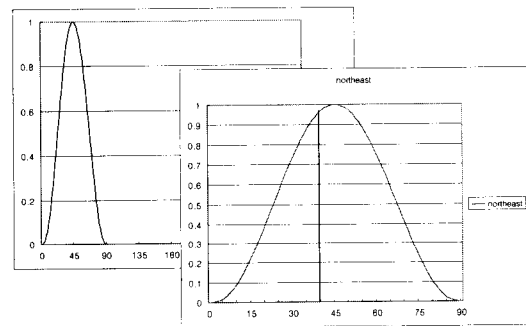


그림 8 북동쪽(northeast)에 대한 멤버 함수

예를 들어, 질의의 공간 관계가 북동쪽인 경우 퍼지 멤버 함수에 대한 그림은 그림 8과 같으며, 멤버 함수를



이용하여 이미지 p의 두 객체가 이루는 각도가 40°인 경우  $\mu_{northess}(40^\circ) = 0.97$  정도로 이미지 p는 질의의 공간 관계와 유사함을 나타낸다.

5.1.2 각도에 대한 퍼지 멤버 함수

K-weight 함수를 이용한 각도 공간 관계에 대한 소속성 함수는 그림 9과 같다. 그림 9는 30, 60, 350도에 대한 소속성 함수를 나타낸 것이다. 질의에 주어진 각도 (a)에 대해 두 객체가 이루는 각도의 소속성 정도를 나타내는 멤버 함수는 다음과 같다.

**정의 18.** 각도  $a \in [0, 360]$ 에 대한 멤버 함수  $\mu_a$ 는 다음과 같다.

$$\begin{aligned} b &= a - 360, \text{ 만일 } 315 < a \leq 360 \text{ 이고 } 0 \leq x < 45 \text{ 이면} \\ b &= a + 360, \text{ 만일 } 0 \leq a < 45 \text{ 이고 } 315 < x < 360 \text{ 이면} \\ b &= a, \text{ 그외} \end{aligned}$$

$$\mu_a(x) = C * (45^2 - (x - b)^2)^k$$

예를 들어, 질의의 공간 관계가 30°인 경우 퍼지 멤버 함수에 대한 그림은 그림 10과 같으며, 이미지 p의 두 객체가 이루는 각도가 40°인 경우 멤버 함수를 이용하여  $\mu_{30}(40^\circ) = 0.88$  정도로 이미지 p는 질의의 공간 관계와 유사함을 나타낸다.

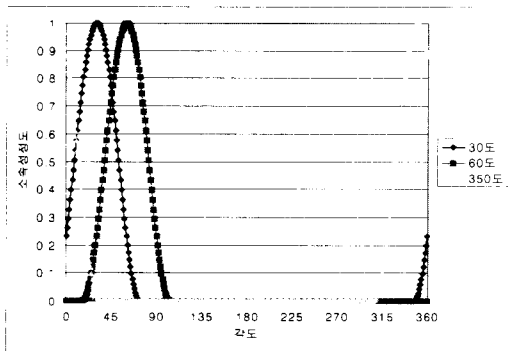


그림 9 각도에 대한 멤버 함수

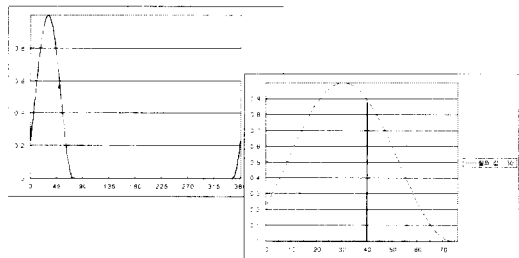


그림 10 질의 30도에 대한 멤버 함수

5.2 색상 특징 정보

색상 특징 정보는 이미지 검색 시스템에서 가장 대표적이면서 사람이 쉽게 분별할 수 있는 정보 중 하나이다. 본 논문에서는 이미지 색인기로부터 인식된 객체에 대해 색상 정보를 64개로 클러스터링하여 색상 특징 벡터를 추출하였으며, 효과적인 검색을 지원하기 위해서 CS-Tree 색인 기법을 이용하였다.

5.2.1 색상(Color)

일반 사용자들이 주로 사용하는 RGB 좌표계는 인간의 색인식 요소와 다르며 색차를 측정하는데 부적합하므로, 인간 시각의 등보성을 가지는 HSI 좌표계로 변환이 필요하다. RGB 좌표계의 데이터를 HSI 좌표계로 변환하는 공식은 식 (1)과 같다.

$$\begin{aligned} I &= \frac{1}{3}(R + G + B) \\ S &= 1 - \frac{3}{(R + G + B)} [\min(R, G, B)] \\ H &= \cos^{-1} \left\{ \frac{\frac{1}{2} [(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right\} \end{aligned} \quad (1)$$

식 (1)에서 H는 Hue 성분을 의미하며 색상에 해당되며, S는 Saturation으로 채도 성분, I는 Intensity로 명도 성분에 근사적으로 해당된다.

본 논문에서는 RGB 좌표계에서 HSI 좌표계로 변환하는데 있어, 데이터의 차원을 너무 세분하면 계산량과 메모리 요구량이 많아지기 때문에 Hue를 8 level(3 bit), Saturation은 2 level(1 bit), Intensity는 4 level(2-bit)로 하여 총 64 level(6 bit)로 양자화하였다. 양자화 레벨을 기준으로 클러스터를 생성하였으며, 이는 다음에 이미지의 픽셀이 어느 클러스터에 속하는지를 판별하는데 사용된다. 각 좌표값에 따라 양자화 level에 차이를 둔 이유는 인간의 시각이 색상을 인식하는데 Hue 성분, Intensity 성분, Saturation 성분 순으로 감도가 다르기 때문이다. 색상 특징 벡터를 구하는 과정은 다음과 같다.

```
float[] getColorFeature(object){
    short[][][] rgbColor =
    getRGBColorFromImage(object);
    float[][][] hsiColor = convertRGB2HSI(rgbColor);
    float[] frequencyVec = calculateDistanceFromCluster
        (hsiColor, cluster);
    float[] featureVector = makeFeatureVectorwith
        Normalization(frequencyVec);
    return featureVector;
}
```

먼저 해당 객체의 RGB 색상을 추출하여 HSI 색상으로 변환한다. HSI로 변환된 각 픽셀과 클러스터간의 거리를 계산하여 거리가 제일 짧은 클러스터의 특징 빈도수 벡터에 누적시킨다. 이렇게 누적된 값은 정규화 과정을 통해 최종적으로 특징 벡터가 생성된다. 픽셀(p)과 클러스터(c)의 거리 계산은 식 (2)과 같이 Euclidean 거리 계산 방식을 이용하였다.

$$dist(p, c) = \sqrt{(hue_p - hue_c)^2 + (sat_p - sat_c)^2 + (int_p - int_c)^2} \quad (2)$$

약 1000개의 다양한 이미지(128 X 85)에 대해서 실험을 실시하였다. 이 실험 결과는 본 논문의 목적과 크게 관련이 없기 때문에 여기서는 논하지 않고 다른 논문으로 넘기기로 한다. 아래 그림 11은 질의 이미지에 대해 단순 색상과 여러 색이 복합된 이미지의 상위 10개 검색 결과를 보인 것이다.

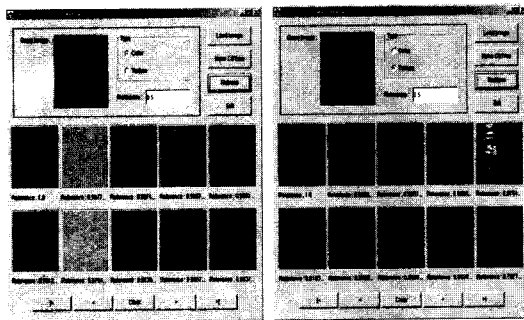


그림 11 색상 질의 결과

5.2.2 다차원 색인 구조

E-COIRS는 셀-기반 시그니처 트리(CS-트리)[19]를 사용하여 색상 특징 벡터를 색인하였다.

CS-트리에 저장된 객체의 색상 정보(o.color)와 질의 객체의 색상 정보(q.color)의 유사도를 구하는 방법은 식 (3)과 같다.

$$f_{simColor}(o, color, q, color) = \frac{\sum_{i=0}^n o.color_i \cdot q.color_i}{\sqrt{\sum_{i=0}^n (o.color_i)^2 \cdot \sum_{i=0}^n (q.color_i)^2}} \quad (3)$$

여기서 n은 색상 특징 벡터의 차원을 나타내는 것으로 본 논문에서는 64이다. 예를 들어, 질의 객체(q)의 색상 특징 벡터가 [0.1,0.5,0.0,...,0.0,0.9]이고, 객체(o)의 색상 특징 벡터가 [0.7,0.6,0.0,...,0.0,0.9]일 때, 두 객체간의 색상 유사도는 0.88이 된다.

6. 질의처리

질의는 트리플로 구성된 단위 질의나 단위 질의들의 논리합으로 구성된 합 단위소 질의 또는 합 단위소 질의들의 논리곱으로 구성된 곱 단위소 질의가 될 수 있다. 따라서 질의 처리의 기본은 단위 질의가 되며, 합 단위소 질의는 단위 질의의 max 값을 곱 단위소 질의는 min 값을 사용한다. 본 논문에서는 개념적으로 이해하기 쉽고 또 일반적인 퍼지 min/max 표준 연산자를 사용하였다.

다음은 질의를 정형화하기 위한 기본적인 정의이다.

**정의 19.** 단위질의 ( $Q_m$ )는 사용자 질의의 기본 단위로 트리플로 구성된다. 객체간의 공간관계는 방위 또는 각도로 표현된다.

$$Q_m = \langle o_i, o_j, d_{ij} \rangle, o_i, o_j \in O \text{ 이고 } d_{ij} \in D \cup [0, 359]$$

예를 들어, “붉은 색 소파(s) 북쪽에 TV(tv)가 있는 그림”을 검색하는 질의는  $Q_m = \langle s, tv, north \rangle$ , s.name = ‘소파’, tv.name = ‘TV’, s.color = [1.0, 0.0, ..., 0.0]이다. 또한, “작업대(w) 30° 방향에 오디오(au)가 있는 그림”을 검색하는 질의는  $Q_m = \langle w, au, 30 \rangle$ , w.name = ‘작업대’, au.name = ‘오디오’이다.

정의 19의 단위 질의는 질의에 포함된 IS-A객체와 각도로 주어진 공간 관계를 확장하여 확장 단위 질의  $Q_e$ 들을 생성한다. 따라서 단위 질의는 확장 단위 질의들의 논리 합 ( $Q_m = \bigvee_{i=1}^l Q_e$ )으로 구성된다.

**정의 20.** 확장 단위 질의 ( $Q_e$ )는 단위 질의를 확장한 것으로 트리플로 구성되며, 각 객체의 이름과 객체간의 공간 관계는 방위로 변환되어 표현된다.

$$Q_e = \langle o_i, name, o_j, name, d_{ij}^* \rangle, o_i, o_j \in O \text{ 이고 } d_{ij}^* \in D$$

질의 확장 과정에서 단위 질의의 IS-A 객체는 정의 11의 개념 확장함수( $f_{extSACon}$ )를 이용하여 하위 IS-A 객체로 확장하며, 단위 질의의 공간 관계가 각도인 경우 8방위로 확장한다. 8방위로 확장하는 함수는 다음과 같다.

**정의 21.**  $b_i = 45 * i, i = 0, \dots, 8$ 이고,  $d_0$ 과  $d_8 = east$ 이고,  $d_i \in D, i = 1, \dots, 7$ 의 순서는 반 시계 방향으로 방위를 나타내는 경우, 질의에 주어진 공간 관계  $a \in [0, 359]$ 를 8방위로 확장하는 함수  $f_{angToDeg}(a)$ 는 다음과 같다.

$$f_{angToDeg}(a) = \begin{cases} d_i, & a \text{ 와 } b_i \text{가 같은 경우} \\ d_i \vee d_{i+1}, & b_i < a < b_{i+1} \text{인 경우} \end{cases}$$

그림 12와 같이 두 객체 a, b는 객체 o의 66, 68도의 방향에 존재하므로 정의 3에 의해 각각 northeast와

north 방위로 변환되어 역화일에 저장된다. 만일 사용자의 질의에 공간 관계 값이 67도가 입력되면 northeast의 방위에 있는 역화일 값만 참조하므로 바로 근처에 있는 68도 값은 제외된다. 따라서 정의 21의 확장 함수를 사용함으로써 이를 해결할 수 있다.

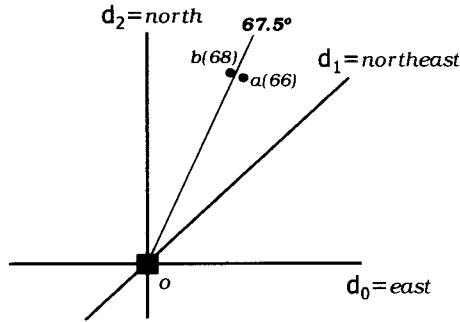


그림 12 각도 변환의 예

예 6. “소파 60° 방향에 TV가 있는 그림을 검색”하는 질의의 단위 질의는  $Q_m = \langle s, tv, 60 \rangle$ 이다. 그림 13에서 처럼 각도로 주어진 공간 관계 60은  $b_1 < a < b_2$ 이므로, 8방위는 northeast  $\vee$  north가 된다. 따라서 단위 질의를 확장한 확장 단위 질의  $Q_e = \langle \text{소파}, TV, \text{north} \rangle \vee \langle \text{소파}, TV, \text{northeast} \rangle$ 이다.

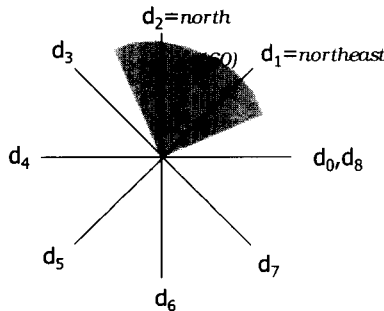


그림 13 각도 60의 8방위 변환

예 7. “의자(c) 60°에 라디오(r)가 있는 그림을 검색하라”는 질의 확장은 다음과 같이 처리된다.

단위 질의  $Q_m = \langle c, r, 60 \rangle$ 는 먼저 그림 3의 퍼지 시소러스를 이용하여 IS-A 객체인 ‘의자’에 대해서  $Q_e = \langle \text{‘소파’, ‘라디오’, } 60 \rangle \vee \langle \text{‘벤치’, ‘라디오’, } 60 \rangle$ 로 확장된다. 그리고, 공간 관계인 각도(60°)에 대해 8방위로 확장하여 최종적인 확장 단위 질의  $Q_e = \langle \text{‘소파’, ‘라디오’, north} \rangle$

$\vee \langle \text{‘벤치’, ‘라디오’, north} \rangle \vee \langle \text{‘소파’, ‘라디오’, northeast} \rangle \vee \langle \text{‘벤치’, ‘라디오’, northeast} \rangle$ 가 된다.

이렇게 확장된 확장 단위 질의는 역화일과 트리플 정합을 실행하여 객체 정보를 생성한다. 그리고, 트리플 정합에서 얻어진 객체 정보에 대해 개념, 공간, 그리고 색상 특징 정합을 처리한 후, 사용자가 정의한 가중치를 적용하여 이미지의 유사도를 계산한다.

유사도(similarity)는 역화일에서 조회된 객체 정보(s), 단위 질의( $Q_m$ ), 그리고 사용자가 설정한 가중치(wf)를 이용하여 계산하며, 가중치 계산에서 사용되는 개념 관련 정도( $r_{con}$ ), 공간 관련 정도( $r_{sp}$ ), 그리고 색상 관련 정도( $r_{col}$ )는 다음과 같이 계산된다.

정의 22.  $o_i, o_j, d_{ij} \in Q_m$ 이고  $o_i, o_j, a_{ij} \in s$ 일 때, 개념 관련 정도, 공간 관련 정도 그리고 색상 관련 정도는 다음과 같다.

$$r_{con} = \min(CR(o_i, name, o_j, name), CR(o_j, name, o_i, name))$$

$$r_{sp} = \mu_{d_{ij}}(a_{ij})$$

$$r_{col} = \min(f_{simColor}(o_i, color, o_j, color), f_{simColor}(o_j, color, o_i, color)).$$

정의 22에서 구해진 관련 정도 값과 사용자가 설정한 가중치를 이용하여 유사도를 계산하며, 식 (4)에 나타내었다. 그리고 사용자가 설정하는 가중치의 합은 1이 되도록 하였다.

$$f_{calcSimilarity}(s, Q_m, wf) = \frac{r_{con} * wf_{con} + r_{sp} * wf_{sp} + r_{col} * wf_{col}}{wf_{con} + wf_{sp} + wf_{col}} \quad (4)$$

여기서,  $wf_{con}, wf_{sp}, wf_{col}$ 는 각각 개념, 공간, 그리고 색상에 대한 가중치를 나타낸다.

앞서 설명한 확장 단위 질의를 처리하는 함수(Ext QueryProcess)는 다음과 같으며,  $\langle p, sim \rangle$  쌍으로 구성된 답 집합을 반환한다.

```
Function ExtQueryProcess(eQry, weight)
Input : eQry, weight
Output : RQ(Result set of ExtQuery)

RQ = {};
while(eQry != null)
  sSet = lookupInvertedFile(eQry.t.);
  while(sSet != null)
    sim = calcSimilarity(sSet.sj, unitQry, weight);
    RQ = RQ ∪ {<sSet.sj.p, sim>};
    j = j + 1;
  end.
  i = i + 1;
end.
return RQ;
end.
```

예 8. 예 7의 확장 질의에서  $\langle \text{‘소파’, ‘라디오’, north} \rangle$ 를 ExtQueryProcess에서는 다음과 같이 처리한

다.

먼저, 확장 질의  $Q_e = \langle \text{'소파'}, \text{'라디오'}, \text{north} \rangle$ 는 역화일 정합을 수행한다. 그림 2의 역화일로부터 정합을 통해 얻은 후보 답집합은  $\{ \langle 2, s, r, 85 \rangle, \langle 4, s, r, 100 \rangle \}$ 이 된다. 만일, 사용자가 질의 입력에서 공간 관계만을 고려했다면 공간 관계에 대한 가중치(weight)가 1.0이 되며, CalcSimilarity에서 공간 관계에 대한 유사도만을 계산한다. 따라서, 최종적으로 ExtQueryProcess에서 계산된 결과는  $\{ \langle 2, 0.97 \rangle, \langle 4, 0.88 \rangle \}$ 이다. 여기서 0.97는 공간 관계  $\mu_{north}(85)$ 를 나타낸다.

**정의 23.** 단위 질의 ( $Q_m$ )의 답집합은 다음과 같다.

$$\|Q_m\| = \left\{ \langle p, s \rangle \mid \langle p, s_k \rangle \in \bigcup_{i=1}^l \|Q_{e_i}\| \text{ and } s = \max(s_k) \text{ } k=1, \dots, l \right\}$$

**정의 24.** 합 단위소 질의 ( $Q_d$ )는 기본 단위 질의 ( $Q_m$ )들의 논리합으로 구성된다.

$$Q_d = \bigvee_{j=1}^m Q_m$$

**정의 25.** 합 단위소 질의 ( $Q_d$ )의 답집합은 다음과 같다.

$$\|Q_d\| = \left\{ \langle p, s \rangle \mid \langle p, s_k \rangle \in \bigcup_{j=1}^m \|Q_{m_j}\| \text{ and } s = \max(s_k) \text{ } k=1, \dots, m \right\}$$

**정의 26.** 곱 단위소 질의 ( $Q_c$ )는 합 단위소 질의 ( $Q_d$ )의 논리곱으로 구성된다.

$$Q_c = \bigwedge_{i=1}^n Q_d$$

**정의 27.** 곱 단위소 질의 ( $Q_c$ )의 답집합은 다음과 같다.

$$\|Q_c\| = \left\{ \langle p, s \rangle \mid \langle p, s_k \rangle \in \bigcap_{i=1}^n \|Q_{d_i}\| \text{ and } s = \min(s_k) \text{ } k=1, \dots, n \right\}$$

**예 9.** “식탁 북동쪽에 부엌세트가 있는 그림을 검색하라”는 질의는 다음과 같이 처리한다.

질의  $Q = Q_m = \langle t, k, \text{northeast} \rangle$ 는 단위 질의이므로 질의에 포함된 IS-A 객체나, 각도로 주어진 공간 관계에 대해서 확장한다. 그러나, 두가지 모두 해당 사항이 없기 때문에, 확장 단위 질의  $Q_e = \langle \text{식탁}, \text{부엌세트}, \text{northeast} \rangle$ 를 생성하며, 역화일의 정합을 통해 객체 정보  $s = \{ \langle p_1, t, k, 60 \rangle, \langle p_2, t, k, 40 \rangle, \langle p_3, t, k, 45 \rangle \}$ 를 얻는다. 객체 정보와 단위 질의의 객체 속성(개념, 공간, 그리고

색상)에 대한 관련 정도를 계산한 후 각각에 대해서 사용자의 가중치를 적용한 최종 답집합을 구한다. 만일 사용자가 공간 관계에 대한 가중치를 1.0로 설정했다면, 답집합은  $\{ \langle p_1, 0.75 \rangle, \langle p_2, 0.97 \rangle, \langle p_3, 1.0 \rangle \}$ 이다. 여기서 0.75는 공간 관계  $\mu_{northeast}(60)$ 를 나타낸다.

**예 10.** “식탁 북동쪽에 가전제품이 있거나 부엌세트가 있는 그림을 검색하라”는 질의는 다음과 같이 처리한다.

질의  $Q = Q_d = \langle t, h, \text{northeast} \rangle \vee \langle t, k, \text{northeast} \rangle$ 는 두 개의 단위 질의로 구성된 합 단위소 질의이다. 따라서 각각 단위 질의로 처리한 후 동일한 이미지에 대해서는 max 값을 취한다. 단위 질의  $Q_{m_1} = \langle t, h, \text{northeast} \rangle$ 는 질의에 IS-A 객체를 포함하고 있으므로 퍼지 시소러스를 이용하여 확장 단위 질의  $Q_{e_1} = \langle \text{식탁}, \text{에어콘}, \text{northeast} \rangle \vee \langle \text{식탁}, \text{TV}, \text{northeast} \rangle \vee \langle \text{식탁}, \text{비디오}, \text{northeast} \rangle \vee \langle \text{식탁}, \text{라디오}, \text{northeast} \rangle$ 를 생성한다. 또한,  $Q_{m_2} = \langle t, k, \text{northeast} \rangle$ 는 예 9에서처럼 확장 단위 질의  $Q_{e_2} = \langle \text{테이블}, \text{부엌세트}, \text{northeast} \rangle$ 를 생성한다. 이렇게 생성된 확장 단위 질의에 대해 역화일의 정합을 통한 객체 정보를 얻는다. 확장 단위 질의  $Q_{e_1}$ 는 4개의 트리플에 대해 객체 정보를 얻은 후 단위 질의와의 관련 정도를 계산한 후 동일 이미지의 경우 max 값을 갖는 답집합을 생성한다. 만일 4개의 객체 정보가 각각  $s_{11} = \{ \langle p_1, t, a, 45 \rangle, s_{12} = \{ \langle p_1, t, tv, 40 \rangle, s_{13} = \{ \langle p_2, t, v, 40 \rangle, s_{14} = \{ \langle p_3, t, r, 60 \rangle \}$ 와 같고, 사용자의 가중치 설정이 개념과 공간 관계를 0.5씩 설정한 경우에는 답집합은  $\{ \langle p_1, \max(0.7, 0.685) \rangle, \langle p_2, 0.7225 \rangle, \langle p_3, 0.6 \rangle \}$ 이며,  $Q_{e_2}$ 에 대한 답집합은  $\{ \langle p_1, 0.875 \rangle, \langle p_2, 0.985 \rangle, \langle p_3, 1.0 \rangle \}$ 이다. 따라서 질의의 최종적인 답집합은  $\{ \langle p_1, \max(0.7, 0.875) \rangle, \langle p_2, \max(0.7225, 0.985) \rangle, \langle p_3, (0.6, 1.0) \rangle \} = \{ \langle p_1, 0.875 \rangle, \langle p_2, 0.985 \rangle, \langle p_3, 1.0 \rangle \}$ 가 된다.

## 7. 시스템 구현 및 실험 결과

### 7.1 가시적 이미지 색인기

E-COIRS에서는 이미지내의 객체들을 명명하는 것과 객체간의 공간 관계를 쉽게 처리할 수 있도록 가시적 이미지 색인기를 사용하였다. 그림 14는 가시적 이미지 색인기의 화면이다. 여기서 도메인 전문가는 이미지 색인기를 사용하여 이미지내에 적절한 영역을 객체로써 명명할 수 있다. 명명된 객체들은 객체 간의 공간 관계와 색상과 같은 특정 정보를 자동으로 생성하며, 두 객체의 공간 관계를 포함한 트리플 정보는 트리플 뷰어를

통해 보여진다. 아래 그림은 5부분의 영역이 MBR를 사용하여 객체로 지정되었으며, 각 객체에 대한 특징 정보는 화면 오른쪽 상단에 있는 객체 특징 창을 통해 나타낸다.

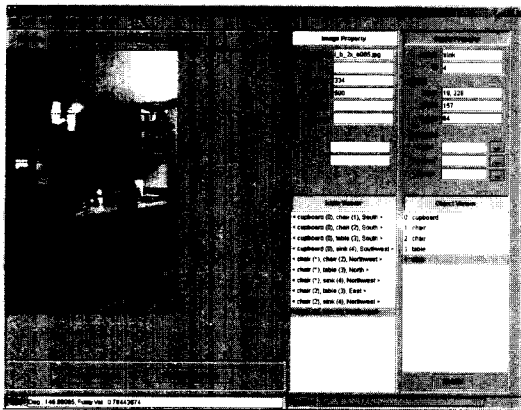


그림 14 가시적 이미지 색인기 화면의 예

가시적 이미지 색인기는 그림 15와 같이 6개의 모듈로 구성된다. 트리플 생성기는 입력으로 받아들인 이미지에 대해 도메인 전문가가 선택한 객체들을 트리플로 생성한다. 특징 추출기는 도메인 전문가에 의해 명명된 객체와 확장 트리플의 복합 객체에 대한 색상 특징 정보를 추출하여 JAVA Native Interface(JNI)을 이용하여 CS-Tree에 저장한다. 공간 관계 처리기는 도메인 전문가에 의해 명명된 객체간의 공간 관계와 확장 트리플의 공간 관계를 계산하며, 확장 트리플 생성기는 트리플 규칙을 이용하여 복합 객체를 처리한 후 확장된 트리플 집합을 생성한다. 확장된 트리플 집합과 각 객체에 대한 정보는 각각 역화일 관리자와 객체 정보 관리자에 의해 DB에 저장된다.

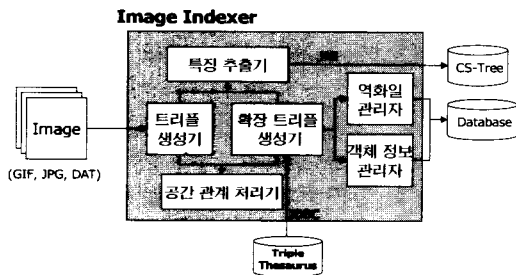


그림 15 가시적 이미지 색인기 구성도

### 7.2 질의 처리기

그림 16은 질의 처리기의 화면이다. 사용자는 화면 왼쪽 상단에 있는 시각 인터페이스를 통해서 질의를 생성한다. 질의는 단위 질의 항목에 대한 색상 정보를 표현할 수 있으며, 객체간의 공간 관계를 시각 인터페이스를 통해 질의할 수 있다. 또한, 사용자는 가중치 패널에 있는 6개의 항목에 대한 가중치를 설정할 수 있으며, 설정된 가중치에 따라 최종 검색된 이미지가 질의와의 유사도 값을 내림차 순으로 정렬하여 덤 네일 창(thumbnail windows)에 표시된다.

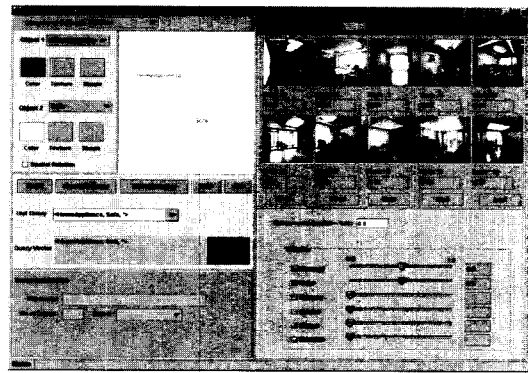


그림 16 질의 처리 화면의 예

질의 처리기는 그림 17과 같이 4개의 모듈로 구성된다. 질의 확장기는 시각 인터페이스를 통해 입력된 질의를 해석한 후, IS-A 개념은 퍼지 용어 시소러스를 사용하여 확장하며, 각도로 주어진 공간 관계는 정의 21의 확장함수를 사용하여 확장된 질의를 생성한다. 역화일 정합기는 확장된 질의에 대해 역화일과의 정합을 수행하여 객체 정보를 포함하는 후보 답집합을 구한다. 관련성 처리기는 후보 답집합과 질의에 포함된 개념, 공간 관계, 그리고 색상 특징 정보에 대한 관련 정도를 계산한다. 유사도 처리기는 사용자가 설정한 가중치를 이용

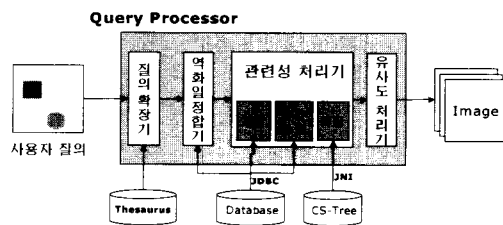


그림 17 질의 처리 시스템 구성도

하여 최종 유사도를 계산한 후 질의와 가장 유사도가 높은 것부터 내림차순으로 정렬된 답집합을 반환한다.

### 7.3 실험 결과

본 실험에서는 인테리어 이미지 500개를 대상으로 실험하였다. 먼저 이미지에 있는 객체들을 가시적 이미지 색인기를 이용하여 추출한 후 DB에 저장하였으며, 단순 객체(질의1, 6), 색상 속성(질의2), 각도 공간 관계(질의3), IS-A 객체(질의4), 복합 객체(질의5)를 포함하는 질의를 처리하였다. 실험에 사용된 질의는 다음과 같다.

질의 1: TV 동쪽에 소파가 있는 그림

질의 2: TV 동쪽에 검정색 소파가 있는 그림

질의 3: TV 30도 방향에 소파가 있는 그림

질의 4: 가전제품 동쪽에 소파가 있는 그림

질의 5: 응접세트 북쪽에 가전제품이 있는 그림

질의 6: TV 북쪽에 오디오나 서쪽에 의자가 있는 그림

질의 평가는 30명의 학생을 대상으로 실시하였다. 500개의 이미지에 대해 주어진 질의와의 관련도에 따라 질의와 아주 일치함(A), 조금 일치함(B), 일치하지 않음(C), 전혀 일치하지 않음(D)까지 4등급으로 구분토록 하였다. 그리고, A와 B 등급에 속한 결과 이미지 답집합과 E-COIRS의 답집합을 이용하여 각각 정확율과 재현율을 계산하였다.

표 1 실험 결과

|      | 정확율   | 재현율   |
|------|-------|-------|
| 질의 1 | 81.6% | 96.4% |
| 질의 2 | 86.5% | 72.6% |
| 질의 3 | 94.5% | 88.3% |
| 질의 4 | 85.2% | 89.2% |
| 질의 5 | 89.2% | 70.8% |
| 질의 6 | 82.4% | 93.3% |

개념과 공간관계에 관련된 질의(1, 3, 4, 6)에 비해 색상과 복합 객체에 대한 질의(2, 5)에 대해서는 낮은 재현율을 나타내고 있다. 색상과 관련된 재현율이 낮은 이유는 사용자 질의에 포함된 색상이 유사도 계산에서 근접에 있는 색상에 영향을 받기 때문이다. 만일 색상 특징 벡터를 64개보다 더 크게 하면 더 높은 재현율을 얻을 수 있으나, 차원이 커짐에 따라 검색 시간과 공간이 많이 소요되므로 응용 분야에 따라 특징 벡터의 조정이 요구된다. 그리고, 복합 객체에 대한 질의의 결과는 트리플 시소러스에 표현된 8방위 값에 따라 낮은 재현율을 나타내었다. 예를 들어, 사람들은 소파 근처에 탁자

가 있는 경우에 응접세트로 생각하지만, 본 논문에서는 8방위만을 고려함으로써 소파와 탁자가 멀리 떨어져 있는 경우에 대해서도 복합 객체로 인식되어 재현율에 영향을 나타내었다. 또한, 방위로 주어진 질의 보다는 각도로 주어진 질의의 정확율이 높게 나타나고 있다. 이는 사용자가 방위에 대해서는 각도보다 폭 넓은 값을 할당한 것으로 볼 수 있다. 예를 들어, 북쪽 방향인 경우 북서 또는 북동 방향에 대해서도 적합한 이미지로 선택하지만, 90도처럼 각도로 주어진 경우에 대해서는 90도 근처에 있는 이미지에 대해서만 답 집합으로 선택한다.

## 8. 결론 및 향후 연구

본 연구에서는 개념 기반 이미지 검색 시스템에 내용 기반 이미지 검색을 확장한 E-COIRS를 설계하고 구현하였다.

E-COIRS는 개념이나 이미지 특징만을 지원하는 다른 시스템과 비교해서 많은 장점을 가지고 있다. 첫째, E-COIRS는 개념과 관련 특징 정보들로 정제된 질의를 제공함으로써 재현율(recall)을 높게 하였다. 둘째, E-COIRS는 개념과 특징 정보의 속성을 포함하는 객체 기반 질의를 제공하도록 설계되었다. 따라서 E-COIRS는 질의 처리에서 다른 속성들을 쉽게 수용할 수 있다.

향후 연구로는 위상 공간 관계를 포함하는 3차원 이미지(VRML)를 지원하도록 확장하는 연구와 트리플 질의를 보다 쉽게 읽을 수 있도록 자연어 인터페이스에 대한 연구가 수행되어야 할 것이다. 또한, “등그런 모형의 가전 제품 북쪽에 부드러운 소파가 있는 그림을 검색”하는 질의와 같이 감성 용어와 애매한 모형 용어를 처리할 수 있도록 감성 색상 시소러스와 모형(shape) 시소러스의 개발이 요구된다.

## 참고 문헌

- [1] A.W.M. Smeulders, "Content-Based Image Retrieval at the End of the Early Years," IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, pp. 1349-1379, 2000.
- [2] G. Baxter and D. Anderson, "Image Indexing and Retrieval: Some Problems and Proposed Solutions," New Library World, Vol. 96, No. 1123, pp. 4-13, 1995.
- [3] A. Pentland, R. W. Picard, S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," International Journal of Computer Vision, 18(3), 1996.
- [4] M. Flickner, H. Sawhney, W. Niblack, and et al.,

"Query by Image and Video Content: The QBIC System," IEEE Computer, September, 1995.

[5] W. Y. Ma, "NETRA: A Toolbox for Navigating Large Image Databases," Ph.D. Dissertation, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, June 1997.

[6] J. D. Yang and H. J. Yang, "A Formal Framework for Image Indexing with Triples: Toward a Concept-Based Image Retrieval," International Journal of Intelligent Systems, Vol. 14, Issue 6, 1998.

[7] Wen-Syan Li, et al., "Hierarchical image modeling for object-based media retrieval," Data & Knowledge Engineering Vol. 27, pp. 138-176, 1998.

[8] S. K. Chang, Q. Y. Shi and C.W. Yan, "Iconic indexing by 2D string," IEEE Transaction, Pattern Analysis, pp. 413-428, 1987.

[9] C. C. Chang and S. Y. Lee, "Retrieval of Similar Pictures on Pictorial Databases," Pattern Recognition, pp. 675-680, 1991.

[10] D. Papadias and T. Sellis, "Spatial Reasoning Using Symbolic Arrays," Proceeding of Int. Conference GIS-From Space to Territory Theories and Method of Spatio-Temporal Reasoning in Geographic Space. Pisa, Italy, 1992.

[11] J. Wang, W. J. Yang and R. Acharya, "Color Space Quantization for Color-Content-Based Query Systems," Multimedia Tools and Applications, 13, pp. 73-91, 2001.

[12] Markus Stricker and Markus Orengo, "Similarity of Color Images," Proc. SPIE Storage and Retrieving for Image and Video Databases, 1995

[13] J. Huang, "Color-Spatial Image Indexing and Application," Ph.D. Dissertation, Dept. of Computer Science, Cornell University, August, 1998.

[14] White. D.A and Jain. R, "Similarity Indexing with the SS-tree," In Proc. 12th Intl. Conf. On Data Engineering, New Orleans, pp.516-523, 1996.

[15] Katayama N., Satoh S., "The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries," Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 369-380, 1997.

[16] K.I. Lin, H. Jagadish, C. Faloutsos, "The TV-tree: An Index Structure for High Dimensional Data," VLDB Journal, Vol. 3, pp. 517-542, 1994.

[17] Berchtold S., Keim D., Kriegel H.-P., "The X-tree: An Index Structure for High-Dimensional Data," 22nd Conf. on Very Large Databases, 1996, Bombay, India.

[18] J.D. Yang, "An Image Retrieval Model Based on Fuzzy Triples," Fuzzy Sets and Systems, Vol. 121, pp. 459-470, 2001.

[19] K. T. Song, J. W. Jang, "CS-Tree : Cell-based Signature Index Structure for Similarity Search in High-Dimensional Data," Proceedings on KISS, 127(1), pp. 134-136. 2000.



김 용 일

1984년 전남대학교 전산통계학과(이학사). 1986년 한국과학기술원 전산학과(공학석사). 1986년 ~ 1994년 한국원자력연구소 전자전산실(선임연구원). 1994년 ~ 2000년 초당대학교 컴퓨터학과(조교수). 2000년 ~ 현재 전북대학교 전산통계학과 박사과정. 2002년 ~ 현재 호남대학교 인터넷학부(전임강사). 관심분야는 멀티미디어 정보검색, 멀티미디어 DB



양 재 동

1983년 2월 서울대학교 컴퓨터공학과(학사). 1985년 2월 한국과학기술원 전산학과(석사). 1991년 2월 한국과학기술원 전산학과(박사). 1995년 1월 ~ 1996년 1월 Univ. of Florida, Visiting Scholar. 현재는 전북대학교 컴퓨터학과 교수, 영상·정보 신기술 연구소 연구원. 관심분야는 OODBs, Expert System, CASE



양 형 정

1991년 2월 전북대학교 전산통계학과(학사). 1993년 2월 전북대학교 전산통계학과(석사). 1998년 8월 전북대학교 전산통계학과(박사). 2000년 3월 ~ 현재 동신대학교 컴퓨터 응용학과군 전임강사. 관심분야는 OODBs, 이미지/ 비디오 정보

검색