웨이브렛을 이용한 영상기반 인쇄 한글 단어 검색
(Image-based Retrieval of Printed Korean Words using Wavelets)

김 혜 금 † 양 진 호 †† 이 진 선 ††† 오 일 석 ††††
(Hye-Geum Kim) (Jin-Ho Yang) (Jin-Seon Lee) (Il-Seok Oh)

요 약 내용-기반 문서 검색의 필요성이 급속히 증가하고 있다. 기존의 OCR-기반 텍스트 변환 방법은 명백한 한계를 갖고 있기 때문에 영상-기반 매칭 방법이 대안으로서 인기를 얻고 있다. 새로운 매칭 방법은 빠른 속도와 좋은 검색 성능의 두 가지 요구사항을 충족해야 한다. 이 논문은 웨이브렛의 좋은 특성을 기반으로 개발된 한글 단어에 대한 영상-기반 매칭 알고리즘을 제안한다. 실험은 고품질과 저품질 단어 영상을 가지고 수행하였으며, 실험 결과 제안한 알고리즘이 검색 성능과 속도 면에서 우수함을 확인하였다.

Abstract The need for content-based document retrieval is increasing rapidly. Since the OCR-based text conversion approach has a clear limitation, the image-based matching is gaining a popularity as an alternative. The new matching algorithm should meet two requirements, high speed and good retrieval performance. In this paper, we present an image-based matching algorithm for Korean words which has been developed by using the good properties of wavelets. The experiments have been performed with high-quality and low-quality word images and the experimental results showed the superiority of the proposed algorithm in terms of retrieval performance and speed.

1. 서 론

전통적인 문서 검색 시스템들은 문서의 제목과 요약문으로부터 추출한 몇 개의 키워드들만으로 미리 구축된 색인 구조에 의존하고 있다. 최근 들어 컴퓨터 성능이 급속도로 발전되고 웹이 보편화됨에 따라 문서 검색에 대한 필요성이 빠르게 증가하고 있다. 따라서 문서 검색 사용자가 증가하면서 그들의 요구사항도 증가하고 있는데 이를 충족하기 위해서는, 동적 질의를 통해 임의의 키워드 검색을 가능하게 하는 것과 문서 내의 특정 키워드의 중요성을 측정하기 위해 단어의 빈도 수를 셀 수 있는 것과 같은 새로운 기능이 필요하다. 이러한 요구사항들은 *내용-기반(content-based)* 문서 검색이라고 요약할 수 있다 [Doermann97].

내용-기반 검색을 위한 접근 방법으로는 텍스트 변환(text conversion)을 기반으로 한 방법과 영상-기반(image-based) 방법을 들 수 있다. 먼저 텍스트 변환을 기반으로 한 검색 방법은 스캔된 문서 영상들을 텍스트 파일로 변환하기 위해서 OCR 소프트웨어를 사용한다. 만약 성공적인 텍스트 파일 변환이 가능하다면 이 방법은 다른 어떤 방법보다 뛰어난 최적의 방법이 될 수 있다. 그러나, 현재 OCR에 대한 기술과 소프트웨어의 한계 때문에, 인식된 문자들을 검증하기 위해 많은 노동력이 필요한 후처리를 할 수밖에 없다. 이러한 후처리는 새로운 방법론 개발의 필요성을 대두시켰다.

또 하나의 접근 방법으로는 영상-기반 검색 방법이 있다. 이 방법은 스캔한 영상 자체를 문서 형식 분석 정보와 함께 데이타베이스 안에 저장하는 방법이다. 만약 단어 영상에 대한 빠르고 신뢰할 수 있는 매칭 알고리즘을 가지고 있다고 가정하면, 성공적인 내용-기반 문서

· 이 논문은 과학재단 특정 기초과제(98-0102-02-3)의 지원에 의한 것입니다.
† 비 회 원 : 전북대학교 전산통계학과
hgkim@cs.chonbuk.ac.kr
†† 학생회원 : 전북대학교 전산통계학과
jhyang@cs.chonbuk.ac.kr
††† 정 회 원 : 우석대학교 정보통신컴퓨터공학부 교수
jslee@core.woosuk.ac.kr
†††† 종신회원 : 정보 검색시스템 연구센터
전북대학교 컴퓨터과학과 교수
isoh@moak.chonbuk.ac.kr
논문접수 : 2000년 1월 26일
심사완료 : 2000년 12월 20일

검색 시스템을 구축할 수 있다. 이와 같은 알고리즘은 다음 두 가지 요구사항을 모두 만족해야 한다.

• 디지털 라이브러리 응용과 같은 데이타베이스에는 아주 방대한 양의 단어 영상들이 있기 때문에 매칭 알고리즘이 매우 빨라야 한다.

• 알고리즘은 재현율(recall)과 정확률(precision) 관점에서 신뢰성 높은 성능을 제공해야 한다.

영상-기반 문서 검색 방법에 대한 간략한 사례조사는 [Doermann97]에서 찾아 볼 수 있다. 서양 언어 문서들을 다루는 최근 논문들에서는 상태가 좋지 않은 인쇄 문서들을 위해 pseudo 2-D HMM(Hidden Markov Model)에 기반하여 모델링하고 DP(Dynamic Programming)를 이용해 매칭하는 방법 [Kuo94], 속도와 신뢰도 관점에서 이산과 연속 HMM을 비교하는 기법 [Chen95], 디지털 라이브러리에서 적용되는 기법[Belaid 98], 단순한 모양 구조들을 사용하는 기법[Spitz99], 그리고 영상-기반 매칭과 OCR을 혼합해 접근하는 방법 [Chung99] 등이 있다. Zhu 등은 가설 생성과 검증 기법을 사용하여 중국어 단어 인식을 위한 영상-기반 기법[Zhu97]을 제안했다.

웨이브렛 이론(wavelet theory)은 최근 들어 발전되었지만 이의 응용은 통신, 신호 처리, 멀티미디어, 컴퓨터 그래픽과 비젼, 패턴 인식과 같은 다양한 영역으로 급속히 확대되고 있다 [Stollnitz96]. 패턴 인식 분야에서 성공적인 응용 예는 필기 문자의 인식 [Shioyama 93, Wunsch95, Lee96, Ma97], 문서 이미지 분석 [Tang 97, Hwang98], 웨이브렛 계수로부터 특징 추출 [Pittner 99], 군집 분석 [Murtagh98]등을 주제로 한 논문들에서 찾아 볼 수 있다. 패턴 인식 문제를 해결하기 위해서 웨이브렛을 사용하는 공통적인 동기는 웨이브렛이 다중 해상도(multiresolution) 표현의 여러 가지 이점들을 제공한다는 점에 있다.

문서 검색에 웨이브렛을 응용한 논문은 아직껏 없다. 그러나 웨이브렛을 내용-기반 칼라 영상 검색에 적용하여 좋은 결과를 얻은 논문이 있다 [Jacobs95]. 이 논문은 칼라 영상으로부터 웨이브렛 계수들을 추출하고, 그들을 내림차순으로 정렬한 후, 정렬된 계수로부터 큰 값을 가진 계수를 취하였다. 이 기법은 하르 웨이브렛과 같은 정직교 웨이브렛 기저 함수들(orthonormal wavelet basis functions)중 값이 큰 계수들이 원본 영상에 대한 대부분의 정보를 갖고 있다는 사실에 기반을 두고 있다. 이 논문에서 제안한 영상 질의 측정 방법(image query metric)은 기존 방법보다 성능이 더 우수하다는 것이 판명되었다.

웨이브렛은 다중 해상도 공간에서 신호를 자연스럽게 표현하는데 여러 가지 이점을 갖고 있는 수학적 도구이다. 그리고 이것은 큰 크기를 갖는 계수들 중 단지 몇 개만을 사용해서 원래 영상을 잘 표현할 수 있게 해준다. 그래서 영상 압축 응용에 많이 사용된다 [DeVore92]. 웨이브렛은 디지털 라이브러리와 같이 많은 양의 문서들을 가진 응용 도메인에서, 압축된 계수들의 집합만을 데이타베이스에 저장함으로써 높은 저장공간 효율을 얻을 수 있다. 그리고 웨이브렛은 인식 문제와 연관된 좋은 특성들이 있기 때문에 패턴 인식에서도 매우 유용하다. 압축된 웨이브렛 계수들의 집합이 인식 문제에서 우수한 식별 능력을 가진 특징(features)으로 사용될 수 있다는 사실을 이론적이고 경험적인 근거를 통해 제시한 논문들이 있다 [Jacobs95, Lee96, Pittner99]. 또한 매칭을 압축된 계수들의 집합을 사용하여 수행하기 때문에 매우 빠르게 만들 수 있다.

이 논문에서는 단어 영상 인식을 위한 새로운 접근 방법으로 웨이브렛을 적용하고, 앞에서 언급한 웨이브렛 이점들을 이용하여 성능이 좋은 매칭 알고리즘을 개발하였다. 이 매칭 알고리즘을 한글 단어 영상에 적용하였다. 하지만 한글이 아닌 중국어와 서양 언어에도 이 알고리즘을 약간 수정하여 쉽게 적용할 수 있을 것이다.

이 논문은 문서 페이지 영상이 이미 단어 영상으로 분할되어 있고, 이 단어 영상은 각각의 문자 영상으로 분할되어 있다고 가정한다. 한글의 몇 가지 특징 때문에 단어 대 단어의 매칭보다는 문자 대 문자의 매칭을 선택하였다. 한 단어를 구성하는 각각의 문자들에 대한 매칭 결과를 사용하여 단어에 대한 최종 매칭 결과를 구하였다. 문자 대 문자 매칭은 중국어에 대해서도 적용될 수 있다. 그러나 서양 언어들에 대해서는 단어 대 단어의 매칭이 문자 대 문자의 매칭보다 더 적합할 것이다.

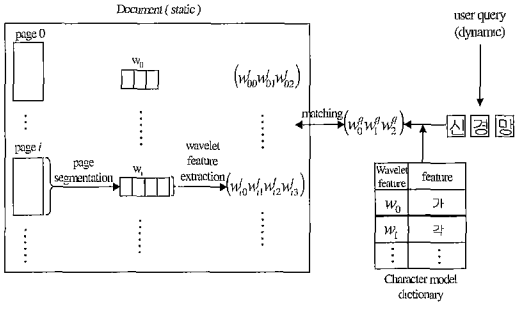
한 개의 문자 영상에 대해 웨이브렛 변환을 수행하여 웨이브렛 계수들을 얻는다. 이들 웨이브렛 계수들을 내림차순으로 정렬하고, 오류율을 넘지 않는 범위에서 원본 문자 영상을 잘 표현할 수 있도록 큰 값을 가진 K개의 계수들을 선택한다. 이렇게 선택된 계수들만 문서 데이타베이스에 저장한다. 앞으로 문서 데이타베이스에 저장된 계수들은 페이지 영상을 복원하기 위한 것뿐만 아니라 질의 단어를 가지고 매칭할 때도 사용된다.

2장에서는 문자 영상의 웨이브렛 변환 과정과 한글 단어 매칭 알고리즘을 설명한다. 실험 결과는 3장에서 기술한다. 4장에서는 응용환경에 대해 기술한다. 마지막으로 5장에서는 이 논문의 결론 및 향후 연구 과제에 대해서 기술한다.

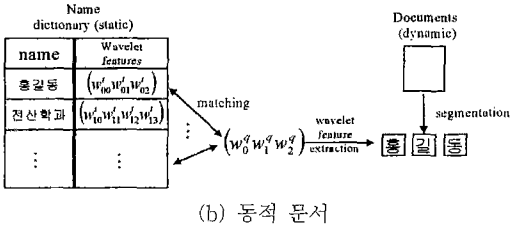
2. 응용 시나리오

이 논문에서 제안한 알고리즘은 효율적인 한글 문서 검색 응용을 염두에 두고 개발하였는데, 많은 문서 검색 응용은 그림 1에 보인 두 가지 시나리오 중 하나로 구분할 수 있다. 그림 1(a)의 시나리오에서는 문서들이 정적(static)이다. 즉, 모든 문서 페이지 영상들은 미리 행, 단어, 문자들로 분할되어 있고 모든 필요한 특징들도 추출되어 데이타베이스에 저장되어 있다. 디지털 라이브러리 응용이 이 시나리오에 속한다. 이 시나리오에서 사용자 질의(user query)는 동적(dynamic)으로 주어진다. 즉, 임의의 키워드(keyword)를 임의의 시간에 시스템에 입력하여 질의할 수 있다. 이러한 질의 단어가 입력됐을 때, 단어를 구성하는 각각의 문자들에 대한 특징 벡터(feature vector)를 구성하기 위해 문자 모델 사전(character model dictionary)을 참조한다. 전형적인 디지털 라이브러리 응용에서 문서 데이타베이스의 단어 수는 수백 만개 이상이다. 따라서 매칭 알고리즘은 매우 빠르게 실행되어야 한다.

다른 시나리오는 팩스 수신자 구별 작업에서 살펴볼 수 있다. 이 시나리오에서는 그림 1(b)에서 볼 수 있는 바와 같이 팩스 수신자 이름들이 웨이브렛 변환을 통해 얻어진 특징 벡터들로 사전에 저장된다. 팩스 문서



(a) 정적 문서



(b) 동적 문서

그림 1 두 가지 검색 시나리오

들이 입력되면 입력된 문서에서 수신자 이름 영역이 분할되고 분할된 부분 영상으로부터 특징 벡터를 추출한다. 이 시나리오에서는 문서들이 동적으로 입력되고, 이름 사전(name dictionary)이 정적이다.

두 가지 시나리오에서 시스템은, 페이지 형식 분석을 통한 단어/문자 분할과 빠르고 신뢰성 높은 매칭을 통한 적절한 단어 검색의 두 가지 기능을 제공해야 한다. 이 논문은 이 중 두 번째 기능을 위해 높은 성능의 단어 검색 방법을 제안한다. 제안한 방법에서 데이타베이스는 방대한 양의 목표 단어들(target words)로 구성되어 있기 때문에 매칭 알고리즘은 어느 정도 높은 신뢰성을 유지하면서 빠른 속도로 수행되어야 한다.

3. 웨이브렛에 기반한 단어 매칭

3.1 개 요

그림 2는 몇 개의 한글 문장 영상을 보여준다. 한글에서 조사(즉, '인공지능은'에서 '은')는 공백 없이 명사 뒤에 붙기 때문에 질의 단어에 해당하는 명사만 떼어내는 작업이 쉽지 않다. 또한 띄어쓰기 규칙이 엄격하게 정해져 있지 않기 때문에 또 다른 문제가 발생한다. 또한 문장 영상에서 인접한 문자들 사이의 간격(즉, 장평)이 일정하지 않기 때문에 단어 대 단어 매칭은 높은 오류율이라는 문제를 안고 있다.

그래서 이 논문에서는 단어 대 단어 매칭 방법 대신 문자 대 문자 매칭 방법을 선택했다. 한글 문자들은 일정한 넓이와 높이를 갖고 있기 때문에 단어 영상에서 각각의 문자들을 분할하는 작업은 어렵지 않다.

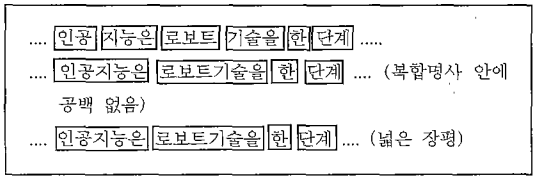


그림 2 한글 문장 영상들

검색 시스템에서 일반적인 사용자 질의는 몇 개의 명사나 합성명사를 포함하는 구(phrase)나 여러 개의 단어로 구성되는 절과 같은 임의의 형태를 취할 수 있다. 이러한 시스템에서 검색 문제는 문서 데이타베이스를 구성하는 모든 단어들을 일렬로 연결시킴으로서 만들어지는 긴 목표 문자열(target string) 중 부분 문자열(substring)을 찾는 것으로 생각할 수 있다. 다음 pseudo 코드는 이러한 상황에서의 검색 알고리즘이다.

```
Substring searching:
  Target string = (C0^t, C1^t, ..., Ci^t, Ci+1^t, ..., CN-1^t) (N = 수백만개 또는 그이상)
  Query string = (C0^q, C1^q, ... ,CK-1^q) (K = 보통으로 10미만)
procedure Word_retrieval ;
Input: target and query strings
Output: list of locations in target string where query string is successfully matched
{
  for( i=0 ; i<=N-K ; i++ ) {
    success = matching_successful(C0^q, Ci^t) AND
              matching_successful(C1^q, Ci+1^t) AND
              .....
    matching_successful(CK-1^q, Ci+K-1^t); ---------- 식 (3.1)
      if(success if TRUE) put the location i into the list;
  }
}
```

위에서, W는 한 문자에 대한 특징 벡터(feature vector)이다. 프로시져 *Word_retrieval*에서 matching_successful( ) 함수는 매칭이 성공적인지 아닌지를 가리키는 Boolean 값을 넘겨준다. 식(3.1)을 대치하는 다른 방법은 K개 문자의 매칭 점수들에 대한 평균을 사용하는 것이다.

3.2 1차원 하르 웨이브렛 변환

3.2.1 웨이브렛 분해(wavelet decomposition)

웨이브렛 변환이 어떻게 계산되는지 알기 위해 간단한 예를 들어 설명하고자 한다. 아래 그림 3에서와 같이 4개의 원소, 즉 [9 8 2 6]으로 된 1차원 배열이 있다고 하자. 배열의 개수를 '해상도(resolution)'라고 표현한다. 즉, 이 배열의 해상도는 4이다. 이 배열에서 인접한 한 쌍씩 평균을 구해가면서 해상도가 낮은 배열을 생성한다. 그러면 해상도는 2로 낮아지고 [8.5 4]의 평균값을 갖는 배열을 구할 수 있다.

이렇게 하면 몇몇 정보가 손실되는데, 해상도가 4인 원래의 배열로 복원할 수 있도록 잃어버린 값을 얻어낼 수 있는 몇 개의 계수를 추가할 수 있다. 변환 후 얻어진 [8.5 4] 중, '8.5'라는 값은 변환 전 9와 8의 평균이다. 그렇다면 9는 평균인 8.5보다 0.5크고 8은 평균보다 0.5 작다는 사실을 알 수 있다. 마찬가지로 2와 6의 평균인 '4'를 고려해보면, 2는 평균인 4보다 -2크고 6은 평균보다 -2 작게 된다. 이렇게 얻어진 0.5와 -2라는 값들을 바로 '세부계수(detail coefficient)라고 부른다. 이와 같은 방법으로 해상도 4인 배열을 해상도 2인 배열로 웨이브렛 분해한 결과는 [8.5 4 0.5 -2]와 같다. 다시 한 번 해상도를 낮추면 8.5와 4의 평균인 6.25와 세부계수 2.25를 구할 수 있다.

위와 같이 배열을 평균값과 세부계수들로 표현하는 방법을 '웨이브렛 변환(wavelet transform)' 또는 '웨이브렛 분해(wavelet decomposition)' 라 한다.

그림 3은 하르 웨이브렛 분해하는 과정이다.

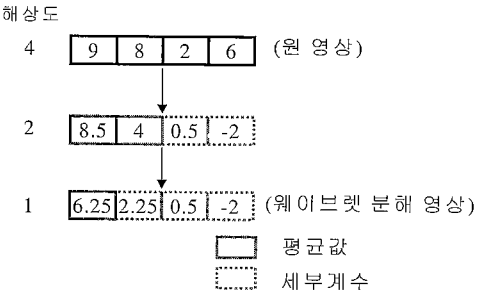


그림 3 웨이브렛 분해의 예

3.2.2 웨이브렛 복원(wavelet reconstruction)

앞 절에서 살펴본 바와 같이 웨이브렛 분해의 결과는 원 영상을 표현하는 하나의 평균과 나머지 즉, 세 개의 세부 계수들로 구성된다. 이것들을 가지고 역으로 원 영상의 값을 찾아가는 것을 '웨이브렛 복원(wavelet reconstruction)'이라 한다.

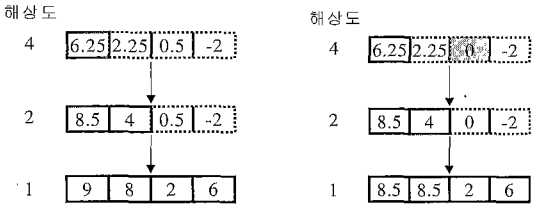
앞 절에서 살펴본 [9 8 2 6]의 4개 원소로 이루어진 배열을 가지고 복원하는 방법을 살펴보자. [9 8 2 6]을 웨이브렛 분해한 결과는 [6.25 2.25 0.5 -2]이다. 먼저 해상도가 2인 단계로 복원하기 위해서 [6.25 2.25]를 이용한다. 평균값인 6.25에 세부계수 2.25를 더하면 8.5가 되고, 평균값 6.25에서 세부계수 2.25를 빼면 4가 되므로 복원한 결과는 [8.5 4 0.5 -2]와 같다. 이 결과는 해상도 2일 때의 웨이브렛 분해 결과와 같다. [8.5 4 0.5 -2]를 가지고 다시 한번 복원을 수행하면 평균값 8.5에 세부계수 0.5를 더한 값 9와 평균값 8.5에서 세부계수 0.5를 뺀 값 8이 되고 마찬가지 방법으로 평균값 4와 세부계수 -2를 가지고 복원을 수행하면 2와 6이 된다. 따라서 해상도가 4인 원래의 배열로 복원한 결과는 [9 8 2 6]과 같다.

그림 4(a)는 정보 손실이 전혀 없이 복원하는 과정의 예이다.

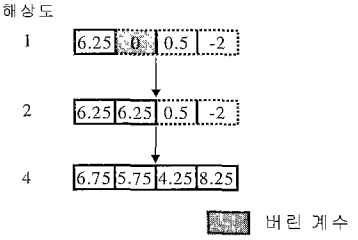
그림 4(b)는 웨이브렛 복원을 할 때 어느 정도 작은 값을 갖는 계수를 무시하고 복원하는 경우의 예이다. [6.25 2.25 0.5 -2]의 원소 중 절대값이 가장 작은 값인 0.5를 무시하고 복원할 경우 결과는 [8.5 8.5 2 6]이다. 이것은 무손실 복원 결과인 [9 8 2 6]과 거의 비슷한 값을 갖는다고 말할 수 있다. 즉, 이것은 아주 작은 값을 갖는 계수들을 무시하고 복원해도 원 영상과 유사한 영상을 얻을 수 있다는 말이다.

그림 4(c)는 [6.25 2.25 0.5 -2]의 원소 중 어느 정도 큰 값인 2.25를 무시하고 복원한 경우의 예이다. 복원 결과는 [6.75 5.75 4.25 8.25]이며 이것은 무손실 복원 결과인 [9 8 2 6]과 아주 큰 차이를 보이고 있다.

웨이브렛 분해한 결과 얻어진 계수 값들은 값이 클수록 원 영상에 대한 정보를 많이 갖고 있으므로 어느 정도 작은 값을 갖는 계수들을 무시하고 웨이브렛 복원을 수행해도 원 영상에 유사하게 접근한다는 사실을 알 수 있다. 이 논문에서는 이러한 웨이브렛의 특성을 이용해 단어 영상 매칭에 사용하고 있다.



(a) 무손실 복원 (b) 작은 계수 버림(적은 손실)



(c) 큰 계수 버림(큰 손실)

그림 4 웨이브렛 복원의 예

3.3 문자 영상의 웨이브렛 변환

하르 웨이브렛(Haar wavelet)은 0과 1 사이에서는 1, 그 이외에서는 0 값으로 이루어진 간결토대(compact support)를 갖는 단순한 상자형 함수(box function) $\varphi(X)$를 사용하는 가장 간단한 방법이다. 하르 웨이브렛은 정직교성(orthonormality)과 빠른 계산 능력 등의 좋은 특성들 때문에 영상 압축 [DeVore92]과 영상 검색 [Jacobs95]에 널리 사용되어 왔다. [Stollnitz96]의 2-3장은 웨이브렛 분해의 계산과정과 예가 잘 나타나 있으므로, 하르 웨이브렛에 관한 수학을 위해서 참조하기 바란다.

아래에 기술한 *Haar_wavelet_decomposition* 프로시져를 호출하여 하르 웨이브렛 분해를 수행할 수 있다. 이 프로시져는 웨이브렛 계수들을 저장하기 위해 배열 W를 사용하고 원본 영상을 저장하기 위해 배열 I를 사용한다. C언어와 유사한 형태로 의사 코드(pseudo code)를 작성하였다. I와 W는 동일한 크기의 배열이며, 모든 연산은 W상에서 in-place로 이루어진다.

```
Procedure Haar_wavelet_decomposition;
Input: I[2^n] [2^n]    /* original 2-dimensional image */
Output: W[2^n] [2^n]   /* resulting wavelet coefficients */
{
copy I to W;
   /* all the operations below are done on W in-place */

/* normalize for orthonormality */
      for (each pixel in W at (row,col))
            W[row][col] = W[row][col]/2^n;

for (j=n; j>=1; j--) {
length = 2^j;
 /* decomposition along rows */
for (row=0; row<length; row++) {
   for (i=0; i<length/2; i++) {
     T[i] = (W[row][2i]+W[row][2i+1])/2^1/2;
     T[length/2+i] = (W[row][2i]-W[row][2i+1])/2^1/2;
   }
   copy T[.] to W[row][.];
}
/* decomposition along columns */
for (col=0; col<length; col++) {
   for (i=0; i<length/2; i++) {
    T[i] = (W[2i][col]+W[2i+1][col])/2^1/2;
    T[length/2+i] = (W[2i][col]-W[2i+1][col])/2^1/2;
   }
   copy T[.] to W[.][col];
}
}
}
```

프로시져 *Haar_wavelet_decomposition*의 실행이 끝나면 W[0][0]에 저장되고 α(즉, 입력 영상의 전체 평균)로 표기되는 한 개의 스케일 함수(scaling function) 계수와 W의 나머지 원소들에 저장되며 ω_1(즉, 세부 정보)로 표기되는 $2^{2n}-1$개의 웨이브렛 함수(wavelet function) 계수들을 얻는다. 이렇게 얻은 계수들은 *웨이브렛 계수 열(wavelet coefficient list)* $L = (\alpha, \omega_1, \omega_2, \ldots\ldots, \omega_N)$, $N = 2^{2n}-1$ 로 표기할 수 있다.

영상 압축에 웨이브렛을 사용하는 근거는 크기가 큰 웨이브렛 계수들이 원본 영상 I의 대부분 정보를 포함하고 있다는 사실에 기반을 두고 있다 [DeVore92]. 다시 말해서, 만약 정직교 웨이브렛 기저 함수들(orthonormal wavelet basis functions)을 사용한다면, 크기가 L인 계수를 제거하는 것이 L보다 작은 다른 어떤 계수를 제거하는 것보다 항상 더 큰 정보 손실을 가져온다. 이러한

사실에 기반하여 이 논문에서는 L의 계수들 ω_i를 정렬하고 잘 정렬된 계수열로부터 큰 값을 갖는 계수를 선택하는 방법을 사용하여 정보 손실을 최소로 줄이고 있다. 위의 프로시져 Haar_wavelet_decomposition에서 정직교성을 확보하기 위해 영상의 화소값을 2^n으로 나누는 정규화 과정이 포함되어 있다[Stollnitz96]. 정렬 계수열(*ordered coefficient list*)과 압축 계수열(*compressed coefficient list*)은 $L_{ordered}$와 $L_{compressed}$로 표기한다. 즉, 아래와 같이 정의 할 수 있다.

$L = (\alpha, \omega_1, \omega_2, ..., \omega_N)$, $N = 2^{2n}-1$.

$L_{ordered} = (\alpha, \omega_{\pi(1)}, \omega_{\pi(2)}, ..., \omega_{\pi(N)})$, $\pi(i)$는 내림차순으로 정렬된 색인들을 가리킴.

$L_{compressed} = (\alpha, \omega_{\pi(1)}, \omega_{\pi(2)}, ..., \omega_{\pi(K)})$, K는 압축 비율을 결정함.

Jacobs 등은 내용-기반 칼라 영상 검색에 대한 논문에서 위에서 살펴본 하르 웨이브렛 기저 함수들의 특성을 사용했다 [Jacobs95]. 이 논문에서는 목적 영상(target image)과 질의 영상(query image)으로부터 추출한 두 개의 압축 계수열의 매칭에 적합한 매칭함수를 개발하였다. 압축 계수열은 128*128 영상에 대해 내림차순으로 정렬된 계수열(즉, 16,384개의 계수를 가짐)에서 큰 값을 가진 계수 60개(약 0.37%정도)를 선택하여 구성한다. 이것은 허용할 수 있는 범위 내의 정보 손실로서, 방대한 정도의 데이타 축소이다. 이 논문은 실험 결과를 근거로 하여, 이러한 압축 계수열은 원본 영상에서 너무 세부적인 내용을 나타내는 계수들을 적당히 무시하기 때문에 좋은 식별 능력을 제공한다고 주장하고 있다. 이런 주장이 모든 경우에 적용된다고 말할 수는 없지만 적절한 경우에 대해 이런 식별 능력에 관한 좋은 특성을 사용할 수 있다. 프로시져 *Haar_wavelet_reconstruction*은 위의 프로시저와 반대 역할을 수행한다. 이 프로시져는 원본 영상 I를 복원하기 위해 W를 사용한다. W의 모든 계수들을 사용한다면 원본 영상 I를 정보 손실없이 복원할 수 있다.

```
Procedure Haar_wavelet_reconstruction;
Input: W[2^n] [2^n]   /* wavelet coefficients */
Output: I[2^n] [2^n]   /* original 2-dimensional image */
{
copy W to I;   /* all the operations below are done on I
in-place */

for (j=1; j<=n; j++) {
length = 2^j;
/* reconstruction along columns */
for (col=0; col<length; col++) {
   for (i=0; i<length/2; i++) {
   T[2i] = (I[i][col]+I[length/2+i][col])/2^1/2;
   T[2i+1] = (I[i][col]-I[length/2+i][col])/2^1/2;
   }
   copy T[.] to I[.][col];
}

/* reconstruction along rows */
for (row=0; row<length; row++) {
   for (i=0; i<length/2; i++) {
   T[2i] = (I[row][i]+I[row][length/2+i])/2^1/2;
   T[2i+1] = (I[row][i]-I[row][length/2+i])/2^1/2;
   }
   copy T[.] to I[row][.];
}
}
      /* undo the normalization */
for (each pixel in I at (row,col)) I[row][col] = 2^n*I[row][
col];
}
```

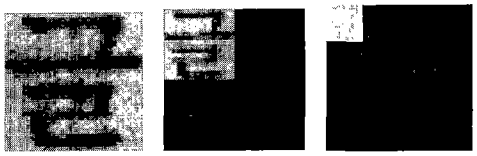
그림 5는 한글 문자 영상 ($I[2^5][2^5]$로 크기정규화된 패턴)과 프로시져 Haar_wavelet_decomposition을 이용한 영상의 웨이브렛 분해 과정을 보여주고 있다. 윗줄 왼쪽 영상은 원본 문자 영상(256단계 명암 영상), 나머지 두 개와 다음 줄의 왼쪽 두 개의 영상은 중간 결과, 그리고 두 번째 줄의 가장 오른쪽 영상은 최종적으로 분해된 결과 (즉, $W[2^5][2^5]$)를 보여준다. 계수열의 길이, 즉 $L_{compressed}$의 K가 복원된 영상의 질에 미치는 영향을 살펴보기 위해서, 다양한 K값을 가지고 프로시져 Haar_wavelet_reconstruction을 적용한다. 그림 6은 K = 50, 100, 150, 200, 250, 300의 압축계수열을 가지고 복원한 영상을 보여준다. 50개 정도의 압축계수열을 사용해서 복원한 영상도 문자 모양의 형태를 어느 정도는 유지하고 있음을 알 수 있다. 이와 같은 실험을 통해 1,024개의 계수들 중 큰 값을 가진 150개 정도의 계수들만을 사용해서 원본 문자 영상을 어느 정도 복원할 수 있다는 것을 알게 되었다.

이런 특성을 이용해서, 개개 문자 영상에 대해 전체 영상 대신 압축계수열만 문서 데이타베이스에 저장한다. 어떤 단어가 검색되었으면 해당 페이지에 속한 문자들의 압축계수열을 사용해서 원본 영상과 유사한 페이지 영상을 복원하고 사용자에게 복원된 페이지 영상을 제공한다.

원본 영상 단계 1 단계 2

단계 3 단계 4 단계 5(최종결과)

(a) '한'자의 웨이브렛 변환

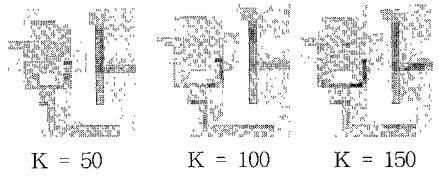


원본 영상 단계 1 단계 2

단계 3 단계 4 단계 5(최종결과)

(b) '글'자의 웨이브렛 변환

그림 5 한글 문자 영상과 그들의 웨이브렛 변환



K = 50 K = 100 K = 150

K = 200 K = 250 K = 300

(a) '한'자의 복원

K = 50 K = 100 K = 150

K = 200 K = 250 K = 300

(b) '글'자의 복원

그림 6 압축계수열을 가지고 복원

3.4 매칭 알고리즘

우리의 매칭 알고리즘은 문자 대 문자로 수행하므로, 목적 단어(target word)와 질의 단어(query word)에 속하는 두 개의 문자 영상을 대상으로 생각하면 된다. 목적 문자(target character)는 그것의 압축 계수열 $L^t_{compressed}$에 의해 표현되고, 질의 문자(query character)는 그것의 모든 웨이브렛 계수를 갖는 웨이브렛 계수 목록 L^q로 표현된다. $L^t_{compressed}$와 L^q는 다음과 같이 쓸 수 있다.

$L^t_{compressed} = (\alpha^t, \omega^t_{\pi(1)}, \omega^t_{\pi(2)}, ..., \omega^t_{\pi(K)})$ 와

$L^q = (\alpha^q, \omega^q_1, \omega^q_2, ..., \omega^q_N)$, $N = 2^{2n}-1$.

우리는 $L^t_{compressed}$와 L^q사이의 매칭 점수를 계산하기 위해 아래의 공식을 사용한다.

$MS(L^t_{compressed}, L^q) = \Sigma_{i=1,K}|\omega^t_{\pi(i)} - \omega^q_{\pi(i)}| + h*|\alpha^t - \alpha^q|$ ------ 식 3.2

$MS(L^t_{compressed}, L^q)$는 동일한 색인을 갖는 웨이브렛 계수 $\omega^t_{\pi(i)}$와 $\omega^q_{\pi(i)}$의 차를 계산하고, 모든 차의 합계를 계산한다. 또한 α값은 대부분 ω값보다 크기 때문에 가중치 계수 h를 α^t와 α^q의 차에 곱한다. 논문에서는 h값으로 0.8을 사용하였다. 이렇게 구해진 두 수를 더함으로써 최종 결과 값을 얻는데, 이 값이 작을 수록 두 문자 영상이 잘 매칭 된다는 것을 의미한다.

위에 제시한 문자 대 문자 매칭 공식을 이용하여 두 개의 단어 영상을 매칭하기 위한 알고리즘을 제시한다. 이 알고리즘은 두 개의 단어 영상이 (C_0^t, C_1^t, ... , C_{k-1}^t)와 (C_0^q, C_1^q, ... , C_{k-1}^q)에 의해 표기되었을 때, C_i^t에서 $L^t_{compressed}$, C_i^q에서 L^q를 얻은 후 이를 이용하여 식 3.2의 $MS(L^t_{compressed}, L^q)$식을 계산하여 k개 문자에 대한 매칭 점수를 계산한다. 만일 K개의 문자 단위 매칭 점수가 다음 조건을 만족하면 두 개의 단어가 성공적으로 매칭되었다고 출력한다.

단어 매칭 조건:

C_i^t와 C_i^q , $0 \le i < k$, 사이의 매칭 점수가 T_1보다 작고, k개 매칭 점수들의 평균이 T_2 보다 작다.

4. 실험 결과

4.1 실험 환경

문자 모델 사전을 구축하기 위해서 한글 워드 프로세서, 흔글™을 사용하여 2,350개의 한글 문자 영상을 화면상에 생성하고, 영상 편집 소프트웨어를 사용하여 문자 영상을 편집하였다. 이들 문자에 대한 원본 영상은 256단계 명암 영상이며, 이를 32*32로 크기-정규화 하였다. 이런 문자 영상들에 2장에서 소개한 프로시져 Haar_wavelet_decomposition을 적용하여 웨이브렛 계수

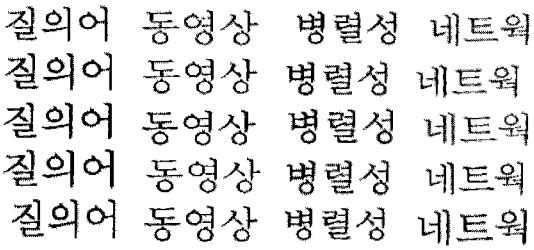
열 L을 구성한 후 이를 문자 모델 사전에 저장하였다.

제안한 단어 매칭 알고리즘의 성능을 테스트하기 위해 두 종류의 실험을 수행하였다. 첫 번째 실험은 우리가 자체 수집한 단어 영상을 사용하였고, 두 번째 실험은 전남대학교에서 구축한 한글 단어 영상 데이타베이스를 사용하였다.

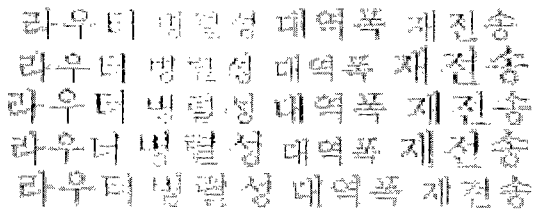
첫 번째 실험 (실험 1)에서는 세 종류의 단어 영상을 수집하여 각각에 대해 실험하였다. 수집한 단어 영상은 모두 컴퓨터 과학에서 주로 사용하는 세 개의 문자로 구성된 키워드를 선택했다. 첫 번째 실험 자료(실험자료 1)는 최근 발행된 *한국 정보 과학회 논문지 (Journal of Korean Information Science Society)*를 300dpi로 스캔하여 얻은 페이지 영상으로부터 100종류의 키워드를 선택하고, 하나의 키워드에 대해 15개의 단어 영상, 즉 총 1,500개의 단어 영상을 수집하여 구축하였다. 수집한 단어 영상의 폰트는 신명조체이며 폰트 사이즈는 9이다. 그림 7은 실험에 사용한 100종류의 키워드들을 열거하고 있다. 두 번째 실험 자료(실험자료 2)는 제안한 알고리즘의 질이 떨어지는 문서에 대한 성능을 테스트하기 위한 것으로서, 실험 자료 1에서 사용했던 것과 동일한 논문지를 150dpi의 저해상도로 스캔하여 얻은 페이지 영상으로부터 수집했다. 세 번째 실험 자료(실험자료 3)는 문서의 질이 좋지 않은 팩스 문서를 고해상도로 스캔하여 얻은 문서 영상으로부터 구축하였다. 그림 8은 세 종류의 실험 자료에 포함되어 있는 단어 영상의 예제이다.

질의어 유효성 페이지 엔트리 일관성 이벤트 세선화 재전송 유전자 태스크
이용률 라우터 디스크 실시간 동영상 서비스 사이클 논리적 대역폭 복잡도
메시지 가시화 렌더링 동기화 스트림 프레임 세분화 결합화 클래스 시스템
데이타 회로망 메모리 수집기 분할점 키워드 스레드 그래프 추상화 구현부
페이스 재스쳐 경계선 명령어 스칼라 윈도우 병렬성 비순차 트래픽 볼러킹
추적법 난반사 선인출 윤곽선 송신자 수신자 부호화 블록화 단조성 다면체
메쏘드 라우팅 입출력 동의어 플러쉬 모듈라 유니온 테스트 연산자 교통량
교차로 가중치 유사도 화살표 엑세스 무효화 게실행 카운터 사용자 관리자
미디어 매크로 네트웍 선택률 무결성 관계형 모호성 색인어 피드백 결합도
필터링 신경망 스키마 이미지 인터넷 단말기 비디오 오디오 한국어 주파수

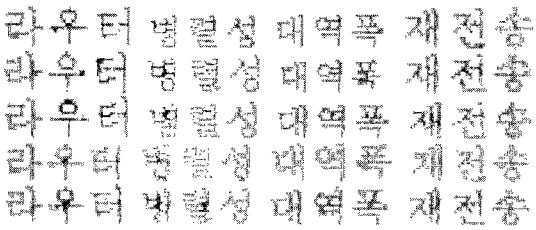
그림 7 실험 1에 사용한 100개의 키워드



(a) 실험자료 1 (300dpi)



(b) 실험자료 2 (150dpi)



(c) 실험자료 3 (fax 문서 300dpi)

그림 8 실험 1에 사용한 단어영상 예제들

두 번째 실험 (실험 2)이 사용한 실험 자료를 표 1이 보여주고 있다. 이 데이타베이스는 전남대에서 구축하였으며 고딕체와 명조체 각각에 대해 bold와 regular 속성을 가지고 있다. 또한 폰트 크기도 10-14 points로 다양하다. 그림 9은 이 데이타베이스의 단어 영상 예제를 보여준다.

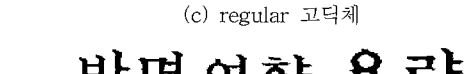
표 1 실험 2가 사용한 단어 영상 데이타 베이스

폰트 종류	속성	단어 영상 갯수
고딕체	bold	1,175
	regular	1,115
명조체	bold	1,183
	regular	1,098

(a) regular 명조체

(b) bold 명조체

(c) regular 고딕체



(d) bold 고딕체

그림 9 실험 2가 사용한 단어 영상 예제들

표 2 실험 1의 검색 성능 표

K	실험자료 1			실험자료 2			실험자료 3		
	Recall ratio(%)	Precision ratio(%)	Speed	Recall ratio(%)	Precision ratio(%)	Speed	Recall ratio(%)	Precision ratio(%)	Speed
10	86.67	96.58	37,500	74.67	91.06	37,500	75.67	92.28	37,500
20	93.33	93.96	18,750	77.34	89.23	18,750	86.67	93.53	18,750
30	94.60	95.88	12,500	79.00	91.15	12,500	80.34	94.51	12,500
40	93.73	96.90	9,375	76.34	91.60	9,375	88.67	94.66	9,375
50	92.93	97.55	7,500	76.00	93.06	7,500	89.67	92.76	7,500
60	95.27	97.74	6,250	78.00	91.41	6,250	79.67	94.84	6,250
70	89.47	98.75	5,357	80.34	88.93	5,357	86.00	93.82	5,357
80	89.73	98.39	4,688	79.34	89.82	4,688	83.67	92.96	4,688
90	95.13	97.47	4,166	80.34	91.98	4,166	86.67	95.24	4,166
100	93.47	98.23	3,750	80.34	91.98	3,750	81.67	94.23	3,750
110	94.27	96.65	3,409	79.34	93.47	3,409	83.00	93.26	3,409
120	91.60	98.42	3,125	82.00	92.48	3,125	87.34	94.58	3,125

제안한 단어 매칭 알고리즘의 성능을 테스트하기 위하여 주어진 키워드 각각에 대해 모든 단어 영상에 일대 일로 매칭을 시도하였다. 검색 성능 측정을 위해 아래와 같이 정의되는 재현율(recall)과 정확률(precision)을 사용하였다.

재현율=(검색된 적합 단어의 수)/(적합 단어의 총 수)

정확률=(검색된 적합 단어의 수)/(검색된 단어의 총 수)

즉, 재현율은 문서 내에 존재하는 적합한 단어 중에서 실제 검색한 결과 나타나는 적합한 단어의 출현 빈도가 얼마나 높은가의 관점에서 검색 성능을 측정하는 것이고 정확률은 실제 검색한 결과 나타나는 모든 단어(적합, 부적합 단어 모두 포함) 중에서 적합한 단어를 찾은 비율이 얼마나 높은가의 관점에서 검색 성능을 측정하는 것이다.

32*32로 크기정규화된 영상을 웨이브렛 변환하면, 총 1,024(=32*32)개의 웨이브렛 계수를 얻는다. 실험은 총 1,024개의 웨이브렛 계수를 갖는 $L_{ordered}$에서 K개만 선택한 $L_{compressed}$를 구성하여 사용하는데 K를 10부터 120까지 10씩 증가시키며 재현율과 정확률을 측정하였고, 매칭 속도 측정을 위해 초당 매칭하는 단어 수를 계산하였다.

비교 대상으로는 유클리드 거리 매칭 알고리즘을 사용하였다. 이 알고리즘은 원래 문자 영상에 대해 아래와 같은 값을 계산한다.

$$DIST\ (I_t, I_q) = \sqrt{\sum_{i=1,2^n} \sum_{j=1,2^n} (I_t[i][j] - I_q[i][j])^2}$$

$I_t[2^n][2^n]$: 목적 문자 영상

$I_q[2^n][2^n]$: 질의 문자 영상

4.2 실험 1

표 2와 그림 10은 실험 1에서 사용한 세 가지 실험 자료에 대한 검색 성능을 보여준다. 실험 자료 1의 경우 10개에서 120개의 계수를 사용한 모든 경우에서 아주 큰 폭으로 정확률과 재현율이 떨어지는 경우는 없었다. 60개의 계수(즉, 전체 계수의 5.9%)를 사용했을 때는 재현율과 정확률이 모두 95%이상이며, 계수의 수를 늘려도 검색 성능은 거의 향상되지 않았다. 이것은 전체 계수 1,024개를 사용했을 때 보다 연산량이 1/16정도로 줄어든 것이므로, 검색 성능의 저하 없이 연산 시간을 1/16정도로 줄일 수 있음을 의미한다. 또한 20개의 계수만을 사용한 결과를 보면 정확률과 재현율 모두 93%이상의 수준을 유지하는데, 이것은 최적의 성능은 아니지만 속도가 중요한 상황에서는 사용할 수도 있는 정도이다. 20개의 계수만을 사용한다면 전체 계수 1,024개를 모두 사용한 연산량의 1/50로 줄어들 것이고, 따라서 연산 시간도 1/50로 줄일 수 있다. K=60개의 계수를 사용

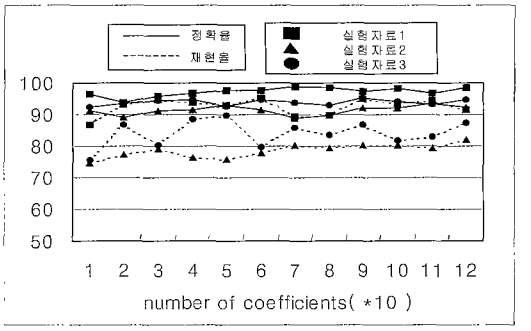


그림 10 실험 1의 검색 성능 그래프

하는 경우 초당 6,250개의 단어 영상을 처리할 수 있다. 한 페이지는 약 500개 정도의 단어들로 구성되어 있으므로, 이는 초당 12.5페이지를 처리할 수 있다고 말할 수 있다. k=20개를 사용하면 초당 18,750개의 단어를 처리할 수 있다. 이것은 초당 37.5페이지 처리량에 해당한다.

실험 자료 2의 경우 단어 영상의 질이 떨어지기 때문에 실험 자료 1보다는 좋지 않지만 10개에서 120개의 계수를 사용한 모든 경우에서 아주 큰 폭으로 정확률과 재현률이 떨어지는 경우는 없었다. 90개의 계수(즉, 전체 계수의 8.8%)를 사용했을 때 재현율과 정확률이 어느 정도의 수준을 유지하고 있다는 것을 알 수 있다.

실험 자료 3은 문서의 질이 좋지 않은 팩스 문서를 대상으로 한 실험이다. 즉, 실험 자료 1과 실험 자료 2에서 사용한 페이지 영상보다 노이즈가 매우 심한 경우의 문서 영상을 사용한 실험이다. 노이즈로는 긴 가로획, 세로획에서 중간에 끊어지는 것, 노이즈로 인해 획들의 구분이 없이 뭉쳐버린 것, 획들 사이에 노이즈가 생겨서 실제 존재하지 않는 자음 또는 모음이 되는 것, 획들에 계단 현상이 발생하는 것 등을 들 수 있다. 단어 영상의 질이 나쁘기 때문에 제안한 매칭 방법이 매우 좋은 성능을 나타내지는 못한다. 하지만 K를 10부터 120까지 10씩 증가시키면서 측정한 재현율과 정확률이 큰 폭으로 떨어지는 경우는 없었다. 90개의 계수(즉, 전체 계수의 8.8%)를 사용했을 때 가장 좋은 성능을 나타내고 있으며, 40~50개의 계수(즉, 전체 계수의 3.9~4.9%)를 사용했을 때도 유사한 성능을 나타내고 있다.

성능 비교를 위해 동일한 데이타를 이용하여 유클리드 거리 방법으로 매칭을 수행하였다. 표 3은 두 가지 매칭 방법을 비교 분석한 결과이다. 실험자료 1의 경우 정확률과 재현율 측면에서 살펴보면 제안한 방법이 약 1%정도 우수하며, 매칭 속도 면에서도 16배정도 빠르다는 것을 알 수 있다. 실험 자료 2에서는 제안한 방법이 유클리드 거리 방법보다 재현율과 정확률 면에서 약 3%정도 떨어지는 결과를 보이고 있으나 속도 면에서는 약 9배 정도 빠르다는 것을 알 수 있다. 실험 자료 3의 경우 제안한 방법이 유클리드 거리 방법보다 재현율과 정확률 면에서 1~2%정도 좋은 성능을 보이고 있다는 것을 알 수 있다. 또한 속도 면에서도 제안한 방법이 유클리드 거리 방법보다 약 11배정도 빠르다는 것을 알 수 있다.

위에서 살펴본 바와 같이 제안한 방법은 검색 성능과 속도 면에서 기존의 방법보다 우수하다. 또한 이 방법은 3.4절의 매칭 알고리즘의 임계값 T_1, T_2와 압축률을 나타내는 K값을 조절함으로써 사용자 요구에 적절하게 대응할 수 있다. 사용자가 재현율보다 정확률을 강조하는 경우에는 T_1과 T_2값을 낮추고 반대로 재현율을 강조하고 싶은 경우에는 T_1과 T_2값을 높여 매칭을 수행한다. 또한 속도보다 검색 성능이 중요한 매칭에서는 K의 값을 높여 수행하고, 어느 정도의 성능을 유지하면서 빠른 매칭을 수행하고 싶을 경우에는 K의 값을 적당한 수준으로 낮춰 매칭을 수행하면 된다. 이와 같이 제안한 방법은 다른 매칭 방법들과 달리 정확률과 재현율 간의 tradeoff와 검색 성능과 계산 시간 간의 tradeoff 조절이 가능하다는 장점을 갖고 있다.

실험 1의 결과를 분석한 결과 입력한 질의어에 대해 적합 단어가 아닌 경우에도 적합 단어로 찾는 경우는 대부분 질의어와 두 개 이상의 문자가 일치하는 단어, 또는 모양이 비슷한 단어들이다. 이것은 비슷한 위치에서 같은 길이나 두께를 갖는 획이 있을 경우 웨이브렛 변환한 계수의 값이 거의 비슷할 가능성이 높기 때문이다. 그림 11은 잘못 검색된 단어의 예이다. 또한 재현율을 떨어뜨리는 원인은 적합 단어가 검색되지 않는 경우인데, '스', '트', '이', '지' 등과 같이 모양이 간단한 단어들을 많이 포함하는 경우에 나타난다. 이와 같이 모양이 간단한 문자들은 모양이 복잡한 다른 문자들에 비해 웨이브렛 계수들이 문자의 특징을 잘 나타내지 못하기 때문이다. 이러한 문제를 해결하는 방안 중의 하나는 다중 특징을 이용하는 것이다. 예를 들어 한글이 수평과 수직 획으로 구성되어 있다는 사실에 근거하여, 획 성분을 잘 표현하는 특징들을 같이 사용하여 성능을 높일 수 있을 것이다. 그림 12은 적합 단어가 검색되지 않은 예이다.

표 3 성능 비교

종류	매칭 방법	Recall ratio(%)	Precision ratio(%)	Speed(number of word/second)
실험자료1	유클리드 거리	94.60	96.53	390
	제안한 방법 (K=60)	95.27	97.74	6,250
실험자료2	유클리드 거리	85.30	95.50	390
	제안한 방법 (K=120)	82.00	92.48	3,125
실험자료3	유클리드 거리	84.00	94.20	390
	제안한 방법 (K=90)	86.67	95.24	4,166

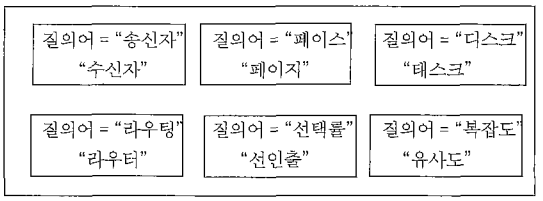


그림 11 잘못 검색된 단어 영상 예제

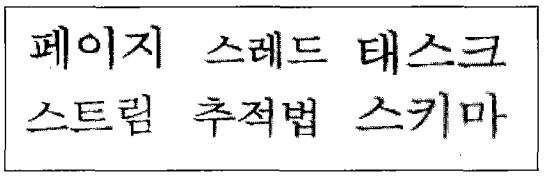
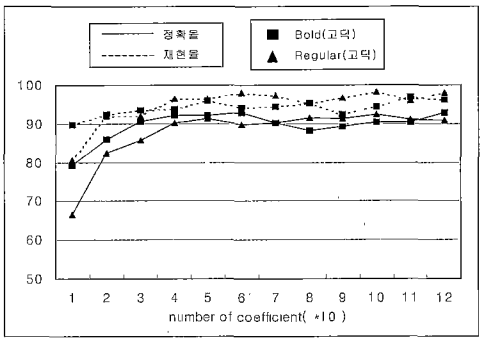


그림 12 미검색된 단어 영상 예제

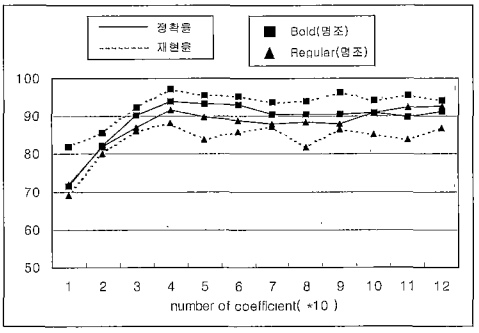
4.3 실험 2

표 4는 실험 2에 대한 성능을 요약하여 보여주고 있다. 또한 그림 13는 각각 고딕체와 명조체에 대한 성능을 그래프로 보여준다. 실험 1에서와 마찬가지로 특징 갯수가 많아짐에 따라 성능이 좋아지는데 약 40개 이상에서는 성능 향상이 거의 없다. 즉, 40개의 계수 (전체 계수의 3.9%)만 사용함으로써 검색 성능 저하없이 25.6배의 속도 향상을 얻을 수 있었다. 4.2절의 실험 1과 비교하면, 절대 성능에 약간의 차이가 있지만 전체적인 경향은 비슷하다는 사실을 알 수 있다.

실험 1에서는 문서 품질에 따라 세 종류의 실험 자료를 사용하였고, 실험 2에서는 폰트에 따라 4종류의 실험 자료를 사용하였다. 결론적으로 다양한 문서 품질과 폰트에 대해 제안한 웨이브렛 특징과 이를 이용한 매칭 알고리즘이 높은 검색 성능을 제공하며 사용자의 요구에 적응적으로 동작할 수 있는 장점을 제공한다고 말할 수 있다.



(a) 고딕체



(b) 명조체

그림 13 실험2의 검색 성능 그래프

표 4 실험 2의 검색 성능 표

종류	고딕체				명조체			
속성	bold		regular		bold		regular	
K	Recall ratio(%)	Precision ratio(%)	Recall ratio(%)	Precision ratio(%)	Recall ratio(%)	Precision ratio(%)	Recall ratio(%)	Precision ratio(%)
10	89.76	79.18	80.60	66.44	81.93	71.54	69.02	71.99
20	92.49	85.83	91.97	82.27	85.61	82.21	80.13	81.94
30	93.51	90.55	91.97	85.73	92.30	90.24	86.16	87.09
40	93.85	92.09	96.32	90.07	97.32	93.89	88.21	91.66
50	95.90	92.09	96.32	91.19	95.65	93.31	83.83	89.81
60	94.19	92.83	97.99	89.79	95.31	93.11	85.85	88.81
70	94.53	90.21	97.32	90.18	93.64	90.41	87.20	87.86
80	95.22	88.25	95.31	91.47	93.97	90.49	81.81	88.44
90	92.49	89.30	96.65	91.19	96.32	90.44	86.53	88.04
100	94.53	90.44	98.32	92.47	94.31	91.08	85.18	91.07
110	96.92	90.50	95.98	91.13	95.65	89.74	83.83	92.36
120	96.24	92.83	97.99	90.85	94.31	91.24	86.86	92.48

5. 결 론

OCR-기반의 텍스트 변환 방법의 대안으로서 영상-기반 한글 단어 인식 방법을 제안하였다. 제안한 방법은 웨이브렛 변환의 여러 가지 이점들을 이용하여 설계하였다. 제안한 방법을 테스트하기 위해 문서 품질과 폰트 종류에 따라 다양한 실험 자료를 가지고 실험하였다. 실험 결과 제안한 알고리즘이 기존의 알고리즘보다 검색 성능 면에서는 다소 우수하고 검색 속도 면에서는 매우 우수하다는 것을 알 수 있었다. 또한 제안한 방법은 정확률과 재현율 간의 tradeoff와 검색 성능과 계산 시간 간의 tradeoff 조절이 가능하다는 장점을 갖고 있다. 제안한 방법은 빠른 속도, 저장공간의 효율성, 신뢰할 수 있는 검색 성능을 제공해 주기 때문에 디지털 라이브러리 구축 등의 응용에 유용하게 사용할 수 있을 것이다.

향후 연구에서는 문자 모델 구성, 단어에서 문자 분할, 그리고 알고리즘이 갖고 있는 여러 파라미터 값을 정하는 것과 같은 작업을 통해 알고리즘의 성능을 향상시킬 수 있다. 또한 다중 특징을 이용하여 검색 성능을 향상시킬 수 있다.

참 고 문 헌

[1] A. Belaid, Retrospective document conversion: application to the library domain, *International Journal on Document Analysis and Recognition*, Vol.1, No.3, pp.125-146, December 1998.

[2] F.R. Chen, L.D. Wilcox, and D.S. Bloomberg, A comparison of discrete and continuous hidden Markov models for phrase spotting in text images, *Proceeding of ICDAR95*, Montreal, pp. 398-402, 1995.

[3] K.S Chung and H.U. Kwon, A feature-based word spotting for content-based retrieval of machine-printed English document images, Journal of Korean Information Science Society (B), Vol.26, No.10, pp.1204-1218, October 1999 (in Korean).

[4] R.A. DeVore, B. Jawerth, and B.J. Lucier, Image compression through wavelet transform coding, *IEEE Trans. on Information Theory*, Vol.38, No.2, pp.719-746, March 1992.

[5] D. Doermann, The retrieval of document images: a brief survey, *Proceedings of ICDAR97*, Ulm, pp.945-949, 1997.

[6] W.L. Hwang and F. Chang, Character extraction from documents using wavelet maxima, *Image and Vision Computing*, Vol.16, pp.307-315, 1998.

[7] C.E. Jacobs, A. Finkelstein, and D.H. Salesin, Fast multiresolution image querying, *Proceedings of SIGGRAPH95*, pp.277-286, 1995.

[8] S.-S. Kuo and O.E. Agazzi, Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models, *IEEE Trans. on Pattern analysis and Machine Intelligence*, Vol.16, No.8, pp.842-848, August 1994.

[9] S.W. Lee, C.H. Kim, H.Ma, and Y.Y. Tang, Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network, *Pattern Recognition*, Vol.29, No.12, pp.1953-1961, 1996.

[10] H. Ma, Y.Y. Tang, J. Liu, B.F. Li, and C.Y. Suen, Wavelet transform extracting features in Chinese character recognition, *Proceedings of ICPOL97*, pp.262-265.

[11] F. Murtagh and J.-L. Starck, Pattern clustering based on noise modeling in wavelet space, *Pattern Recognition*, Vol.31, No.7, pp.847-855, 1998.

[12] S. Pittner and S.V. Kamarthi, Feature extraction from wavelet coefficients for pattern recognition tasks, *IEEE Trans. on Pattern analysis and Machine Intelligence*, Vol.21, No.1, pp.83-88, January 1999.

[13] T. Shioyama, H.Y. Wu, and T. Nojima, Recognition algorithm based on wavelet transform for handprinted Chinese characters, *Proceedings of ICPR98*, Brisbane, pp.229-232, 1998.

[14] A.L. Spitz, Shape-based word recognition, *International Journal on Document Analysis and Recognition*, Vol.1, No.4, pp.178-190, May 1999.

[15] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin, *Wavelets for Computer Graphics*, Morgan Kaufmann, San Francisco, 1996.

[16] Y.Y. Tang, H. Ma, J. Liu, B.F. Li, and D. Xi, Multiresolution analysis in extraction of reference lines from documents with gray level background, *IEEE Trans. on Pattern analysis and Machine Intelligence*, Vol.19, No.8, pp.921-926, August 1997.

[17] P. Wunsch and A.F. Laine, Wavelet descriptors for multiresolution recognition of hadprinted characters, *Pattern Recognition*, Vol.28, No.8, pp.1237-1249, 1995.

[18] J. Zhu, T. Hong, and J.J. Hull, Image-based keyword recognition in Oriental language document images, *Pattern Recognition*, Vol.30, No.8, pp.1293-1300, 1997.

김 혜 금

1998년 전북대학교 컴퓨터과학과(이학사). 2000년 전북대학교 전산통계학과(이학석사). 2000년 ~ 현재 호원대학교 시간강사. 관심분야는 패턴인식, 컴퓨터 비전.

양 진 호

2000년 호원대학교 전자계산학과(이학사). 2000년 ~ 현재 전북대학교 전산통계학과 석사과정. 관심분야는 컴퓨터 비전, 문서 및 문자 인식.

이 진 선

1985년 전북대학교 전산통계학과(이학사). 1988년 전북대학교 대학원 전산통계학과(이학석사). 1995년 전북대학교 대학원 컴퓨터공학과(공학박사). 1988년 ~ 1992년 한국전자통신연구원 연구원. 1995년 ~ 현재 우석대학교 정보통신컴퓨터공학부 조교수. 관심분야는 영상처리, 패턴인식, 멀티미디어.

오 일 석

1984년 서울대학교 컴퓨터공학과 학사. 1984 ~ 1992년 한국과학기술원 전산학과 석사 박사. 1992년 ~ 현재 전북대학교 컴퓨터과학과 교수. 관심분야는 컴퓨터 비전, 문서 및 문자 인식 등임.