

RBF 신경망을 이용한 내용 기반 영상 검색

(Content-Based Image Retrieval using RBF Neural Network)

이 형 구 [†] 유 석 인 ^{**}

(Hyoung K. Lee) (Suk I. Yoo)

요 약 내용 기반 영상 검색에서 대부분의 기존 방법들은 서로 다른 특징들 사이의 선형 관계를 가정하고 또 사용자가 직접 각 특징의 가중치를 설정하도록 한다. 허나 특징들 사이의 관계가 선형적으로 가정된 하에서는 고차원의 개념과 인간의 지각 주관성을 충분히 표현할 수 없는 단점이 있다. 본 논문에서는 신경망에 기반한 영상 검색 모델이 제안된다. 이는 RBFN을 이용한 내용 기반 영상 검색 기법과 인간 컴퓨터 상호작용의 접근 방법을 기반으로 구축되었다. RBFN을 이용하여 특징들 사이의 비선형적 관계를 추출해낼 수 있고 사용자가 처음에 질의 영상을 선택하고 관련성 피드백을 통하여 점차적으로 목표 영상을 찾아나가도록 함으로써 영상의 비교를 더 정확하게 할 수 있다. 실험은 145개의 클래스로 구분되며 1,015개의 영상을 포함하는 데이터베이스를 사용하여 재생과 정도를 계산하였다. 실험 결과는 제안된 방법의 재생과 정도가 각각 93.45%과 80.61%로서, 기존의 선형 결합 방법이나 순위 기반 방법 그리고 역전파 알고리즘에 기반한 방법보다 더 뛰어난 검색 성능을 지님을 보여준다.

키워드 : 내용기반 영상검색, RBFN, 선형 결합 방법, 순위 기반 방법, 관련성 피드백

Abstract In content-based image retrieval (CBIR), most conventional approaches assume a linear relationship between different features and require users themselves to assign the appropriate weights to each feature. However, the linear relationship assumed between the features is too restricted to accurately represent high-level concepts and the intricacies of human perception. In this paper, a neural network-based image retrieval (NNIR) model is proposed. It has been developed based on a human-computer interaction approach to CBIR using a radial basis function network (RBFN). By using the RBFN, this approach determines the nonlinear relationship between features and it allows the user to select an initial query image and search incrementally the target images via relevance feedback so that more accurate similarity comparison between images can be supported. The experiment was performed to calculate the level of recall and precision based on a database that contains 1,015 images and consists of 145 classes. The experimental results showed that the recall and precision level of the proposed approach were 93.45% and 80.61% respectively, which is superior than the existing approaches such as the linearly combining approach, the rank-based method, and the backpropagation algorithm-based method.

Key words : content-based image retrieval, radial basis function network, linearly combining method, rank-based method, relevance feedback

1. 서론

일반적으로, 영상을 검색할 때 처음의 검색 결과에서는 불완전한 결과를 얻는다. 그럴 경우에는 또 다른 질의를 사용하여 데이터베이스를 검색하는 것이 필수적이

다[1]. 영상 검색 시스템에 관한 이전의 연구는 “완전 자동화된 시스템”에 중점을 두고 “하나의 최적 특징”을 찾아내려고 했다[1, 2]. 그러한 접근 방법에서는 “최상의” 특징들과 표현 그리고 대응되는 가중치들이 고정되어서 고차원의 개념과 사용자의 지각 주관성을 효과적으로 반영시킬 수 없다. 더욱이 가중치를 사용자가 설정하도록 하는 것은 검색 시스템에 사용되는 저차원의 특징 표현들에 대한 심도있는 지식을 필요로 하므로 사용자에게는 적합한 방법이 아니다.

[†] 정 회 원 : 한국전자통신연구원 영상정보처리연구팀 연구원
hgl@etri.re.kr

^{**} 종신회원 : 서울대학교 컴퓨터공학부 교수
siyoo@ailab.snu.ac.kr

논문접수 : 2000년 4월 3일
심사완료 : 2001년 12월 24일

그러한 방법의 한계 때문에 내용 기반 영상 검색(content-based image retrieval: CBIR)에 관한 최근의 연구 방향은 사람을 검색 과정의 일부분으로 포함시키는 “상호작용 시스템” 또는 “검색과정 안의 사람”으로 옮겨졌다[1, 3, 4, 5]. 상호작용으로 영상 데이터베이스에 주석첨가[3], 검색전에 감독 학습(supervised learning)의 사용[4], 그리고 영상 검색 성능을 향상시키기 위해서 상호작용을 통한 중심어와 고차원 개념의 결합[5]을 그 예로 들 수 있다.

본 논문에서는 불완전한 검색 결과를 인간과 컴퓨터의 상호작용으로 보완하는 신경망 기반 영상 검색(neural network-based image retrieval: NNIR) 모델을 제안한다. 제안되는 방법은 훈련을 위해서 신경망 모델 중 radial basis function network (RBFN)[6]을 이용한다. RBFN을 이용한 이유는 영상 데이터베이스와 같이 전체 데이터베이스의 크기에 비해 훈련시킬 수 있는 자료의 크기가 작을 때는 역전파 알고리즘(backpropagation algorithm : BP)보다 RBFN이 더 적합하다[7]고 알려져 있기 때문이다. RBFN을 이용한 NNIR은 영상내에서 서로 다른 특징들 사이의 비선형적 관계를 추출한다. RBFN에 대한 입력은 각 영상 쌍의 차이 측정값들이고 출력은 다양한 입력 특징들에 기반한 영상의 유사성을 의미하는 0과 1 사이의 숫자이다.

RBFN의 훈련은 먼저 오프라인에서, 시각적으로 비슷한 영상들(positive examples)과 비슷하지 않은 영상들(negative examples)을 훈련시켜 적절한 가중치를 알아낸다. 그런데 사용자가 검색 결과에 만족하지 않는다면 사용자는 검색된 영상 각각에 ‘alike’, ‘similar’ 또는 ‘different’ 태그 중 하나를 지정한다. NNIR은 이들 태그와 태그가 설정된 영상들을 새로운 입력으로 하여 훈련 과정을 다시 반복하여 가중치를 정제한다. 이렇게 함으로써 NNIR은 영상 특징들을 결합시키는 새로운 방법을 제공한다. 각각의 특징들 대신에 특징들의 조합을 사용하는 것은 특히 어떤 한 개의 특징이 두드러지지 않는 일반적인 영상 데이터베이스에 대해서 효율적이다. 실험 결과에서 제안된 방법이 영상의 이질적 특징들을 사용하는 내용 기반 영상 검색을 효율적이고 유연하게 수행하는 것을 볼 수 있었다.

본 논문의 구성은 다음과 같다. 2 절에서는 내용 기반 영상 검색에서 이질적 특징을 결합시키는 것과 관련된 연구를 살펴보고, 3 절에서는 RBFN을 이용한 이질적 특징의 결합 방법을 설명한다. 4 절에서는 제안하는 내용 기반 영상 검색 모델의 구조와 구성 요소들에 대해서 설명한다. 제안된 방법의 실험 및 분석이 5 절에서

설명되고 6 절은 결론이다.

2. 관련 연구

영상의 내용은 색(color), 질감(texture), 모양(shape)이나 주석(text annotation)과 같은 서로 다른 종류의 특징들로 표현될 수 있다. 이들 각각의 특징 종류에 대해서 서로 다른 형식의 특징들을 추출할 수 있다. 예를 들면, 색에 대한 특징으로는 보통 히스토그램(histogram)이 사용되지만 모멘트(moment) 방법이나 색 집합과 같은 다른 방법들도 색에 대한 특징을 추출하기 위해서 사용될 수 있다. 질감에 대한 특징은 웨이블릿 기반 특징 추출 방법(wavelet-based feature extraction method)[8]에 의해서 생성될 수 있다. Mehtre 등[9]은 외곽선 기반 방법들(chain code, Fourier descriptor), 영역 기반 방법들(moment invariants, Zernike moments) 그리고 두 개를 혼합한 방법(moment invariants와 Fourier descriptor)과 같은 모양 특징을 추출하기 위한 여러 가지 방법들을 소개하였다.

두 영상 사이의 유사성은 보통 특징 공간에서 두 영상의 특징 벡터들 사이의 거리로 측정된다. 색, 질감 그리고 모양 특징들은 서로 다른 계산 방법으로 생성되므로, 이들을 비교하기 위해서는 각각의 유사성 측정 방법을 따라야 한다. 이런 이유 때문에 내용 기반 영상 검색 과정이 보통은 각각의 특징에 대해서 수행된다. 영상을 구분할 때 사람이 인식하는 경우를 생각해보면 이들 특징들이 같은 정도의 중요성을 갖지는 않는다. 따라서, 시각적 검색 시스템은 질의 영상에 대한 데이터베이스 영상의 전체적인 유사성을 결정하기 위해서 각 특징들의 중요도를 고려해야 한다.

주어진 질의 영상에 대한 일련의 유사 영상들은 각각의 특징에 기반하여 선택될 수 있다. 그러나, 질의에 대한 최종의 유사 영상들은 이들 영상 각각의 특징들을 조합하는 것에 의해서 선택되어야 한다. 현재 Photobook[10], QBIC[11], Virage[12] 그리고 NETRA[13]와 같은 대부분의 내용 기반 영상 검색 시스템들은 서로 다른 특징들의 유사성 측정치를 조합하기 위하여 가중치가 부여된 선형 방법(weighted linear method)을 사용한다. 즉, 특징 클래스 f_1, \dots, f_r 에 대해서 질의 영상과 데이터베이스의 비교되는 영상에 대한 유사성 측정값이, 대응되는 가중치 w_1, \dots, w_r 과 함께 m_1, \dots, m_r 로 주어지면, 전체 유사성은 $\sum_{i=1}^r w_i m_i$ 로 계산된다. 여기서 r 은 고려되는 특징 수이다. 그러나 다

양한 특징들이 반드시 선형적으로 연관된 것은 아니다. 예를 들면, 색과 질감의 유사성 측정값들은 일반적으로 선형적인 교환관계를 나타내지 않는다[14]. 즉, 색과 질감의 개별적 검색 결과와 선형 조합을 만들었을 때의 검색 결과는 매우 다르게 나타난다. 따라서 이질적 특징들을 조합하기 위해서는 그들의 조합을 더 쉽게 하기 위해 각각의 유사성 측정치에 비선형적 변환을 적용시키는 것이 중요하다. 그러한 변환중의 한가지는 연관성 점수에 따라서 검색된 영상에 부여하는 순위(rank)이다. 범위, 평균 그리고 분포가 서로 다른 유사성 점수를 직접 조합하기보다는 검색된 영상들의 유사성을 정할 수 있는 새로운 척도로서 순위들을 결합시키는 것이다[15]. 보통은 순위들의 최소값, 평균 또는 최대값이 사용된다. 이 방법은 문서 검색에 대해서는 좋은 결과를 보여주었고 서로 다른 종류의 질감 특징들을 조합하기 위해서도 사용되었다[16]. 그러나 Sheikholeslami 등[14]은 단지 순위에 기반하여 조합하는 것은 영상들의 올바른 전체 순위를 이끌어내지 않음을 지적하였다. 그리고, 영상 검색 분야에서의 신경망 이용은 주로 특징들의 색인 구성에 관한 색인 기법(indexing technique)에 제한돼 왔다[17].

또 다른 중요한 고려 사항은 특징들의 가중치이다. 특징들의 선형 조합을 이용하는 기존의 대부분의 시스템에서는 사용자가 직접 가중치를 설정해야 한다[18]. 그러나 사용자들은 보통 이런 종류의 질의 방법으로 영상을 정렬하지 않는다. 특히, 특징의 수가 증가하면 특징들 사이의 상대적인 가중치를 어떻게 정해야 할 지에 관한 직관을 잃어버리게 된다. 또한, 유사성의 측정값들은 보통 0과 1사이의 범위에 있기 때문에 보통은 거리 측정값을 정규화하여 유사성 값으로 변환시킨다. 그러나 각 특징 값의 정규화 과정은 특징 공간과 특징 벡터 분포가 서로 다른 점을 고려하여 정규화해야 하는데 이를 위한 과정은 복잡하다[3].

본 논문에서는 영상의 이질적 특징들을 RBFN을 이용하여 비선형적으로 조합하는 모델을 제안한다. 제안되는 방법은 일련의 훈련 자료를 이용하여 특징들에 가중치를 부여함으로써 사용자가 직접 가중치를 설정해야 하는 부담을 덜어준다.

3. Radial Basis Function Network을 이용한 이질적 특징들의 결합

Radial Basis Function Network[6]을 이용하여 이질적 특징의 비교값들을 결합시키는 방법을 살펴본다.

영상 I_1 과 I_2 의 특징 벡터들간의 r -차원 차이벡터를 $\vec{v} = (v_1, v_2, \dots, v_r)$ 라고 했을 때 RBFN에 대한 입력은 \vec{v} 이다. 영상 I_1 과 I_2 의 모든 특징들이 비슷하면 RBFN의 결과값은 1에 가깝게 되고 그렇지 않으면 0에 가깝게 된다. RBFN은 입력 \vec{v} 와 출력 y_i 를 갖는 일련의 함수들 $\{f_i(\vec{v}) | y_i = f_i(\vec{v}), i=1, 2, \dots, m_1\}$ 을 구현한다. 여기서 m_1 은 구현되는 함수의 수이다. 모든 함수를 RBFN으로 근사시킬 수 있다는 것은 증명되어 있다[19]. 따라서 RBFN 모델은 특징 클래스간의 선형적인 관계뿐만 아니라 비선형적인 관계도 표현할 수 있다.

3.1 Radial Basis Function Network

M 개의 r -차원 입력 벡터와 스칼라(scalar) 출력을 각각 $\{\vec{v}^i | \vec{v}^i = (v_1, v_2, \dots, v_r), i=1, \dots, M\}$ 와 $y_i, i=1, \dots, M$, 라 하고 입력 출력 관계를 다음과 같이 표현하자.

$$y_i = f(\vec{v}^i) + \epsilon \quad (3.1)$$

여기서 f 는 알려지지 않은 함수이고 ϵ 은 평균적으로 0인 무작위 잡음(random noise)이다. M 개의 입력 출력 쌍 $\{(\vec{v}^i, y_i)\}_{i=1}^M$ 가 주어졌을 때, RBFN은 식 (3.2)를 최소화시키는 함수 f 를 생성한다.

$$H(f) = \sum_{i=1}^m (y_i - f(\vec{v}^i))^2 + \lambda \|Df\|^2 \quad (3.2)$$

여기서 m_1 은 RBFN의 중심값의 개수로서 보통은 입력 출력 쌍의 수 M 보다 작거나 같다. D 는 원형적 대칭성(radial symmetry)을 갖는 미분 연산자이며 $\|\cdot\|$ 는 유클리드 노름(euclidean norm)이고, λ 는 조정(regularization) 계수로서 양의 상수이다. 첫번째 항은 주어진 출력 패턴(y_i)과 계산되는 f 사이의 거리를 측정하고, 두번째 항은 평활화(smoothness)의 정도를 조정한다. 조정 계수 λ 는 이 두 항의 비중을 결정한다.

$\|Df\|^2$ 을 적절하게 선택했을 때, 식 (3.2)의 해는 다음과 같은 형태로 주어진다.

$$f(\vec{v}) = \sum_{j=1}^m w_j g(\|\vec{v} - \vec{c}^j\|) \quad (3.3)$$

여기서 w_j 와 \vec{c}^j ($\vec{c}^j \in \{\vec{v}^1, \vec{v}^2, \dots, \vec{v}^m\}, j=1, 2, \dots, m_1, m_1 \leq M$)를 각각 RBF 계수와 RBF 중심이라 부르고, 실제로 이들은 학습 패턴으로 RBFN을 훈련시키는 과정에서 결정된다. 본 논문에서 RBF 중심의 학습은 OLS (orthogonal least squares) 알고리즘[20]을 사용하였다. OLS 알고리즘에서는 매 epoch마다 하나의 중심이 추가되는데, 중심값의 선택은 현재까지 선택된 중심값들과의 거리차가 가장 큰 값으로 한다. g 는 식 (3.4)의 형태를 갖는 가우스 함수이다.

$$g(\|\bar{v}-\bar{c}\|) = \exp\left(-\frac{\|\bar{v}-\bar{c}\|^2}{2\sigma^2}\right) \quad (3.4)$$

여기서 인자 σ 는 가우스 함수의 표준 편차이고 실제 구현에서는 $\sigma=1$ 로 설정하였다. 함수 g 는 입력 \bar{v} 와 중심들 \bar{c} 중 하나와의 r 차원 유클리드 거리이므로 "Radial Basis Function"이라 부른다. 함수 f 는 이들 RBF의 선형 결합으로 표현되므로 "RBF 네트워크"라고 부른다. RBF 계수, w_j 는 식 (3.5)의 해로 결정된다.

$$(G+\lambda I)\bar{w} = \bar{y} \quad (3.5)$$

여기서

$$\begin{aligned} \bar{y} &= (y_1, y_2, \dots, y_{m_1})^T \quad (m_1 \times 1) \\ \bar{w} &= (w_1, w_2, \dots, w_{m_1})^T \quad (m_1 \times 1) \\ (G)_{ps} &= g(\|\bar{v}^p - \bar{v}^s\|) \quad (1 \leq p, s \leq m_1) \end{aligned}$$

p 와 s 는 행렬 G 의 행과 열을 가리키고, 행렬 $(G+\lambda I)$ 는 $\lambda \geq 0$ 일 때 항상 역행렬을 갖는다. 계수 λ 는 과적합(overfitting)이나 지나친 평활화(oversmoothing)가 일어나지 않도록 경험적으로 조절된다. λ 를 키우면 더 단순화된 근사가 이루어지지만, 학습 패턴과 네트워크 출력의 차이는 더 커지게 된다. λ 는 학습 패턴에 포함된 잡음의 양에 비례해야 한다고 알려져 있다.

그림 1은 사용된 RBFN의 구조를 보여준다. RBFN의 입력층은 입력 패턴 \bar{v} 의 원소 수만큼의 뉴런으로 구성되며, 은닉층의 뉴런 각각은 하나의 중심을 표현한다. 매 epoch마다 하나의 중심이 추가되는데, 이 과정은 '자승합 오차(sum squared error : SSE)가 주어진 기준 이하로 떨어지거나 은닉 노드의 수가 제한점을 넘는 순간까지 반복된다.

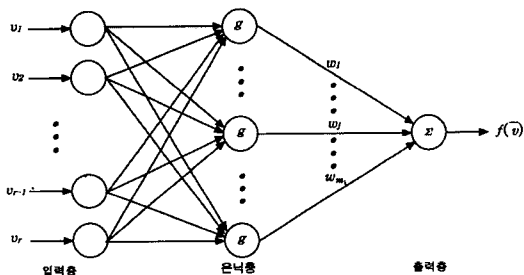


그림 1 RBFN의 구조

RBFN의 훈련은 먼저 오프라인에서, 시각적으로 비슷한 영상들¹⁾과 비슷하지 않은 영상들을 훈련시켜 적절한

1) 본 논문에서 시각적으로 비슷한 영상들은 표 2의 변환방법으로 변환된 같은 클래스의 영상들을 의미한다.

가중치를 알아낸다. 이렇게 RBFN이 훈련되면 네트워크의 특징 클래스들은 적절한 가중치를 갖게되어 실제 질의 영상이 주어졌을 때 이질적 특징들을 결합할 수 있게된다. 그런데 사용자가 검색 결과에 만족하지 않는다면 사용자는 검색된 영상 각각에 'alike', 'similar' 또는 'different' 태그 중 하나를 지정한다. NNIR은 이들 태그와 태그가 설정된 영상들을 새로운 입력으로 하여 훈련 과정을 다시 반복하여 가중치를 정제한다.

이러한 RBFN을 사용함으로써, NNIR은 선형 결합을 사용하는 방법들과는 달리 각 특징 클래스의 유사성이나 거리를 다른 특징 클래스들과 독립적으로 측정할 수 있고, RBFN의 입력으로 직접 사용할 수 있다. 이런 면에서, NNIR 모델은 이전의 선형 결합 방법들보다 더 유연하다고 할 수 있다.

4. NNIR 영상 검색 모델의 구현

본 연구에서 제안한 NNIR 모델의 구성을 보면 그림 2와 같다. 질의 영상이 주어지면 먼저 서로 다른 특징 요소별로 질의 영상의 특징이 추출된다. 데이터베이스 영상들에 대해서도 같은 형식의 특징들을 추출하여 미리 저장한다. 다음 단계로, NNIR은 질의 영상의 특징과 데이터베이스 영상의 특징을 각각의 특징별로 비교하여 일련의 비교값 벡터를 생성한다. 다음 단계로는 특징별로 얻어진 비교값들을 각 특징의 중요도에 의거하여 결합하여서 검색된 영상의 집합을 만든다. 만약 사용자가 검색결과에 만족하지 않는다면 사용자는 검색된 영상 각각에 'alike', 'similar' 또는 'different' 태그 중 하나를 지정한다. NNIR은 이들 태그와 태그가 설정된 영상을 이용하여 RBFN을 추가적으로 훈련시키고 검색을 다시 수행한다.

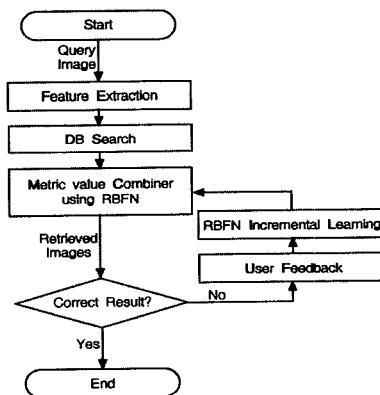


그림 2 NNIR 영상 검색 모델의 구현도

그림 2에서 각각의 구성 요소는 다음과 같은 동작을 수행한다.

A. Feature Extraction(FE): 주어진 질의 영상 $q \in Q$ (Q 는 영상 데이터베이스)에 대해서 그것의 이질적 특징들 F^q 를 추출한다. 식 (4.1)의 F^q 는 색, 질감 그리고 모양 등의 영상과 연관된 특징들의 집합이다.

$$F^q = \{\overline{f_1}, \overline{f_2}, \dots, \overline{f_n}\} \quad (4.1)$$

질의 영상 $q = (R_q, G_q, B_q)$ 는 $M \times N \times 3$ 크기의 자료 행렬로 저장되는 RGB 영상으로서 R_q, G_q 그리고 B_q 는 각각 빨간색, 녹색, 파란색 공간을 나타내는 $M \times N$ 크기의 자료 행렬이다.

식 (4.1)의 F^q 중에서 $\overline{f_1}, \overline{f_2}, \overline{f_3}$ 은 각각 영상 q 의 빨간색, 녹색 그리고 파란색 히스토그램이고 이때 각각은 식 (4.2)로 계산된다.

$$f_i^q[k] = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \begin{cases} 1 & \text{if } q[i][x, y] = k \\ 0 & \text{otherwise} \end{cases}, \quad (4.2)$$

for $0 \leq k \leq L-1, i = 1, 2, 3$

여기서 L 은 영상의 색 수준 수이다. f_4, f_5, f_6 은 각각 영상 q 에 대한 빨간색, 녹색 그리고 파란색 히스토그램의 일차 모멘트(평균)이고, f_7, f_8, f_9 는 이차 모멘트(분산)이다. 이들의 계산식은 각각 (4.3)과 (4.4)와 같다.

$$f_i^q = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{q[i-3][x, y]}{MN}, \quad \text{for } i = 4, 5, 6 \quad (4.3)$$

$$f_j^q = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{(q[j-6][x, y] - f_j^q)^2}{MN}, \quad \text{for } j = 7, 8, 9 \quad (4.4)$$

질감에 대한 특징을 얻기위해 영상 q 에 웨이블릿 변환[8]을 표 1에 나와있는 방법대로 3단계까지 적용하여 상수 행렬 a_k, b_k, c_k , 그리고 d_k ($k = 1, 2, 3$)를 얻는다.

(f_{10}, \dots, f_{19}) 는 이들 웨이블릿 상수 행렬의 각각의 평균이고 (f_{20}, \dots, f_{29}) 는 분산이다. 표 1의 \hat{H} 와 \hat{G} 필터는 각각 식 (4.5)와 (4.6)으로 구해지며, 이 두 식에서 $(a(1), a(2), \dots, a(2n-1), a(2n))$ 은 표 1의 각 행에서 선행 필터에 대해서는 $a_{k-1}(i, j)$ 의 행을 그리고 두 번째의 필터에 대해서는 선행 필터 적용 결과로 얻어진 행

표 1 a_k, b_k, c_k, d_k 의 계산

	변환 순서
$a_k(i, j)$	Rows of $a_{k-1}(i, j)$ with \hat{H} followed by columns of result with \hat{H}
$b_k(i, j)$	Rows of $a_{k-1}(i, j)$ with \hat{H} followed by columns of result with \hat{G}
$c_k(i, j)$	Rows of $a_{k-1}(i, j)$ with \hat{G} followed by columns of result with \hat{H}
$d_k(i, j)$	Rows of $a_{k-1}(i, j)$ with \hat{G} followed by columns of result with \hat{G}

렬의 각 열을 가리키며 $2n$ 은 이때 각 행과 열의 크기이다.

$$a_0 = 1/3(R_q + G_q + B_q)$$

$$\hat{H}(n) = \frac{1}{2} [a(2n) + a(2n-1)] \quad (4.5)$$

$$\hat{G}(n) = \frac{1}{2} [a(2n) - a(2n-1)] \quad (4.6)$$

B. DB Search(DS): 각각의 질의 특징 벡터와 데이터베이스 영상에 대응되는 특징 벡터를 비교하여 비교값 벡터를 만들어낸다.

특징 벡터 $(\overline{f_1}, \overline{f_2}, \dots, \overline{f_n})$ 를 갖는 영상을 질의 영상 q 라 하고, 특징 벡터 $(\overline{f_1}, \overline{f_2}, \dots, \overline{f_n})$ 를 갖는 영상을 데이터베이스에 저장된 비교 영상 i 라 하자. 이 때 비교값 벡터 (v_1, v_2, \dots, v_r) 은 히스토그램 특징에 대해서는 히스토그램 중첩 측정방법에 따른 거리를 그리고 나머지 특징에 대해서는 차이값을 계산해서 다음과 같이 구해진다.

히스토그램 특징들 $(\overline{f_1}, \overline{f_2}, \overline{f_3})$ 를 비교하기 위하여 히스토그램 중첩(Histogram intersection) 측정방법이 사용된다. 히스토그램 f_i 와 f_j 의 중첩에 따른 거리는 다음과 같이 계산된다:

$$v_i = 1 - \frac{\sum_{m=0}^{K-1} \min(f_i^q[m], f_i^i[m])}{\min(|f_i^q|, |f_i^i|)} \quad \text{for } i = 1, 2, 3 \quad (4.7)$$

여기서 $|f_i^q| = \sum_{m=0}^{K-1} f_i^q[m]$, $|f_i^i| = \sum_{m=0}^{K-1} f_i^i[m]$ 이고 K 는 히스토그램의 구간수이며 실제 중첩도는 등호 우변의 두 제 항이다. 거리 v_i 는 0과 1사이의 값으로서 두 개의 히스토그램이 비슷할수록 식 (4.7)에서 등호 우변의 두 제 항의 분자가 커지게 되어 0에 가깝게 된다. 나머지 특징들 (f_4, \dots, f_9) 를 비교하기 위해서는 다음 식을 사용한다:

$$v_i = f_i^q - f_i^i, \quad i = 4, 5, \dots, r \quad (4.8)$$

C. Metric values Combiner using RBFN (MC): 3 절에서 설명된 미리 훈련된 RBFN을 사용하여 앞 단계, DS에서 계산한 비교값 (v_1, v_2, \dots, v_r) 을 조합시킨 후 일련의 검색된 영상을 출력한다.

D. User Feedback(UF): 질의 영상 q 에 대해서 MC 단계가 출력한 영상들에 대하여 사용자의 유사성 판단을 얻는다. 만약 사용자가 결과에 만족하지 않는다면, 사용자는 검색된 영상들의 각각에 자신의 정보 필요 정도와 지각의 주관성²⁾에 따라 'alike', 'similar', 또는

2) 본 논문에서는 지각의 주관성을 자기 자신과 같은 영상을 'alike'로, 표2의 변환 방법으로 생성된 영상을 'similar'로, 그리고 그 외의 영상들을 'different'로 설정한다.

'different'의 태그를 붙인다. 식 (4.9)는 각각의 태그에 대해서 NNIR에 실제로 반영되는 유사도를 가리킨다.

$$UF(q) = \begin{cases} 1, & \text{if } alike \\ 0.9, & \text{if } similar \\ 0.001, & \text{if } different \end{cases} \quad (4.9)$$

E. RBFN Incremental Learning(IL): UF에서 얻은 사용자의 피드백(feedback)과 사용자가 UF 단계에서 태그를 붙인 영상들을 대상으로 RBFN을 추가적으로 훈련시킨다.

그림 3은 비교값 벡터들, RBFN, UF 그리고 IL의 상호관계를 나타내는 것으로 비교값 벡터를 RBFN으로 결합시킨 결과가 만족스럽지 않으면 UF를 받아들여 RBFN을 추가적으로 훈련시키는 관계를 보여준다.

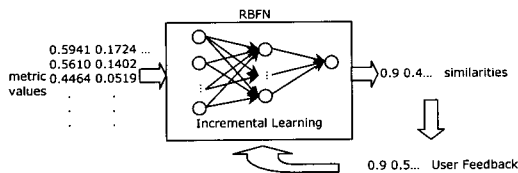


그림 3 차이값 벡터들, RBFN, UF와 IL의 상호관계

위의 검색 과정을 요약하면 다음과 같다.

단계 1: 주어진 질의 영상 $q \in Q$ 에 대해서, 그것의 이질적 특징들 $(\overline{f_1}, \overline{f_2}, \dots, \overline{f_n})$ 을 추출한다.

단계 2: 각각의 특징들 $(\overline{f_1}, \overline{f_2}, \dots, \overline{f_n})$ 와 데이터베이스 영상의 대응되는 특징들 $(\overline{f_1}, \overline{f_2}, \dots, \overline{f_n})$ 을 비교하여 비교값 벡터 (v_1, v_2, \dots, v_n) 을 생성한다.

단계 3: RBFN을 사용하여 비교값 벡터 (v_1, v_2, \dots, v_n) 를 조합하고 결과에 따라서 일련의 검색된 영상을 출력한다. 사용자가 만족하면 종료하고 그렇지 않으면 다음 단계를 수행한다.

단계 4: 사용자의 유사성 판단을 입력받는다.

단계 5: 단계 4의 유사성에 따라 RBFN을 추가적으로 훈련시킨다.

단계 6: 훈련된 RBFN을 가지고 단계 3으로 간다.

단계 3~6은 이질적 특징과 사용자의 유사성 판단에 기반하여 검색하는 과정의 필수적인 부분이다.

5. 실험 및 분석

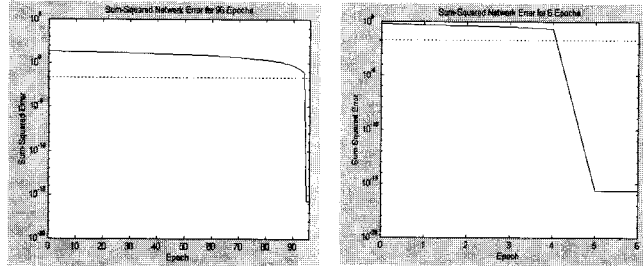
NNIR의 검색 성능을 측정하기 위하여 프로토타입(prototype) 영상 검색 모델이 구현되었다. 또한 검색 성능을 전형적인 선형 결합 방법[10-13], 순위 기반 방법[15, 16], 그리고 역전파 알고리즘을 이용한 방법[14]과 비교하기 위하여 이 방법들도 구현하였다.

구현된 선형 결합 방법은 특징 추출 단계, 유사성 측정 단계, 그리고 유사성 결합 단계로 구성된다. 특징 추출 단계와 유사성 측정 단계로는 4 절에 설명되어 있는 NNIR과 같은 단계를 이용한다. 유사성 결합 단계에서는, 특징 벡터가 (f_1, \dots, f_n) 로, 가중치가 (w_1, \dots, w_n) 로 그리고 질의 영상에 대한 데이터베이스 영상과의 차이 값이 (v_1, v_2, \dots, v_n) 로 주어지면 전체적인 유사성은 $\sum w_i v_i$ 로 계산된다. 가중치는 몇 번의 시행착오를 거쳐 가장 좋은 결과를 나타내는 값으로 설정한다. 순위 기반 방법도 특징 추출과 특징별 차이 측정을 위해서는 NNIR과 같은 단계를 사용하고 유사성 결합 단계가 다르다. 순위 기반 방법은 각 특징의 차이값을 직접 결합시키지 않고 차이값의 순위 중 최대, 최소 또는 평균을 선택한다[15]. 역전파 알고리즘을 이용한 방법은 NNIR에서 RBFN 대신에 역전파 알고리즘을 이용하고 사용자의 피드백을 이용하지 않는다[14]. 구현된 역전파 알고리즘은 20 개의 노드로 구성된 한 단계의 은닉 층(hidden layer)을 갖고 29 개의 입력 뉴런으로부터 각 영상 쌍의 차이 벡터를 입력받으며 하나의 출력으로 유사성의 정도를 나타낸다. 역전파 알고리즘의 훈련은 출력(유사성의 값: 자기 자신은 1, 같은 클래스의 다른 영상은 0.9, 다른 클래스의 영상은 0.001)을 알고있는 100 쌍의 영상으로, 학습률(learning rate)은 0.015로 그리고 자승합오차 목표(sum squared error goal) 인자는 0.005로 설정하고 훈련시켰다.

실험에서는 클래스가 구분되어 있고 그 정보가 알려진 하나의 영상 데이터베이스를 사용하였다. 사용된 영상 데이터베이스는 다음과 같이 생성되었다. 먼저 'XOOM, Inc'에서 제공되는 'Web Clip Empire 250,000'에서 145 개의 영상을 임의로 선택하였다. 선택된 각 영상에 변환 프로그램을 적용하여 같은 클래스에 속하는 6 개의 다른 영상을 생성한다. 그러면 한 클래스는 원래 영상과 변환된 6 개의 영상을 포함하여 7 개의 영상으로 구성된다. 적용된 변환 방법이 표 2에 요약되어 있다.

표 2 원본 영상을 변환시키는 6 가지 방법들

	변환방법		변환방법
1	가우스 잡음 첨가 (평균=0, 표준편차=0.01)	4	90° 회전
2	Salt-and-pepper 잡음 첨가 (잡음밀도=0.02)	5	180° 회전
3	0.6배로 축소	6	270° 회전

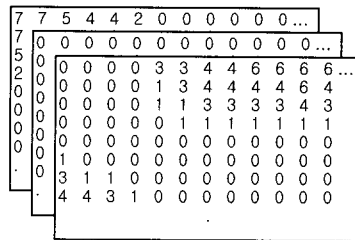


(a) 처음 훈련 후 (b) 한번의 사용자 피드백을 거친 후

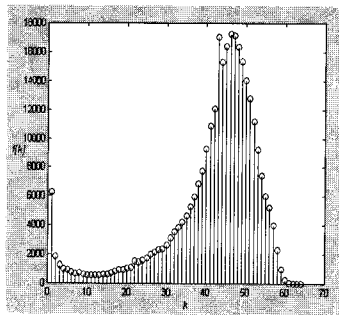
그림 4 RBFN 훈련에서 주기에 따른 오류의 변화



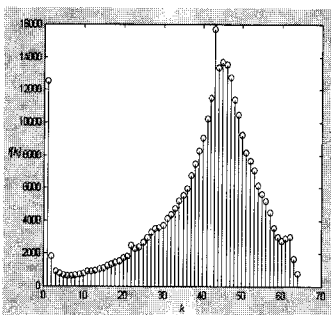
(a) 예제 영상



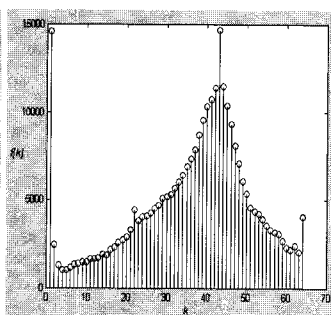
(b) (a)의 480×640×3 자료 행렬



(c) (a)의 빨간색 히스토그램



(d) (a)의 녹색 히스토그램



(e) (a)의 파란색 히스토그램

그림 5 예제 영상과 히스토그램

결과적으로 하나의 클래스가 7개의 영상을 포함하는 145 개의 클래스로 구성된 1,015 개(7×145 = 1,015)의 영상이 생성되었다. 구현된 NNIR은 29 개의 입력 뉴런으로부터 각 영상 쌍의 차이 벡터를 입력받고 하나의 출력으로 유사성의 정도를 나타낸다. RBFN의 훈련은 평활요소의 값을 1로, 중심값의 최대 수를 130으로, 그리고 자승합오차 목표 인자는 0.005로 설정하고, 출력(유사성의 값: 자기 자신은 1, 같은 클래스의 다른 영상은 0.9, 다른 클래스의 영상은 0.001)을 알고있는 100 쌍의 영상으로 훈련시켰다. 그림 4는 RBFN의 훈련과정에서 (a) 처음 훈련때와 4장의 D절에서 설명한 (b) 한

번의 사용자 피드백을 거친 후의 훈련때의 주기(epoch) 별 자승합 오차를 나타내는 그래프로서 처음 훈련때는 96번의 주기를 거친후에야 목표 자승합 오차에 도달할 수 있었지만, 한번의 피드백을 이용할 때는 그전의 훈련 결과에 대한 추가적인 훈련이므로 단지 5번의 주기를 거친후에 목표 자승합 오차에 도달함을 볼 수 있다.

실험에 사용된 시각적 특징은 4 절에서 설명한 것처럼 색 특징으로는 히스토그램과 모멘트를, 질감 특징으로는 웨이블릿 변환결과를 사용하였다. NNIR은 필요에 따라서 다른 시각적 특징들을 쉽게 첨가할 수 있다.

그림 5는 예제 영상 (a)와 데이터베이스에 640×480

×3 크기로 저장된 대응되는 자료 행렬의 일부인 (b), 그리고 식 (4.2)로 계산된 (a)의 R, G, B 히스토그램 분포 (c), (d), (e)를 각각 보여준다.

그림 5(a) 영상을 대상으로 식 (4.3)과 식 (4.4)로 각각 주어지는 빨간색, 녹색 그리고 파란색 히스토그램의 일차 모멘트 f_1^r, f_1^g, f_1^b 와 이차 모멘트 f_2^r, f_2^g, f_2^b 계산 결과는 표 3과 같으며 일차 모멘트와 이차 모멘트 값이 두 배 이상의 차이가 나는 것을 알 수 있다.

표 3 그림 5(a)의 일차 모멘트 (f_1^r, f_1^g, f_1^b)와 이차 모멘트 (f_2^r, f_2^g, f_2^b) 값들

특징	f_1^r	f_1^g	f_1^b	f_2^r	f_2^g	f_2^b
값	164.4767	156.9347	141.9315	49.3602	57.9960	62.0971

그림 6은 그림 5(a) 영상에 웨이블릿 변환을 표 1에 나와있는 방법대로 3단계까지 적용한 결과이고 이들 각각 사각형의 평균과 분산 $f_{10}^g, \dots, f_{29}^g$ 의 값이 표 4에 나열되었다.

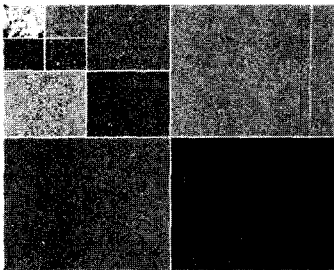


그림 6 그림 5(a)를 3단계까지 웨이블릿 변환시킨 후의 영상

표 4 그림 5(a)의 웨이블릿 상수 행렬의 평균 ($f_{10}^g, \dots, f_{19}^g$)과 분산 ($f_{20}^g, \dots, f_{29}^g$)

특징	f_{10}^g	f_{11}^g	f_{12}^g	f_{13}^g	f_{14}^g	f_{15}^g	f_{16}^g	f_{17}^g	f_{18}^g	f_{19}^g
값	442.5540	-1.0145	0.1434	0.1380	0.0208	0.0787	0.0464	0.0050	0.0103	0.0018
특징	f_{20}^g	f_{21}^g	f_{22}^g	f_{23}^g	f_{24}^g	f_{25}^g	f_{26}^g	f_{27}^g	f_{28}^g	f_{29}^g
값	367.6213	52.1097	51.2923	29.8753	16.1537	15.8936	6.3195	4.3875	3.8838	0.5927

표 5 그림 5(a)와 그림 7의 특징 벡터의 차이값 (v_1, \dots, v_{29})

특징	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
값	0.5941	0.5610	0.4464	48.4742	69.9458	75.3819	-38.4752	2.1076	-3.7060	181.4124
특징	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}
값	-0.7822	-0.8518	0.7465	0.3588	0.0661	-0.0892	-0.0302	-0.0077	0.0033	82.0145
특징	v_{21}	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}	v_{29}	
값	-11.7534	-8.7286	-18.4373	-15.1434	-15.3010	-13.7104	-2.2044	-2.6546	-2.3872	

한편, 질의 영상 q 가 그림 5(a)이고 비교되는 영상 t 가 그림 7일때 계산되는 비교값 $v_i, i=1,2,\dots,29$,는 표 5와 같다. 표 5에서 보듯이 비교값들의 크기는 각각 다양하고 더욱이 저차원의 특징 표현들에 대한 심도있는 지식을 필요로 하므로 사용자가 일일이 가중치를 부여하여 결합시키는 것은 어렵다는 것을 알 수 있다. 제안되는 모델에서는 표 5의 비교값들을 미리 훈련된 신경망을 통하여 결합시킨다.



그림 7 비교 영상

그림 8은 그림 5(a) 영상에 대한 MC 결과를 나타낸다. 그림 8에 나타나는 영상들은 비교값들을 결합시키기 위하여 미리 훈련된 가중치를 사용한 결과로서 질의 영상과 유사한 7개의 영상 중에서 2개밖에 찾지 못한 것을 볼 수 있다. 이것은 질의 영상이 훈련 자료에 포함되지 않은 것과 훈련 결과가 완전하지 않은 것에 기인하는 것으로 사용자의 피드백을 통한 추가적인 훈련의 필요성을 보여준다.

그림 9는 질의 영상인 그림 5(a)에 대한 IL후의 검색 결과로서, 사용자의 피드백을 이용한 추가적인 훈련을

함으로써 검색 성능을 향상시킬 수 있음을 보여준다.

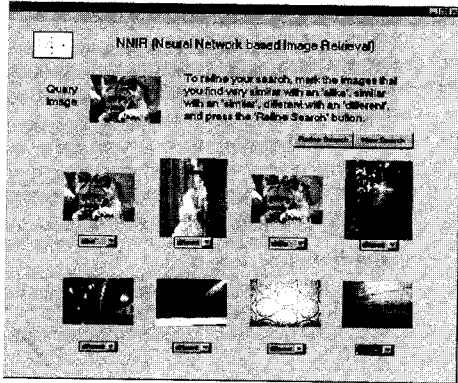


그림 8 그림 5(a)가 질의 영상일 때의 첫 번째 검색 결과 (7개의 유사 영상 중 2개 찾음)

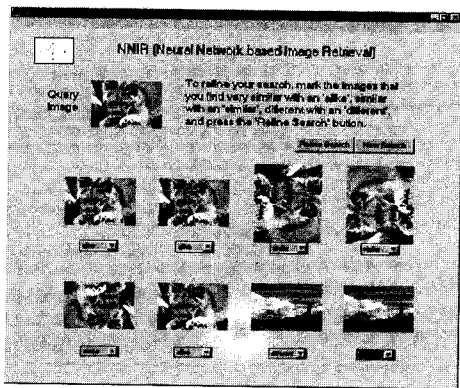


그림 9 질의 영상 그림 5(a)에 대한 II후의 검색 결과 (7개의 유사 영상 중 6개 찾음)

제안된 기법의 성능을 측정하기 위하여 두 가지 통계적 척도를 사용하였으며, 이는 정보 추출(information retrieval)에서 공통적으로 사용되는 재생(recall)과 정도(precision)이다. 재생은 검색 결과로 화면에 보여지는 영상의 수인 창 의 크기 W 가 주어질 때 데이터베이스에 있는 모든 유사 영상(같은 클래스의 영상) 중에서 W 에

포함되는 유사 영상(즉, W 안에서 질의 영상과 같은 클래스에 속하는 검색된 영상)의 비율이다.

$$\text{재생(recall)} = \frac{W \text{에 포함되는 검색된 유사 영상의 수}}{\text{데이터베이스에 있는 모든 유사 영상의 수}}$$

이 척도는 창 의 크기 W 가 커짐에 따라 단조 증가하게 되고 W 가 전체 데이터베이스를 포함하면 100%가 된다. 표 6은 창 의 크기 $W=8$ 인 경우의 재생 값을 보여주며 본 실험에서는 한 클래스의 크기가 7이므로 위 재생 식의 분모는 7로 고정된다.

정도는 100% 재생에 해당하는, 즉 검색된 마지막 유사 영상까지의 갯수중에서 유사 영상 수의 비율이다:

$$\text{정도(precision)} = \frac{\text{유사 영상의 수}}{\text{마지막 유사 영상까지 검색된 영상의 수}}$$

따라서 높은 값의 정도는 검색된 영상들의 대부분이 유사 영상이라는 것을 의미한다.

그림 10과 표 6은 NNIR, BP, 선형 결합 방법, 그리고 순위 기반 방법(최소, 평균, 최대)의 성능을 보여준다.

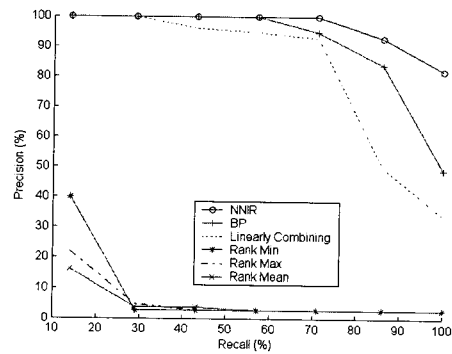


그림 10 47개의 질의 영상에 대한 평균 검색 성능 비교

그림 10과 표 6에서 각 방법의 재생과 정도는 47개의 질의에 대한 평균값을 나타낸 것이다. 각 실험에서 질의 영상은 데이터베이스에서 임의로 선택되고 데이터베이스의 나머지 영상들과 비교되었다. NNIR의 경우에 화면에 출력되는 8개의 영상 중에서 평균적으로 93.45%가 유사 클래스에 속하고 유사 영상을 모두 출력하기 위해

표 6 47개의 질의 영상에 대한 검색 결과

	NNIR	BP	선형 결합 방법	순위 기반 방법(최소)	순위 기반 방법(평균)	순위 기반 방법(최대)
Recall ($W=8$)	93.45 %	81.52 %	78.27 %	8.79 %	4.40 %	7.69 %
Precision (100% recall)	80.61 %	48.37 %	36.94 %	2.61 %	2.76 %	2.57 %

서 필요한 영상들 중에서 평균적으로 80.61%가 유사 영상이다. 이러한 결과는 제안된 방법인 NNIR이 선형 결합 방법이나 순위 기반 방법 그리고 역전파 알고리즘을 이용한 방법보다 더 뛰어난 재생과 정도 성능을 가짐을 보여준다. 특히 정도의 값은 NNIR이 다른 방법들보다 32% 이상의 높은 성능을 보이는데 이는 제안되는 모델에서 사용자의 피드백을 이용하는 추가적인 훈련 과정이 성능 향상에 큰 역할을 하는 것에 기인한다.

6. 결론 및 향후연구

본 논문에서는 내용 기반 영상 검색을 위한 한 방법으로 인간과 컴퓨터의 상호 작용을 포함하는 신경망 기반 영상 검색 모델을 제안하였다. 사용자가 자신이 원하는 영상에 대한 세부 정보를 각 특징값으로 직접 나타내야 하고, 또 각 특징과 연관된 가중치를 직접 조정해야 하는 선형 결합 방법과는 달리, 제안된 방법은 사용자가 처음에 질의 영상만을 선택하고 피드백을 통하여 점차적으로 목표 영상을 찾아나갈 수 있게 한다. 이 방법은 RBFN을 이용하여 특징들 사이의 비선형적 관계를 추출할 수 있으므로 영상 사이의 유사성 비교를 더 정확하게 할 수 있다. 실험 결과는 제안된 방법의 재생과 정도가 각각 93.45%와 80.61%로서 기존의 선형 결합 방법이나 순위 기반 방법 그리고 역전파 알고리즘을 이용한 방법보다 재생은 최소 11.9% 이상 그리고 정도는 최소 32.2% 이상의 더 뛰어난 검색 성능을 지님을 보여준다.

하지만, 현 단계에서는 NNIR에 적용된 증가 학습 방법이 기능적으로 구현된 수준이므로 효율성 및 최적화의 면에서 향후 개선되어야 할 필요가 있고 보다 정확한 실험을 위하여 실험 데이터베이스의 다양화와 검색시간의 비교를 포함한 다양한 형태의 비교 실험이 필요하다. 그리고 제안된 검색 모델이 내용 기반 영상 검색을 위한 것이지만, 비디오나 오디오와 같은 다른 미디어 형식을 다루기 위해서도 확장될 수 있을 것이다.

참고 문헌

- [1] Y. Rui, T. S. Huang, and S. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues", *Journal of Visual Communication and Image Representation*, Vol.10, pp. 39-62, 1999
- [2] R. Jain, "Workshop report: NSF workshop on visual information management systems", *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases I*, pp. 198-218, San Jose, CA, 1993
- [3] T. P. Minka and R. W. Picard, "Interactive Learning using a 'Society of Models'", *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 447-452, San Francisco, CA, 1996
- [4] W. Y. Ma and B. S. Manjunath, "Texture Features and Learning Similarity", *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 425-430, San Francisco, CA, 1996
- [5] J. R. Smith and S. F. Chang, "Visually Searching the Web for Content", *IEEE Multimedia Mag.*, Vol.4, pp. 12-20, Summer 1997
- [6] S. Haykin, *Neural Networks*, 2nd ed., Prentice Hall, New Jersey, 1999
- [7] M. Wood, N. Campbell, and B. Thomas, "Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval", *Proc. ACM Multimedia 98*, pp. 13-20, Bristol, UK, September 1998
- [8] R. M. Rao and A. S. Bopardikar, *Wavelet Transforms*, Addison-Wesley, Massachusetts, 1998
- [9] B. M. Mehre, M. 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
- [10] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases", *Proc. SPIE Conf. Storage and Retrieval of Image and Video Databases II*, pp. 34-47, San Jose, CA, 1994
- [11] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and B. Dom et al., "Query by Image and Video Content: The QBIC System", *IEEE Computer*, Vol.28, No.9, pp. 23-32, 1995
- [12] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Jain, and C. F. Shu, "The Virage Image Search Engine: An Open Framework for Image Management", *Proc. SPIE Conf. Storage and Retrieval for Still Image and Video Databases IV*, pp. 76-87, San Jose, CA, February 1996
- [13] W. Y. Ma and B. S. Manjunath, "NETRA: A Toolbox for Navigating Large Image Databases", *Proc. IEEE Int. Conf. Image Processing*, pp. 568-571, Washington, DC, 1997
- [14] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "NeuroMerge: An Approach for Combining Heterogeneous Features in Content-based Image Retrieval Systems", *Proc. 4th Int. Workshop on Multi-Media Database Management Systems (IW-MMDBMS'98)*, pp. 106-113. Dayton, Ohio, August 1998

- [15] T. P. Minka, "An Image Database Browser that Learns from User Interaction", *Technical Report 365*, MIT Media Laboratory Perceptual Computing, 1996
- [16] F. Liu and R. Picard, "Periodicity, Directionality, and Randomness: World Features for Image Modeling and Retrieval", *Technical Report 320*, MIT Media Laboratory Perceptual Computing, 1996
- [17] H. J. Zhang and D. Zhong, "A Scheme for Visual Feature Based Image Retrieval", *Proc. SPIE Conf. Storage and Retrieval for Image and Video Database III*, pp. 36-46, San Jose, CA, 1995
- [18] J. Dowe, "Content-Based Retrieval in Multimedia Imaging", *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases I*, pp. 164-167, San Jose, CA, 1993
- [19] J. Park and I. W. Sandberg, "Universal Approximation using Radial-Basis-Function Networks", *Neural Computation*, Vol.3, No.2, pp. 246-257, 1991
- [20] S. Chen, C.F.N.Cowan and P.M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks", *IEEE Tr. on Neural Networks*, Vol.2, No.2, pp. 302-309, March 1991



이 형 구

1993년 서울대학교 계산통계학과 학사.
 1996년 서울대학교 전산과학과 석사.
 2001년 서울대학교 컴퓨터공학부 박사.
 2001년 9월 ~ 현재 한국전자통신연구원
 영상정보처리연구팀 선임 연구원. 관심분
 야는 내용 기반 영상 검색, 신경망, 컴퓨

터 비전, 바이오인포매틱스



유 석 인

1977년 서울대학교 전기공학과 졸업(학
 사). 1980년 미국 Lehigh University 전
 산공학(석사). 1985년 미국 University
 of Michigan 전산공학(박사). 1985년 ~
 현재 서울대학교 공과대학 컴퓨터공학부
 교수. 관심분야는 인공지능, 특히 지식기

반 시스템, 기계학습, 컴퓨터비전, Planning, 객체지향 시스
 템 등