

동영상 등장인물의 자동색인을 위한 효율적인 저장과 검색 방법

김진승[†], 한용구^{**}, 이영구^{***}

요 약

대용량 동영상을 대상으로 한 등장인물 색인에 대한 수요가 증가함에 따라, 많은 시간과 비용이 소요되는 수동 태깅의 단점을 보완할 수 있는 자동 태깅을 이용한 자동 색인이 연구되고 있다. 하지만, 자동 색인은 인물을 100% 정확하게 검출하지 못하므로 검출된 인물에 대해 정확도를 함께 표현해야 한다. 본 논문에서는 이러한 정보를 포스팅 리스트에 효율적으로 저장하는 방법과, 등장인물의 검색 시 관련 동영상들을 효율적으로 찾기 위한 순위 결정 방법을 제안한다. 실험을 통하여 제안하는 색인 정보 저장 방법이 포스팅 리스트의 압축에 효과적임을 입증하였다. 또한 제안한 순위 결정 방법이 관련 동영상을 찾는 데 효과적임을 입증하였다.

Efficient Storage and Retrieval for Automatic Indexing of Persons in Videos

Jin-Seung Kim[†], Yong-Koo Han^{**}, Young-Koo Lee^{***}

ABSTRACT

With increasing need for indexing of persons in a large video database, automatic indexing has been attracting great interest which takes advantage of automatic tagging instead of the time-consuming and costly manual tagging. However, automatic indexing approach should provide a degree of recognition proximity because it cannot identify the persons with accuracy of 100%. In this paper, we propose an efficient storage method for storing posting lists efficiently and a novel ranking technique of ordering relevant videos for efficient retrieval. Through experiment evaluations we have shown that our storage method exhibits good performance in compressing the posting list. We have also shown that the proposed ranking method is effective for finding relevant videos.

Key words: Video Indexing(동영상 색인), Frame Based Indexing(프레임 기반 색인), Face Indexing on Video(동영상 인물 색인), Archiving Management System(저장 관리 시스템)

1. 서 론

정보통신기술의 발달로 기록 자료들은 문서 및 사진 중심에서 동영상 및 음성 자료로 멀티미디어화가

되고 있다. 동영상 자료들은 기존의 자료들에 비하여 대용량 일뿐만 아니라 빠른 속도로 증가하여 대량으로 축적되고 있다[1]. 기존에는 동영상 파일의 등장인물 검색을 위하여 인물 또는 인물과 관련된 정보들

※ 교신저자(Corresponding Author) : 이영구, 주소: 경기도 용인시 기흥구 서천동 1번지(446-906), 전화: 031) 201-3732, FAX: 031) 202-2520, E-mail: yklee@khu.ac.kr
접수일: 2010년 12월 11일, 수정일: 2011년 6월 27일
완료일: 2011년 7월 12일

[†] 준회원, 경희대학교 컴퓨터공학과
(E-mail: jinseungk@gmail.com)

^{**} 준회원, 경희대학교 컴퓨터공학과
(E-mail: ykhan@khu.ac.kr)

^{***} 종신회원, 경희대학교 컴퓨터공학과 부교수

※ 본 연구는 행정안전부 국가기록원의 지원을 받아 기록물 보존기술 연구개발(R&D) 사업의 일환으로 이루어졌으며, 이에 감사드립니다.

을 사람이 직접 수동으로 태깅을 하고, 키워드 기반의 질의를 수행하는 방식을 주로 사용하였다. 그런데 이와 같은 방법은 동영상이 대량 및 대용량화가 됨에 따라 수동으로 태깅할 경우 많은 시간이 소요되기 때문에 비효율적이다.

최근 영상 처리 기술이 발달함에 따라 동영상 내에서 얼굴 인식 기술을 이용하여 동영상을 자동 색인하는 연구[2-4]가 활발히 이루어지고 있다. 자동 색인은 그림 1과 같이 동영상의 각 프레임에 대해 얼굴 영역들을 검출하고, 각 얼굴에 대해 인물 데이터베이스를 이용해 인물을 식별하여 자동 태깅한 후 색인에 삽입하는 기술이다. 이 기술은 인물 검색 시스템, 방송 영상편집 시스템, 기록물 관리 시스템 등 다양한 응용 분야에서 활용이 가능하다.

그런데, 얼굴 인식 기술을 활용한 자동 색인 기술은 사람이 수동으로 태깅을 수행하는 비효율성은 해결하지만, 등장인물을 100% 정확하게 검출하지 못하므로 검출된 인물에 대하여 인식 정확도(*proximity*)를 함께 표현한다. 따라서 등장인물 검색을 위한 색인을 구성 시, 색인 정보에는 정확도를 포함하는 (동영상 파일 ID, 프레임 번호, 등장인물, 정확도)와 같은 구조로 제공되어야 한다. 예를 들어 (V1, 345, 홍길동, 80%)는 “동영상 V1의 345번째 프레임에 홍길동과 80% 유사한 인물이 등장함”을 의미한다. 이와 같은 정보는 또한 기존 문서 기반의 키워드 색인 방법[5]과 유사하게 “동영상 ID-문서 ID”, “프레임 번호-단어의 문서 내 위치”, “등장인물의 출현 빈도-특정 단어의 출현 빈도(*term frequency*)”와 같이 매핑되어 역색인(*inverted index*)을 구성함으로써 등장인물이 나타나는 프레임 위치까지 검색할 수 있는 프레임 기반 검색(*frame based search*)을 가능하게 한다. 그러나 기존의 문서 검색을 위한 포스팅 리스트에 포함되는 색인 정보[6]와 그 구성이 다르기

때문에 새로운 포스팅 리스트는 압축 방법을 고려한 효율적인 저장방법이 요구된다.

본 논문에서는 등장인물 자동 색인을 위한 색인 정보를 포스팅 리스트에 효율적으로 표현하는 방법과, 색인 정보를 이용하여 등장인물 검색 시 유사한 동영상들을 효율적으로 찾기 위한 순위 결정 방법을 제안한다. 색인 정보의 효율적인 표현을 위하여, 동영상에서 등장인물은 임의의 프레임부터 연속적으로 나타나는 특성을 이용한다. 등장인물이 나타나는 각각의 프레임들의 리스트를 유지하는 대신, 비슷한 유사도를 갖는 연속적인 프레임들을 프레임 구간으로 표시함으로써 정보 손실을 최소화하는 효율적인 압축 기법을 제시한다. 또한, 등장인물 검색 시 유사한 동영상의 순위를 결정하기 위하여 색인 정보로부터 등장인물의 정확도, 등장인물이 나타나는 시간 등을 계산하여 활용한다. 등장인물 색인 정보를 활용하여 순위를 결정함으로써 단순히 태깅에 의존하는 기존의 방법과 비교하여 더 정확한 검색할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 동영상에서 등장인물 검색을 위한 색인 방법과 기존 문서 기반의 포스팅 리스트 압축 및 순위 측정 방법들에 대해 소개한다. 3장에서는 프레임 기반 등장인물 검색을 위한 포스팅 리스트 압축 방법을 제안하고, 4장에서는 등장인물 질의에 대한 동영상의 순위 측정에 대한 방법을 제안한다. 5장에서는 제안하는 방법들의 성능을 평가하고 6장에서는 결론을 내린다.

2. 관련연구

2.1 동영상 내 등장인물 검색을 위한 색인 방법

일반적으로 문자 기반의 콘텐츠를 갖는 문서와 웹 페이지는 키워드를 기반으로 색인을 구성할 수 있다. 그러나 동영상은 음성과 이미지의 집합으로 이루어진 콘텐츠를 갖기 때문에 문서와는 다른 색인 대상 및 과정이 필요하다. 동영상은 콘텐츠의 내용을 공간 색인을 통하여 직접적으로 구성하는 방법과 음성 및 이미지의 집합을 해석함으로써 키워드를 추출하여 키워드에 대한 색인을 구성하는 방법이 있다[6]. 본 논문은 동영상에서 프레임 별로 인식된 등장인물과 인식 정확도를 이용하여 색인 구성 및 검색에 관한 연구로서 후자의 방법과 관련된 연구이다. 동영상의 등장인물 검색에 있어서 등장인물 인식의 정확도는

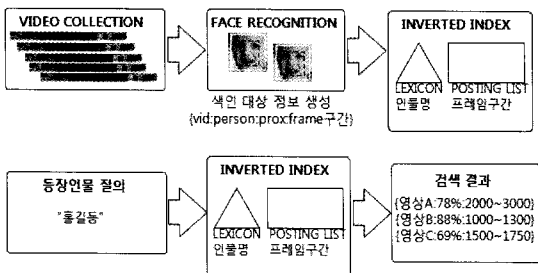


그림 1. 얼굴 인식을 이용한 자동 색인

질의의 범위를 확장하게 되며, 기존의 문서나 웹 페이지에서 나타나는 키워드, 컨텐츠 내의 키워드의 위치, 키워드의 출현 빈도 외에 인식의 정확도라는 새로운 속성이 추가된다. 따라서, 동영상에서 프레임 수준의 등장인물 검색을 위한 색인의 대상은 등장인물, 해당 인물의 동영상 내의 위치(프레임번호), 해당 인물의 출현 빈도, 특정인물과 등장인물 간 인식 정확도들이 된다.

2.2 문서 기반 포스팅 리스트의 압축 방법

문서 기반의 키워드 색인 방법에서 사용되는 인덱스는 식(1)과 같은 구조를 가진다.

$$\langle f_i; d_1, d_2, \dots, d_{f_i} \rangle \quad (1)$$

인덱스 구조에서 f_i 는 $1 \leq d_1 < d_2 < \dots < d_{f_i} \leq N$ (N 은 전체 문서의 개수)의 범위를 가지고 t 라는 키워드를 포함하는 문서의 개수이다[7]. 비슷한 구조로 d_n 에서 t 의 출현위치를 식(2)와 같이 표현할 수 있다.

$$\langle f_i; pos_1, pos_2, \dots, pos_{f_i} \rangle \quad (2)$$

이러한 리스트의 압축방법으로는 각 요소의 차이를 이용하여 숫자를 작은 숫자로 표현하는 방법과 숫자를 인코딩하여 보다 작은 비트를 사용하도록 하는 방법이 있다. 리스트 요소의 차이를 이용하는 방법은 식(3)과 같이 표현된다.

$$\begin{aligned} &\langle f_i; d_1, d_2 - d_1, d_3 - d_2, \dots, d_{f_i} - d_{f_i-1} \rangle \quad (3) \\ &\langle f_i; pos_1, pos_2 - pos_1, pos_3 - pos_2, \dots, pos_{f_i} - pos_{f_i-1} \rangle \end{aligned}$$

예를 들면, apple이라는 키워드에 대하여 다음과 같이 표현하였다면,

$$\langle 5; 100, 410, 520, 1002, 1005 \rangle$$

이는 apple이라는 키워드가 100번, 410번, 520번, 1002번, 1005번 문서에 포함되어있다는 의미이며, 다음과 같이 각 문서 번호 사이의 차이를 이용하여 압축하여 표현할 수 있다.

$$\langle 5; 100, 310, 110, 482, 3 \rangle$$

원래 숫자보다 작게 표현된 리스트는 var-byte Compression, Rice Coding, Golomb Coding 등의 Integer Compressing 방법을 통하여 더 작은 비트를 이용하여 저장되어진다[7].

2.3 문서 순위 측정 방법

문서들의 집합에서 질의와 가장 유사한 문서의 순

위를 측정하는 방법 중 벡터 공간 모델은 문서 내에서 키워드들 간의 거리를 이용하여 질의어와의 유사도를 계산하기 위한 수학적 모델이다[8-10]. 이 모델에서 문서(d_n)은 색인 단어에 대한 문서 벡터($\vec{V}(d_n)$)로 나타낼 수 있고, 문서의 유사도는 식(4)와 같이 문서 벡터 사이의 코사인 유사도를 통하여 계산할 수 있다.

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|} \quad (4)$$

벡터 공간 모델을 사용하기 위해서는 벡터 공간상의 키워드(Term)의 가중치(weight)를 계산해야 하며, Term Frequency(TF)와 Inverse Document Frequency(IDF)를 이용한다. TF는 문서 Vector 내에 포함된 키워드의 개수이며, IDF는 키워드를 벡터에 포함하고 있는 문서와 전체 문서에 대한 평가값을 나타낸다. 가중치는 TF와 IDF의 곱으로 나타낼 수 있으며, 식(5)와 같이 표현된다.

$$\begin{aligned} idf_t &= \log \frac{N}{df_t} \quad (5) \\ tfidf_{t,d} &= tf_{t,d} \times idf_t \end{aligned}$$

df_t 는 t 를 포함한 문서의 개수이고 N 은 전체 문서의 개수이다.

3. 동영상에서 등장인물 검색을 위한 효율적인 포스팅 리스트 저장 방법

본 장에서는 동영상 등장인물 검색을 위한 효율적인 포스팅 리스트 저장 방법에 대하여 설명한다. 인물 인식 기반 이용한 자동 색인 기술은 등장인물에 대한 인식률이 100%가 아니기 때문에 포스팅 리스트에 정확도라는 새로운 요소가 포함된다. 따라서 (등장인물 ID, 정확도, 등장인물이 나타나는 프레임의 번호)와 같이 기존과는 다르게 포스팅 리스트를 구성 및 저장해야 한다.

또한 등장인물이 나타나는 프레임 위치 수준까지 검색할 수 있는 프레임 기반 검색을 위한 색인 구성 시 대량의 색인 정보가 생성되는 문제가 발생한다. 실제 29.97 fps(frames per second)의 프레임 율을 갖는 1시간 분량의 동영상은 107,892개의 프레임들로 구성되기 때문에 하나의 동영상에서 10만개 이상의 색인 정보가 생성될 수도 있다. 이와 같은 색인 정보의 증가는 색인 저장 공간의 낭비와 함께 질의

처리 시간이 길어지는 원인이 되기 때문에 새로운 포스팅 리스트 압축 방법이 요구된다.

동영상 내 등장인물에 대한 프레임 수준의 색인을 구성하기 위한 동영상의 집합(Video Collection)은 $VC = \{V_1, V_2, \dots, V_N\}$ 으로 표현한다. VC로부터 색인을 위해 각 동영상 V_i 는 식(6)과 같은 등장인물 리스트를 갖는다.

$$PersonList_{V_i} = \{PL_{i1}, PL_{i2}, \dots, PL_{im}\} \quad (6)$$

여기서 PL_m 는 포스팅 리스트로서 등장인물, 인식된 인물의 정확도, (인물의 등장이 시작되는 프레임 번호, 인물의 등장이 끝나는 프레임번호)의 정보를 가지는 식(7)과 같은 구조를 가진다..

$$PL = (person; prox; (FrameInfo)_1, (FrameInfo)_2, \dots, (FrameInfo)_n) \\ FrameInfo = (startFrame, endFrame) \quad (7)$$

이러한 구조는 동영상 길이 l 때, $startFrame$ 과 $endFrame$ 으로 표현되는 값이 저장 공간을 많이 사용하기 때문에 이를 압축하기 위한 방법으로 문서기반의 포스팅 리스트에서 차이를 이용한 방법[6]을 이용하여 $FrameInfo$ 의 $endFrame$ 을 해당 구간의 길이로 표현하고, $startFrame$ 을 $FrameInfo_n.startFrame$ 과 $FrameInfo_{n+1}.startFrame$ 의 차이를 이용하여 식(8)과 같은 구조로 표현할 수 있다.

$$PL = (person; prox; (FrameInfo_1.startFrame, length)_1, ((FrameInfo_1.startFrame - FrameInfo_2.startFrame), length)_2, \dots, ((FrameInfo_{n-1}.startFrame - FrameInfo_n.startFrame), length)_n) \quad (8)$$

예를 들면, $PL = (\text{홍길동}; 0.70; (1000, 1200), (1500, 2000), (2200, 2550))$ 은 0.70의 정확도로 인식된 홍길동이라는 인물이 프레임번호 1000~1200, 1500~2000, 2200~2550 구간에서 나타난다는 것을 의미하며 이것은 식(9)와 같이 표현될 수 있다.

$$PL = (\text{홍길동}; 0.70; (1000, 200), (500, 500), (700, 550)) \quad (9)$$

이러한 프레임 번호의 표현에 의한 단순한 압축 방법 외에 본 논문에서는 비연속적인 두 프레임 구간의 병합에 의한 포스팅 리스트 압축 방법을 제안한다. 등장인물에 대한 색인을 구성 시에 연속적인 프레임들에 대해서만 구간으로 표현을 하면 너무 많은 수의 프레임 구간이 생성되어 포스팅 리스트의 크기가 커지는 현상이 발생한다. 실제로 일반적인 동영상에서 10 프레임의 차이는 0.3초 정도에 해당하는 매우 짧은 시간이기 때문에 서로 분리된 프레임 구간으로 취급할 필요가 없다. 따라서 구간 사이의 거리가 임계 조건을 만족할 때 두 구간을 병합하여 하나의 구간으로 간주함으로써 포스팅 리스트의 크기를 줄인다. 예를 들어 프레임 구간으로 구성된 포스팅 리스트 {홍길동, 80%, (345, 560)}과 {홍길동, 80%, (570, 650)}을 {홍길동, 80%, (345, 650)}으로 병합한다.

그런데, 이러한 리스트는 각 동영상에 대해서 인물별 정확도로 작성된다. 하나의 동영상에서 색인 대상의 프레임 구간들은 인물 및 인식 정확도별로 다양하게 나타난다. 정확도와 작성된 동영상들은 그림 2와 같이 하나의 인물에 대해 여러 가지 정확도를 가지게 된다.

따라서, 그림 2와 같은 프레임 구간을 병합 시, 각 프레임 구간들 간의 거리뿐만 아니라 정확도까지 고려해야 한다. 서로 다른 프레임 구간의 병합 시, 해당 구간들의 길이(L_j)와 정확도(P_i)를 이용하여 식(10)과 같이 정확도와 길이의 가중 평균을 계산한다.

$$weightedavg = \frac{\sum(P_i \times L_j)}{\sum L_j} \quad (10)$$

만약 b와 d의 병합, d와 g의 병합과 같이 구간과 구간간의 거리가 큰 경우에는 식(11)과 같은 가중 평균 식을 사용한다.

$$weightedavg = \frac{\sum(P_i \times L_j) + \sum(L_{j,j+1} \times \frac{\min P_i}{2})}{\sum L_j + \sum L_{j,j+1}} \quad (11)$$

식(11)에서 $L_{j,j+1}$ 은 두 구간 L_j 와 L_{j+1} 사이의 빈

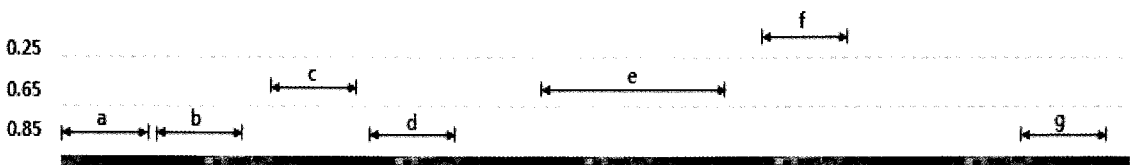


그림 2. 정확도별 프레임 구간 리스트 예

구간의 거리이고 $\min P_i$ 는 얼굴 인식 모듈에서 출력 여부를 결정할 때 사용하는 파라미터로 최소 정확도를 나타낸다. 따라서, 빈 구간의 정확도는 0부터 $\min P_i$ 구간 사이에 존재한다. 빈 구간의 정확도는 중간값인 $\frac{\min P_i}{2}$ 를 사용하기로 한다.

병합 수행 여부를 평가 시 두 구간에 대하여 계산된 가중 평균값과 병합 이전의 각 구간의 가중 평균을 비교하여 주어지는 병합 오차율 한도 ϵ_{th} 보다 작으면 병합을 수행하여, 각 구간은 새로운 하나의 프레임 구간으로 대체된다. 식(8)은 병합 조건을 수식으로 표현한 것이다. 식(12)의 병합 수행 조건에 만족

하여 병합을 수행하면 대상 구간은 새로운 프레임 구간으로 대체되게 된다.

$$\begin{aligned} \text{병합대상구간의 가중평균 } A &= \frac{\sum(P_i \times L_j)}{\sum L_j}, B = \frac{\sum(P_i \times L_{j+1})}{\sum L_{j+1}} \\ \text{병합 후의 가중평균 } C &= \frac{\sum(P_i \times L_j) + \sum(L_{j+1} \times \frac{\min P_i}{2})}{\sum L_j + \sum L_{j+1}} \\ \text{병합수행 조건 } \epsilon_{th} &\geq \frac{|A-C|}{|A|} \text{ 이고, } \epsilon_{th} \geq \frac{|B-C|}{|B|} \end{aligned} \quad (12)$$

이러한 프레임 구간간의 병합을 통하여 프레임 리스트의 수를 줄임으로써, 색인의 크기와 질의 처리 속도를 향상 시킬 수 있다. 표 1은 동영상 인물 색인을 위해 하나의 동영상에 대해서 포스팅 리스트를

표 1. 동영상 인물 색인의 포스팅 리스트 압축 의사코드

```

Algorithm 1. CompressPostingList(PersonList, threshold)
Input : PersonList - 하나의 영상에 대한 등장인물들의 포스팅 리스트들의 리스트
        threshold - 병합 오차율
1: for each person in PersonList do
2:   get posting lists of the current person;
3:   construct FrameList for the current person;
   // FrameList = (person; FrameInfoWithProx1, ..., FrameInfoWithProxi)n
   // FrameInfoWithProx = (FrameInfoj, proxj)
4:   MergeFrameList(FrameList, threshold);
    
```

```

Algorithm 2. MergeFrameList(FrameList, threshold)
1: startpoint=0; //startpoint : 병합 시작점
2: for i=0; i<(length of FrameList)
3:   length, prox, diff = 0; //length : ΣL, prox : ΣP, diff : ΣLi,i+1
4:   if(startpoint-i == 0)
5:     i++;
6:   for j=startpoint; j<i
7:     length+=(length of FrameInfoWithProx(j));
8:     prox+=(length of FrameInfoWithProx(j))*FrameInfoWithProx(j).prox;
9:   for j=startpoint; j<i-1;
10:    diff+=distance between
           FrameInfoWithProx(j) and FrameInfoWithProx(j+1);
11:  evaluateRate=(prox+diff*threshold/2)/(length+diff);
12:  mergeFlag = True; //병합여부, 기본적으로 병합되었다고 가정
13:  for j=startpoint; j<i;
14:    if ((FrameInfoWithProx(j).prox-evaluateRate)
           /FrameInfoWithProx(j).prox > threshold)
15:      mergeFlag =false;
16:      exit for;
17:  if mergeFlag
18:    if i==(length of FrameList)
19:      Merge FrameInfoWithProx(startPoint), FrameInfoWithProx(i);
20:    startPoint = i + 1;
21:  else
22:    Merge FrameInfoWithProx(startPoint), FrameInfoWithProx(i-1);
23:    startPoint = i;
    
```

압축하는 알고리즘 CompressPostingList와 CompressPostingList에서 개별 인물의 포스팅 리스트를 압축하기 위해 호출하는 알고리즘 MergeFrameList의 의사코드이다.

알고리즘 CompressPostingList는 하나의 동영상에 대해 등장인물과 인식 정확도별로 구성되는 PersonList와 병합 오차율을 입력 인자로 호출된다. 즉, 식(6)과 (7)에서 표현된 PersonList를 입력 인자로 받게 된다. 한 명의 등장인물마다 그림 2와 같은 여러 인식 정확도별로 구성되는 다수의 PL(Posting-List)를 가지게 된다. 각 등장 인물에 대해 다음을 수행한다. 2-3행에서 현재 처리할 등장 인물에 대해 인식 정확도별로 나타난 다수의 포스팅 리스트들을 정확도 구분 없이 프레임 순서에 따라 정렬한 통합된 FrameList를 구성한다. 이 때, 인식 정확도의 정보 손실 막기 위해 Posting List에서 사용된 FrameInfo의 구조에 인식 정확도를 포함한 FrameInfoWith-Prox (= (FrameInfo, prox))로 변환하여 사용한다. 4행에서 이렇게 재구성한 인물별 FrameList에 포함된 프레임 구간들을 병합하기 위해 알고리즘 Merge-FrameList를 호출한다. 알고리즘 MergeFrameList는 알고리즘 CompressPostingList로부터 전달 받은 프레임 구간 정보를 담은 리스트를 식(12)를 이용하여 병합한다. 알고리즘 MergeFrameList에서 1-5행은 병합 시작점, 병합 구간의 길이, 인식 정확율, 병합 구간간의 거리를 초기화 하는 과정이며, 6-8행은 병합 후의 가중 평균을 계산하기 위해 병합 대상구간의 길이(ΣL_i)와 병합 대상구간의 정확도와 길이의 곱($\Sigma(P_i \times L_j)$)을 구한다. 9-10행은 빈 구간의 거리($\Sigma L_{j,j+1}$)를 구하고, 11행에서 병합 후의 가중 평균을 구한다. 14행은 병합 수행 조건을 평가하여 병합 여부를 판단한다. 병합 여부를 판단 후, 17-23행에서 실제 병합을 수행한 후에 다음 병합의 시작점을 결정한다.

4. 등장인물 질의에 대한 동영상의 순위 측정

등장인물과 관련된 질의에 대하여 $VC = \{V_1, V_2, \dots, V_N\}$ 로부터 해당 인물을 포함한 동영상을 결과로 반환한다. 단순한 태깅 기반의 인물 색인 방법을 사용할 경우에는 해당 인물 태그가 포함된 모든 동영상을 같은 결과로 반환할 것이다. 그러나

자동 색인 방법으로 등장인물 색인이 구성되면 각 프레임별 등장인물 인식의 정확도와 등장인물이 나타나는 프레임 위치 정보를 알 수 있다. 이와 같은 정보를 이용하여 등장인물이 나타나는 유사한 동영상들에 대하여 등장인물과 더욱 관련이 있는 순위를 부여함으로써 더 정확한 질의 결과를 만들 수 있다.

본 논문에서는 각 동영상 및 프레임 별 등장인물 정확도와 동영상 내에서 등장인물이 나타나는 프레임의 시간을 반영할 수 있는 새로운 인물별 동영상 순위 측정 방법을 제안한다. 등장인물에 대한 동영상의 순위를 결정하기 위하여 다음의 3가지 요소를 고려한다.

1. 동영상 내 검색 인물이 등장하는 프레임 리스트들의 유사도 평균값: 동영상에 등장하는 인물과 검색 인물과의 유사도를 반영하는 요소이며, 식(12)와 같이 각 가중평균에 대한 평균값을 취하며 0.0에서 1.0 사이의 값을 가진다.

$$proxAvg(V_i, person) = \frac{\sum P_i \times L_j}{\sum L_j} \quad (12)$$

2. 전체 동영상 길이의 평균값과 동영상에서 검색 인물이 등장하는 프레임 길이의 비율: 동영상에 검색 인물이 나타나는 프레임을 얼마나 가지고 있는지에 대한 절대 길이 요소이며, 식(13)과 같이 해당 값은 1.0이상이면 1.0이라고 가정한다. 등장인물의 등장구간이 전체 동영상의 평균 길이보다 길다면, 1.0 이상의 값을 갖게 되며, 이는 평균적인 동영상의 길이만큼 해당인물이 등장한다는 의미이기 때문이다.

$$lengthRate(V_i, person) = \min\left(\frac{\sum L_i}{\sum length(V_n)/N}, 1\right) \quad (13)$$

3. 동영상의 길이와 동영상에서 검색 인물이 나타나는 구간의 비율: 동영상에서 검색 인물이 나타나는 프레임을 얼마나 가지고 있는지에 대한 상대 길이 요소이며, 식(14)과 같이 0.0~1.0사이의 값을 가진다.

$$appRate(V_i, person) = \frac{\sum L_j}{length(V_i)} \quad (14)$$

각각의 요소와 요소별 가중치 α, β, γ ($\alpha+\beta+\gamma=1$)를 이용하여 검색 인물에 대해 동영상별로 순위 점수를 식(15)와 같이 측정한다.

$$\text{순위점수}(V_i, person) = \alpha \times proxAvg(V_i, person) + \beta \times lengthRate(V_i, person) + \gamma \times appRate(V_i, person) \quad (15)$$

계산된 순위점수에 따라 검색 인물과 관련된 동영상의 순위가 정해진다. 이와 같은 프레임 수준의 등장인물 색인 정보를 이용한 순위 결정 방법은 단순히 동영상에 부여한 태깅 정보에 의존하는 키워드 기반 검색보다 더욱 정확한 검색을 효율적으로 수행할 수 있다.

5. 성능 평가

본 장에서는 제안하는 동영상 포스팅 리스트의 압축 방법에 대한 저장 공간 사용량 비교와 동영상 문서들에 대하여 순위를 측정 성능을 평가하였다.

5.1 평가 환경

동영상 포스팅 리스트의 압축 성능과 동영상 문서들에 대한 순위를 측정하기 위한 평가환경은 표 2와 같다.

성능평가를 위한 테스트 데이터는 10편의 역대 대통령 영상으로부터 추출된 대통령들이 등장하는 프레임들이다. 테스트 데이터는 10개의 영상에서 등장하는 7명의 역대 대통령과, 각 영상으로부터 추출된 243개의 프레임 정보들이다. 표 3은 테스트 데이터의 내용에 대해 설명한다. 대통령 영상의 특징상 각 영상들은 대통령의 얼굴이 클로즈-업 되어 나타나는 구간이 많으며, 오래된 기록 영상의 경우 흑백의 영상과 해상도로 인하여, 잘못된 얼굴 인식이 나타나기도 한다. 하지만, 본 논문에서는 추출되어진 영상의

표 2. 성능평가에 사용된 시스템 환경

기종	IBM PC
CPU	Intel Core 2 Duo 2.4GHz
MEMORY	2GB(1066MHz DDR3)
HDD	320GB, 5400RPM, 8MB Buffer, S-ATA
OS	Windows XP Professional

표 3. 테스트 데이터

영상	내용	영상	내용
동영상1	김대중 대통령, 강연	동영상6	박정희 대통령, 기록영상
동영상2	김대중 대통령, 다큐멘터리	동영상7	노무현 대통령, 국민 대담
동영상3	전두환 대통령, 시사기획	동영상8	노무현 대통령, 국민 대담
동영상4	이승만 대통령, 기록영상	동영상9	김영삼 대통령, 홍보영상
동영상5	이명박 대통령, 대국민연설	동영상10	김영삼 대통령, 홍보영상

인물 정보를 효율적으로 저장하는 방법과 동영상의 순위 결정 방법에 대하여 논하므로, 인물 검출에 대한 잘못된 인식 등은 논하지 않는다.

5.2 동영상 포스팅 리스트의 압축 성능 평가

그림 3은 프레임 구간 간의 병합을 평가하는 데 있어서 병합 오차율 한도 ϵ_{th} 을 1%, 5%, 10%, 15%, 20%로 부여하였을 때, 그에 따른 원본 대비 저장 공간의 사용률을 나타낸다. x축은 병합오차율을 나타내며, y축은 병합 오차율에 따른 저장공간의 사용율을 나타낸다. ϵ_{th} 가 0% 이면 프레임 구간 사이에 병합이 일어나기 전과 후의 가중 평균값 변화가 없어야 함을 의미하고, 20%는 병합 전과 병합 후의 가중 평균값 차이가 20% 이내에 존재하여야 병합된다는 것을 의미한다. 그림 3에서 ϵ_{th} 이 1% 인 경우는 압축을 하지 않았을 때와 저장 공간 사용율이 거의 유사하며, 20%에 가까워질수록 프레임 구간 사이에 많은 병합이 발생하여 포스팅 리스트의 압축률이 높아진다. 그림 3에서 “김영삼”, “이승만”에 해당하는 동영상은 압축이 거의 일어나지 않았는데, 해당 동영상에서 “김영삼”, “이승만”에 해당하는 프레임 구간이 매우 적고 각 구간이 멀리 떨어져 있어 병합이 발생하지 않았기 때문이다. 그림 3의 결과에서 포스팅 리스

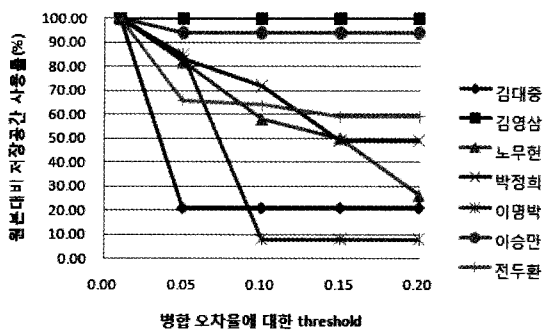


그림 3. 인물별 색인의 원본대비 저장 공간 사용률

트의 압축률이 높아질수록 저장 공간 사용량은 줄어들지만, 병합되는 프레임 구간사이의 거리가 멀어지므로 병합된 프레임 구간에서 찾고자하는 인물이 드물게 나타나게 되는 것을 예상할 수 있다.

5.3 동영상의 순위 측정 성능 평가

제안된 동영상의 순위를 측정하는 기법에는 유사도의 평균값과 길이의 비율, 구간의 비율에 대한 각각의 가중치를 부여하여 동영상의 순위를 측정한다. 실험을 위해, 각각의 가중치를 α, β, γ 라고 할 때 질의어 “김대중”을 통하여, $\alpha = 0.3, \beta = 0.3, \gamma = 0.4$ 로 실험을 한다. 또, 각 가중치를 실험하기 위해 다음과 같은 가중치를 부여하여 실험을 하였다.

$$[\alpha = 1.0, \beta = 0.0, \gamma = 0.0], [\alpha = 0.0, \beta = 1.0, \gamma = 0.0], [\alpha = 0.0, \beta = 0.0, \gamma = 1.0]$$

표 4은 질의어 “김대중”에 해당하는 테스트 데이터이다. 표 5는 테스트 데이터를 이용하여 각 문서에

순위 점수를 측정된 결과를 나타낸다. 각 결과에서 각 평가요소에 가중치를 조절함으로써, 각 가중치별 유효성을 확인하였고, 순위가 가중치별 의미에 맞게 나타나는 것을 표 5의 결과를 통하여 확인 할 수 있다. 예를 들어, $\alpha = 1.0, \beta = 0.0, \gamma = 0.0$ 로 가중치를 부여하였을 경우, α 를 제외한 나머지 두 요소는 무시되고, $\alpha = 0.0, \beta = 1.0, \gamma = 0.0$ 로 가중치를 부여하였을 경우, β 를 제외한 나머지 두 요소가 무시됨으로써, 각 가중치별 의미에 따라 동영상의 순위가 나타나게 된다.

동영상 ID가 1인 동영상은 처음부터 끝까지 “김대중” 대통령에 관련된 동영상이며, 그 정확도도 0.72~0.74 인 관련 영상이다. 동영상 ID가 2번, 8번인 동영상의 경우 정확도와 해당 동영상에서 “김대중” 대통령이 등장하는 구간의 비율에 따라 표 5에서의 순위가 부여됨을 알 수 있다. 예를 들어 동영상 ID가 1번인 동영상은 표 4의 테스트 데이터에서 볼 수 있듯이, 정확도가 다른 동영상에 비해서 높으며, 처음부터 끝까지 “김대중” 대통령이 등장하는 동영상이

표 4. 순위 측정에 사용된 테스트 데이터

동영상ID	정확도	시작프레임	끝프레임	동영상ID	정확도	시작프레임	끝프레임
1	0.749715	1	13479	2	0.638875	9033	9137
1	0.721237	13485	14399	2	0.584701	10753	10759
2	0.753644	1	847	2	0.643939	10905	11025
2	0.679083	1857	2189	2	0.655083	11029	11099
2	0.680303	2713	2832	2	0.671064	11105	11276
2	0.755855	2833	3213	2	0.721217	11329	11518
2	0.838835	3217	3498	2	0.7289	11521	11736
2	0.836977	3545	3693	2	0.658548	11873	12060
2	0.793551	3897	4137	2	0.66967	12061	12181
2	0.76707	4173	4700	8	0.552056	2157	2302
2	0.549905	6205	6270	8	0.542165	8489	8993
2	0.665505	6633	6756	9	0.541522	625	703
2	0.580427	6757	6867	10	0.544302	1485	1558
2	0.610429	8217	8381	10	0.576661	8685	8775

표 5. 가중치별 질의어에 대한 동영상 검색 순위

순위	$\alpha = 0.3 \ \beta = 0.3 \ \gamma = 0.4$		$\alpha = 1.0 \ \beta = 0.0 \ \gamma = 0.0$		$\alpha = 0.0 \ \beta = 1.0 \ \gamma = 0.0$		$\alpha = 0.0 \ \beta = 0.0 \ \gamma = 1.0$	
	동영상ID	순위 점수	동영상ID	순위 점수	동영상ID	순위 점수	동영상ID	순위 점수
1	1	0.92418	1	0.74791	1	1.00000	1	0.99951
2	2	0.45911	2	0.72425	2	0.38799	2	0.31359
3	8	0.19760	10	0.56217	8	0.05575	8	0.04390
4	10	0.17995	8	0.54437	10	0.01400	10	0.01774
5	9	0.16977	9	0.54152	9	0.00670	9	0.01327

고, 평균 동영상의 길이보다 긴 동영상이다. 따라서, 표 5의 결과에서 제안된 측정 기법에 따라 높은 순위 점수로 측정 되었음을 알 수 있다.

5.4 병합에 따른 동영상 순위 변화 성능 평가

본 절에서의 실험은 포스팅 리스트의 병합으로 인한 압축이 인물 검색 시 동영상의 순위에 미치는 영향을 평가한다. 병합 오차율 한도 ϵ_{th} 에 따라 압축된 포스팅 리스트는 동영상의 순위 측정 시에 순위

영향이 주지 않아야 압축에 따른 포스팅 리스트의 정보가 손실 되지 않음을 의미한다. 그림 4는 ϵ_{th} 에 따라 압축된 포스팅 리스트들에 대하여 대통령 별로 동영상의 순위를 평가한 것을 나타낸다. 가로축은 포스팅 리스트가 특정 ϵ_{th} 으로 압축된 것을 나타내고, 세로축은 이에 따른 해당 동영상이 갖는 순위 점수이다. 세로축의 순위 점수는 0부터 1사이의 값을 가지며, 1에 가까울수록 해당 동영상이 해당 인물과 관련도가 높은 영상임을 의미한다. 가로축의 ϵ_{th} 이 0일

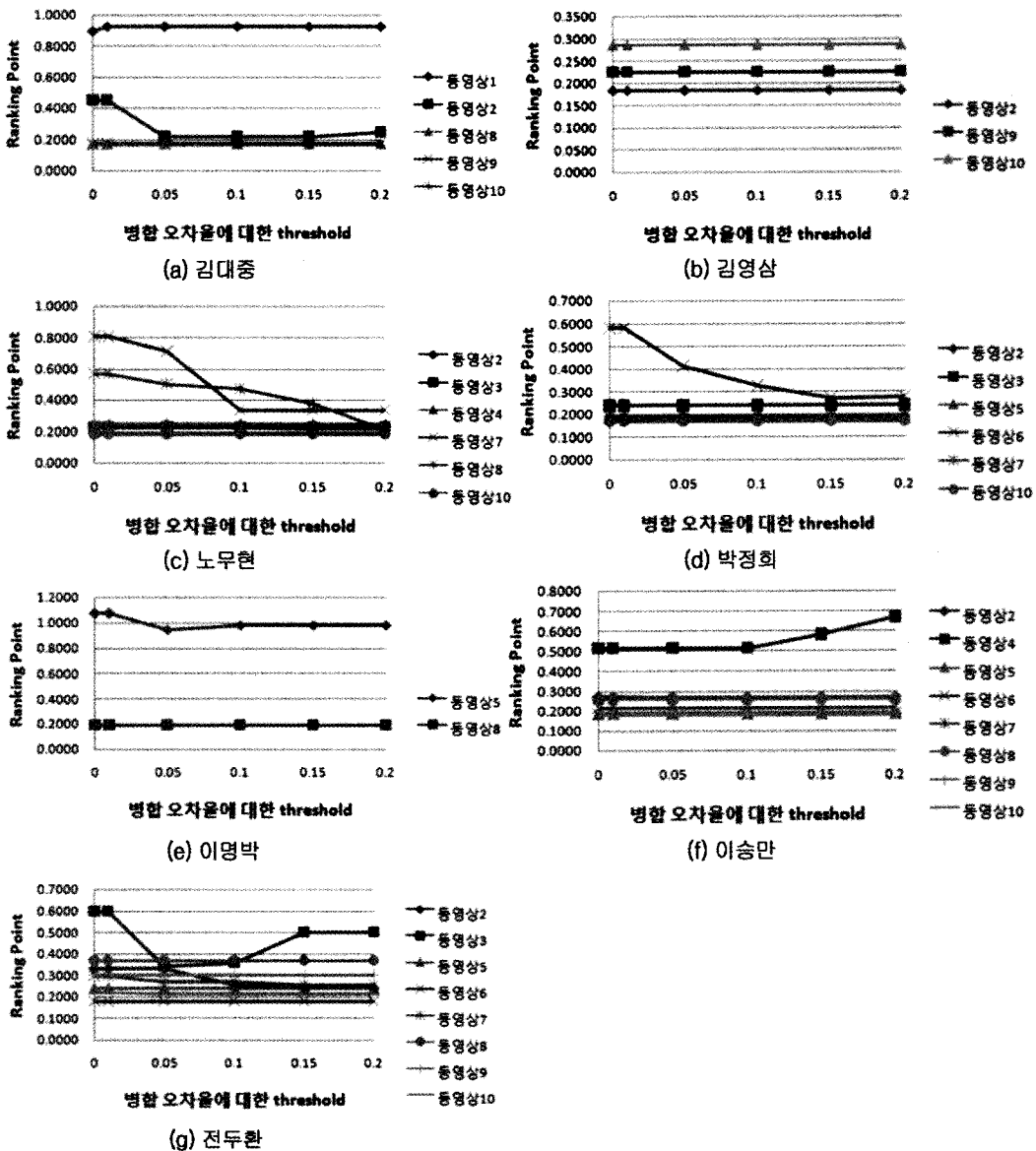


그림 4. 압축된 포스팅 리스트에 대한 인물별 순위 점수 측정

때의 결과는 포스팅 리스트를 압축하지 않았을 경우이다. 예를 들어 그림 4의 (b)는 “김영삼”이 등장하는 동영상들의 순위를 측정된 것으로, 동영상 ID가 2, 9, 10인 동영상의 순위 점수를 측정하였다. 동영상 ID가 10인 영상이 가장 높은 순위 점수로 측정되었고, 동영상 ID가 9인 영상이 그 다음 순위이고, 동영상 ID가 2인 영상이 마지막 순위를 가지게 된다.

포스팅 리스트의 압축은 포스팅 리스트 생성 후, 저장 공간의 크기를 줄이기 위해 행하여진다. 병합 오차율을 높일수록 저장 공간의 크기는 줄어들지만, 과도한 병합은 정보의 손실을 야기한다. 따라서, 저장 공간의 크기를 최소화할 수 있는 병합 오차율을 결정하기 위한 최적의 임계치를 선택해야 한다. 최적의 임계치를 선택하기 위해, 인물별 포스팅 리스트를 생성한 후 압축을 실행할 때, 병합 오차율을 높이면서 압축 전후의 검색 순위와의 변화를 비교하여, 검색 순위가 변하지 않는 최대의 병합 오차율 ϵ_{th} 를 최적의 임계치로 선택한다. 예를 들어, 그림 4의 (a)는 “김대중”에 해당하는 동영상의 순위를 측정된 것이다. 동영상 ID가 1, 2, 8, 9, 10에서 해당 인물의 등장 프레임이 검출 되었고, 병합 오차율을 0.2를 지정하였을 경우에도 순위가 변화하지 않으므로, 해당 키워드의 포스팅 리스트는 병합 오차율 0.2로 압축가능하다. 그림 4의 (b), (d), (e), (f)는 (a)의 경우와 유사하며, 병합 오차율을 0.2로 지정하여 압축이 가능함을 알 수 있다. 하지만, 그림 4의 (c), (g)는 병합 오차율에 따라 순위의 변화가 있음을 확인 할 수 있다. 그래프 (c)의 예에서, “노무현”에 해당하는 포스팅 리스트는 병합 오차율을 0.07로 지정하여 압축하여야 순위의 변동 없이 압축됨을 알 수 있고, (g)의 경우 순위의 변동이 나타나는 0.05 보다 작은 값으로 병합 오차율을 지정해야 함을 알 수 있다.

6. 결 론

본 논문은 동영상 등장인물의 자동 색인을 위한 효율적인 검색과 저장 방법에 관하여 인물들이 등장하는 프레임 구간을 나타내는 포스팅 리스트를 압축하는 기법과 인물에 대한 동영상을 검색할 때, 동영상의 순위를 측정하는 기법에 대하여 제안 하였다. 제안기법은 포스팅 리스트의 프레임 구간에 대하여 해당 구간에서 해당 인물의 정확도와 프레임 구간각

의 거리를 고려하여, 프레임 구간을 병합하는 방법을 사용하였다. 프레임 구간의 병합을 통하여, 프레임 구간에 대한 정보를 줄임으로써, 저장 공간의 크기를 줄였고, 실험을 통하여 병합을 통한 프레임 구간의 압축이 효과적임을 보였다. 또한, 검색시에 동영상의 순위를 측정함에 있어서, 유사도의 평균값과 전체 동영상과 해당 동영상의 길이의 비율, 해당 동영상에서의 질의 인물에 대한 등장 프레임 구간의 비율을 통하여, 순위를 측정하는 방법을 제시하였으며, 실험을 통하여 이 방법이 유효함을 보였다.

제안 기법들은 등장인물의 자동 색인시에 색인의 크기를 효과적으로 개선하고, 검색시에 적절한 검색 결과를 반환함으로써, 제안 기법을 사용하여 동영상 검색에 대해 효과적인 지원이 가능하다.

참 고 문 헌

- [1] 한용구, 김진승, 이승현, 이영구, “iRODS를 이용한 대용량 전자 기록물 관리 시스템,” 한국정보과학회논문지:컴퓨팅의 실제, Vol.16, No.8, pp. 835-836, 2010.
- [2] E. Acosta, L. Torres, A. Albiol, and E. Delp, “An Automatic Face Detection and Recognition System for Video Indexing Applications,” Proceedings of Int’l Conference on Acoustics, Speech and Signal Processing, 2002.
- [3] A. Albiol, L. Torres, C. A. Bouman, and E. J. Delp, “A Simple and Efficient Face Detection Algorithm for Video Database Applications,” Proceedings of the IEEE International Conference on Image Processing, Vol.2, pp. 239-242, 2000.
- [4] 이승호, 최재영, 노용만, “동영상에서 대통령 자동 얼굴색인을 위한 얼굴인식 방법에 관한 연구,” 멀티미디어학회 추계학술대회, 2010.
- [5] 박병권, “정보검색과 데이터베이스 관리 시스템의 밀결합을 위한 역색인 구조와 질의 최적화,” 한국과학기술원 박사학위논문, 1998.
- [6] 류동엽, 임영환, 윤선희, 서정민, 이창훈, 이근수, 이상문, “얼굴 분석과 유사도 비교를 이용한 사용자 인증 시스템,” 한국멀티미디어학회논문지, Vol.8, No.11, pp. 1439-1448, 2005.

- [7] Ian H. Witten, A. Moffat, and Timothy C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, Morgan Kaufmann, 1999.
- [8] O. Vechtomova and M. Karamuftuoglu, "Lexical Cohesion and Term Proximity in Document Ranking," *Information Processing and Management*, Vol.44, No.4, pp. 1485-1502, 2008.
- [9] D. L. Lee, H. Chuang, and K. Seamons, "Document Ranking and the Vector Space Model," *IEEE Transactions on Software*, Vol. 14, No.2, pp. 67-75, 1997.
- [10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.



김진승

1995년 3월~2005년 8월 경희대학교 물리학과 학사, 컴퓨터공학과 학사
 2005년 9월~2007년 8월 경희대학교 컴퓨터공학과 공학석사

2009년 9월~현재 경희대학교 대학원 컴퓨터공학과 박사과정

관심분야: 분산 컴퓨팅, 데이터 마이닝



한용구

1997년 3월~2005년 2월 경희대학교 컴퓨터공학과 학사
 2005년 3월~2007년 2월 경희대학교 컴퓨터공학과 공학석사
 2007년 3월~현재 경희대학교 대학원 컴퓨터공학과 박사과정

관심분야 : 행위인지, 데이터 마이닝, 클라우드 컴퓨팅



이영구

1988년 3월~1992년 2월 한국과학기술원 과학기술대 전산학과 학사
 1992년 3월~1994년 2월 한국과학기술원 전산학과 석사
 1994년 3월~2002년 8월 한국과학기술원 전산학과 박사

2002년 9월~2004년 2월 미국 UIUC 전산학과 Post Doctoral Research Fellow

2004년 3월~현재 경희대학교 컴퓨터공학과 부교수

관심분야 : 대용량 데이터 관리, 데이터 마이닝, 유비쿼터스 데이터관리, 클라우드 컴퓨팅