

MPEG-2 뉴스영상에서 문자영역 추출 및 문자인식

박 영 규[†] · 김 성 국^{††} · 유 원 영^{††} · 김 준 철^{†††} · 이 준 환^{††††}

요 약

본 논문은 MPEG-2로 압축된 뉴스의 NOD(News On Demand)를 위한 내용기반 뉴스색인 및 검색에 핵심적으로 이용될 수 있는 영상의 캡션문자 영역 추출 및 인식 방법을 제안한다. 제안된 방법은 문자가 있는 프레임(frame)을 탐색하는데 있어서 최소한의 복호화만을 함으로써 문자 프레임의 탐색시간을 절약하였으며, 문자 영역에 포함된 잡음을 효과적으로 제거하기 위한 전처리 과정을 포함한다. 또한 문자의 인식방법으로는 뉴스 영상에서 사용되는 문자의 활자체가 다양하지 않기 때문에 비교적 단순하게 개선된 패턴 정합 방법을 이용한다. 제안된 방법은 스포츠 뉴스 비디오에 대해 양호한 인식결과를 얻을 수 있었다.

Extraction and Recognition of Character from MPEG-2 News Video Images

Young-Kyu Park[†] · Sung-Guk Kim^{††} · Won-Young Yoo^{††} · Joon-Cheol Kim^{†††} · Joon-Whoan Lee^{††††}

ABSTRACT

In this paper, we propose the method of extracting the caption regions from news video and the method of recognizing the captions that can be used mainly for content-based indexing and retrieving the MPEG-2 compressed news for NOD(News On Demand). The proposed method can reduce the searching time on detecting caption frames with minimum MPEG-2 decoding, and effectively eliminate the noise in caption regions by deliberately devised preprocessing. Because the kind of fonts that are used for captions is not various in the news video, an enhanced template matching method is used for recognizing characters. We could obtain good recognition result in the experiment of sports news video by the proposed methods.

1. 서 론

MPEG은 오늘날 멀티미디어를 위한 가장 중요한 국제 표준으로 주목받고 있다. 또 뉴스 영상을 비롯한 여러 가지 동영상에서 MPEG-2를 이용한 압축은 보편화되

고 있고 NOD에서의 MPEG의 활용은 화질 및 데이터의 압축효율 때문에 세계적인 추세[1]로 되어가고 있다.

뉴스영상의 문자를 인식하는데 가장 시간 소요가 많은 부분은 문자가 있는 프레임(frame)을 찾는 부분이다. 1초에 30프레임이나 되는 많은 영상들을 일일이 복호화하여 문자 유무를 판단한다는 것은 많은 시간을 요구한다. 본 논문에서는 압축된 데이터에서 빠르게 프레임내의 문자유무를 판단하는 알고리즘을 사용한다. 또한 뉴스영상의 문자 인식의 특징은 많은 노이즈를 포함하기 때문에 문자영역 추출 및 이진화 등의 정

* 본 연구는 과학재단 특정 기초 연구 과제로서 수행되었음(과제 번호 : 97-01-00-01-01-3).

† 정 회원 : 전북대학교 대학원 전자공학과

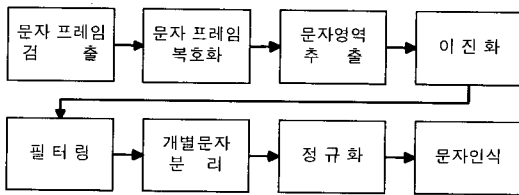
†† 정 회원 : 전북대학교 대학원 전자공학과

††† 정 회원 : 서남대학교 전기전자통신공학부 교수

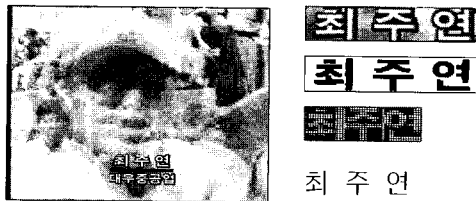
†††† 정 회원 : 전북대학교 전자공학과 교수

논문접수 : 1998년 12월 29일, 심사완료 : 1999년 3월 4일

교한 전처리 과정을 요구한다. 본 논문에서는 추출된 문자 프레임에서 문자영역 검출은 문자영상에서 나타나는 문자의 특징(흰색, 외곽선, 그림자)을 사용하여 검출하였고, 검출된 영역의 이진화시, 비디오영상에 적합한 연속되는 프레임의 합성, 이중문턱치와 영역확장, 분할과 병합(split and merge) 알고리즘을 이용하였다. 마지막으로 이진화된 영상의 노이즈를 제거하기 위해 수리형태학적인 필터링을 사용하였다. 이진화된 영상에 대해서는 평균 문자폭을 이용한 문자분리에 이어 정규화를 거쳐 문자인식을 시도하였다. 문자인식 방법은 구조적 특징을 이용한 인식, 신경망을 이용한 인식, 패턴매칭을 이용한 방법 중에 가장 인식률이 좋은 패턴매칭을 이용한 방법을 사용하였다. (그림 1)은 본 논문에서 제안된 문자인식의 전 과정을 보여 주고 있다.



(그림 1) 캡션문자인식 과정



(그림 2) 문자영역 추출 및 인식결과

(그림 2)의 첫 번째 그림은 검출된 문자 프레임을 보여주고 두 번째는 추출된 문자영역, 세 번째는 이진화를 거쳐 필터링된 영상, 네 번째는 정규화 이미지, 마지막에 인식결과를 보였다.

본 논문에서의 실험영상으로는 NOD들의 서비스를 고려하여 MPEG-2로 압축된 KBS 스포츠 뉴스를 사용하였다. 실험결과 문자가 있는 7개의 영상을 정확히 찾았고, 찾았던 영상의 126개의 문자중 117개의 문자를 인식하여 92.9%의 인식률을 얻었다.

본 논문의 제 2장에서는 MPEG-2 압축도메인에서의 문자프레임 검출방법을 기술하며, 제 3장에서는 문자영

역 추출방법을 제 4장에서는 문자부분만을 강조하기 위한 문자 프레임 합성 방법 및 이진화 과정을 제 5장에서는 구조적분석, 신경망, 패턴매칭을 이용하는 문자인식 방법을 서술하였고, 마지막으로 제 6장에서 제안된 알고리즘들을 이용한 실험결과 및 결론을 서술하였다.

2. MPEG-2 뉴스영상에서의 문자프레임 검출

뉴스영상의 문자를 인식하는데 앞서 먼저 문자가 있는 프레임을 검출하여야 한다. 이러한 검출을 위해서 MPEG으로 압축되어진 스포츠 뉴스의 모든 프레임에 대해서 압축을 푼 후 문자가 존재하는지 검사하고 문자인식을 수행한다는 것은 많은 처리 시간을 필요로 할뿐 아니라 같은 문자가 여러 번 인식되는 결과를 가져올 것이다. 제안된 방식에서는 문자를 포함하고 있는 영상(문자 프레임)을 B와 P를 제외한 I-프레임에서만 찾았으며, 또 DCT 역변환을 하지 않고 검색하였기 때문에 검색시간을 줄일 수 있었다.

F_{00}	F_{01}	F_{02}	...	F_{00}	$F_{01} +$	$-$
F_{10}	F_{11}					
F_{20}				$+$	$+$	$-$
\vdots				$F_{10} -$	F_{11}	$+$

(그림 3) 8×8 DCT Block에서 계수들의 물리적 의미

(그림 3)에 8×8 DCT 블록의 계수들을 보였다. F_{00} 는 DC성분을 의미하며, F_{01} 은 세로방향 경계(Edge)치, F_{10} 는 가로방향 경계치, F_{11} 은 대각선방향 경계치를 나타낸다. 일반적으로 문자를 포함한 DCT 블록은 수평, 수직, 대각선 경계치가 크게 나타나기 때문에, 먼저 DCT 계수들을 이용하여 문자를 포함한 DCT 블록들을 찾고 한 장의 프레임에서 이러한 DCT 블록의 개수를 세어 실험적으로 얻어진 임계치 이상인 경우에 이 프레임을 문자 프레임으로 간주한다. 이러한 과정은 DCT 역변환을 필요로 하지 않는다. 또 TV뉴스에서 문자가 나타나는 부분이 화면 하단이라는 가정을 토대로 화면 하단에서의 매크로 블록들만을 고려하여 화면 전체에 대한 탐색에 비해 시간을 절약하는 방법을 이용하였다.

8×8 DCT 블록 내에서 수평, 수직, 대각선 경계와 DCT계수와의 관계를 <표 1>에 나타내었다.

<표 1> 수직, 수평, 대각선 경계와 DCT 계수의 관계

수 직	$F_{10} = 0$	$ F_{01} \gg 0$
수 평	$F_{01} = 0$	$ F_{10} \gg 0$
대 각 선	$ F_{01} = F_{10} $	$ F_{01} \gg 0$ $ F_{10} \gg 0$

영상내에 수평, 수직, 대각선 경계를 많이 포함하고 있는 경우에는 문자가 없을 경우에도 문자 프레임으로 오인식 할 수가 있다. 그러나 이러한 경우는 드물고 또 복호화하여 캡션문자인식시 문자가 없는 경우로 분류된다. 위에서 제안한 방법은 대용량의 비디오영상에서 문자가 있는 프레임들 DCT 역변환을 거치지 않고 빠른 시간에 찾는 장점을 가진다.

3. 검출된 문자프레임에서 문자영역의 검출

TV비디오 영상에 나타나는 캡션 문자의 일반적인 특징을 살펴보면 이러한 문자들은 배경화면과의 대비를 위해서 여러 가지 처리를 하였다. 가장 일반적인 것은 외곽선 처리이고 그림자 처리, 또는 실크(silk) 처리 등이 사용되고 있다. 이러한 처리 등은 흰색 글씨와 대비를 이루며, 이 부분들에서 일반적으로 픽셀사이의 구배값(gradient)이 크다. 문자영역의 추출은 픽셀사이의 구배값으로 했으며, 먼저 가로방향 영역 추출, 다음으로 세로방향 추출을 하였다.

<표 2>에 가로방향 추출 알고리즘들, (그림 4)는 가로방향으로 추출된 영역을 보여준다. 또 <표 3>에 세로방향 추출 알고리즘을 보였다.

<표 2> 가로방향 문자영역 추출 알고리즘

Step 1 : $G(x', y) = g(x+2, y) - g(x, y)$, where $g(x, y) = \text{gray level of pixel}(x, y)$
Step 2 : $N(y) = \# \text{ of } G(x', y) \text{ that is larger than } T1$ for each line, where $T1$ is a threshold
Step 3 : Row-Region = set of y at that $N(y)$ is larger than $T2$, where $T2$ is a threshold



(그림 4) 뉴스영상에서의 가로방향 영역 추출

<표 3> 세로방향 문자영역 추출 알고리즘

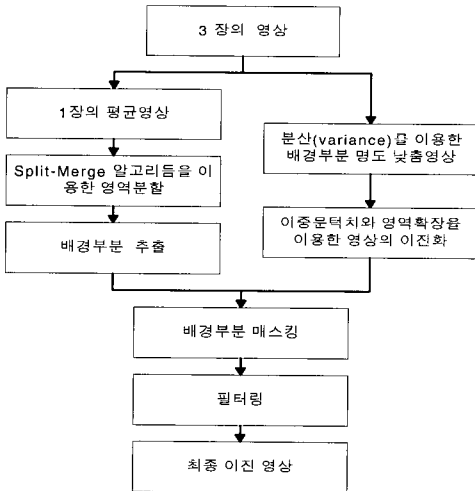
Step 1 : Devide Row-Region into sub-regions of 8 pixel-width moving 4 pixels for a step.
Step 2 : $G(x', y) = g(x+2, y) - g(x, y)$ $G(x, y') = g(x, y+2) - g(x, y)$
Step 3 : $N(y) = \# \text{ of } G(x', y) \text{ that is larger than } T1$ for each y $N(x) = \# \text{ of } G(x, y') \text{ that is larger than } T1$ for each x
Step 4 : Col-Region = set of sub-regions whose $N(x)+N(y)$ is larger than $T2$
Step 5 : Char-Region = region from left to right of Col-Region

4. 문자영상의 이진화 및 개별 문자분리

4.1 문자영상의 이진화

뉴스영상의 이진화는 일반적인 문서영상의 이진화와 달리 문턱치를 이용한 이진화가 용이하지 않다. 문자부분은 흰색이 대부분인데 반해 배경으로 사용된 영상은 여러 가지 색을 가지고 있고, 특히 스포츠 뉴스영상의 경우, 흰색 유니폼, 운동장과 같은 영상의 명도는 문자부분의 명도보다 높은 경우가 많이 나타난다. 이러한 점을 고려하여 문자영역의 이진화 과정을 (그림 5)와 같이 수행하였다. 먼저 같은 문자를 포함하는 연속된 문자프레임에서 배경으로 쓰인 영상은 계속 바뀐다는 것을 이용하여 가장 처음 I-프레임과 가장 끝, 그리고 중간에서 각각 한 장씩 세 장의 영상을 얻는다. 이 세 장의 영상에서 문자부분은 명도 차이가 거의 없고 이에 반해 배경으로 사용된 영상은 계속 바뀌었으므로 프레임사이에 명도 차이가 난다. 그러므로 배경부분에서의 명도값의 분산(variance)을 이용하여 최대로 배경부분의 명도를 낮추어 문턱치를 이용한 이진화를 용이하게 하였다. 자세한 과정은 <표 4>에 나타내었고 이진화의 모든 과정은 추출된 문자영역에 대

해서만 고려하여 실행속도를 높였다.



(그림 5) 문자영역의 이진화 과정

<표 4> 분산을 이용한 영상합성 과정

<p>Step 1 : Calculate $M(x, y) = \text{Mean of } g(x, y) \text{ for } 3 \text{ consecutive frame}$</p> <p>Step 2 : Calculate $V(x, y) = \text{Variance of } g(x, y) \text{ for } 3 \text{ consecutive frame}$</p> <p>Step 3 : Calculate $G(x, y) = M(x, y) - V(x, y)$ $G(x, y)$: gray level of new image</p>
--



(그림 6) 세 장의 영상과 합성된 영상

문턱치를 이용한 이진화는 이중문턱치와 영역확장 [2]을 이용하였으며, 두 개의 문턱치는 추출된 문자영역 중에 문자부분이 25% 정도를 차지한다는 실험치에 의

한 결과에서 CDF(Cumulative Distribute Function)에서 각각 25%, 30% 값을 잡았다. (그림 7)의 (b)에 이중문턱치를 이용하여 이진화를 수행한 결과를 보였다. 그림에서 알 수 있듯이 물론 단일 문턱치를 이용한 경우보다는 나았지만 이 영상 역시 노이즈가 많아 그대로 문자인식에 사용될 경우 많은 오인식을 유발하였다. 이러한 노이즈를 제거하기 위해 추출된 문자영역을 문자부분과 배경부분을 나누는 방법을 제안하였다. 이 방법은 split-merge 알고리즘을 이용하여 문자영역을 영역 분할하고 각각의 영역 중 크기에 따라 문자부분, 배경부분으로 나눔으로써 가능하였다. 곧 문자의 각 자소가 차지하는 영역은 배경(바탕)이 차지하는 영역에 비해 크기가 크지 않으므로 split-merge 알고리즘에서 얻은 각 영역의 크기 데이터를 이용하여 크기 필터링을 할 수 있다. 크기 문턱치는 실험에 의해 추출된 문자영역의 높이의 11배로 잡았으며, 문턱치보다 영역의 크기가 작은 경우를 제거한 영역을 배경영역으로 간주하여 이진영상을 마스크(masking)하는데 사용하였다. split-merge 알고리즘에 사용된 유사성의 척도로서는 화소들의 명도값의 분산값을 이용하였고 이때 사용된 영상은 단순히 세 장 영상의 평균 영상을 사용하였다. 최종적으로 이진화된 영상은 수리형태학적인 필터링을 거쳐 노이즈가 거의 없는 이진영상을 얻을 수 있었으며, (그림 7)에 각 단계의 결과 영상을 보였다.



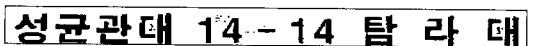
(a) 추출된 문자 영상



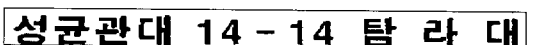
(b) 이중 문턱치에 의해 이진화된 영상



(c) split-merge에 의해 추출된 배경영역



(d) 마스크 후의 이진영상

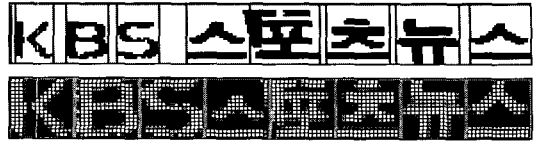


(e) 최종적인 이진영상

(그림 7) 문자영상의 이진화

4.2 개별 문자 분리

이진화 된 영상에서 세로방향의 프로파일을 구하면 개별 문자 분리를 쉽게 할 수 있다. 하지만 개별 문자 분리를 단순히 프로파일을 이용하여 구하게 되면 노이즈에 의해 붙은 문자 혹은 한글의 구조적 특성상 분리되는 문자들이 존재하게 된다. 이러한 문자들의 분리와 병합여부를 판단하여 개별문자를 얻어야 하는데 이러한 과정들은 평균 문자 너비를 계산하여 판단하였다. 먼저 평균 문자 너비를 구하고 프로파일 특성을 이용하여 가능한 모든 분리점을 찾아낸다. 이러한 분리점 후보에서 평균문자 너비를 고려하여 병합여부를 판단하게 된다. 먼저 (그림 8)의 첫 번째 그림과 같은 영상에서 평균 너비를 계산하게 되는데 'K'자는 반으로 나뉘었고, '스'와 '포'는 합쳐져 있다. 이러한 부분의 너비는 평균너비에 비해 상당히 작거나 상당히 크다. 이렇게 평균너비에 비해 상당히 작거나 상당히 큰 너비를 뺀 나머지 너비에서 다시 한번 평균 문자 너비를 계산한다. 다음으로 '스포'와 같이 노이즈에 의해 붙은 문자들은 평균 문자 너비 보다 훨씬 크므로 분리 후보점을 찾게 된다. 분리 후보점은 (그림 8)의 두 번째 그림과 같이 영상을 반으로 나누어 윗 영상과 아랫 영상에서 각각 프로파일을 구해 분리후보점을 찾는다. (그림 8)에서는 각각 하나씩 두 개의 분리 후보점을 찾은 경우이다. 병합 단계에서는 분리점들과 평균 문자 너비를 고려하여 병합하게 되는데 'K'자는 분리된 각각의 영역을 병합한 경우가 보다 평균 문자 너비와 비슷하기 때문에 병합을 하게되고 '스포'에서도 마찬가지로 평균 문자너비에 맞게 병합을 하였다. (그림 8)의 세 번째 그림은 병합 후의 분리점을 보여주고 있다. 마지막으로 분리된 문자에 대해 정규화를 시도하였다. 정규화는 12×12 혹은 15×15 크기로 비교적 작게 하여 처리속도를 높였으며 또 경계선에 생기는 획의 유실에 대해 일반화 능력을 향상 시킬 수 있었다. 정규화 과정은 개별 문자영상을 12×12 혹은 15×15의 메쉬로 나누어 각각의 메쉬에 대해 획이 차지하는 비율을 계산한 명암값 메쉬를 사용하였다. (그림 8)의 네 번째 그림은 정규화 이후의 영상을 보여준다.



(그림 8) 개별 문자 분리 과정

5. 문자 인식

본 논문에서는 구조적 특징을 이용하는 방법[3][4], 신경망을 이용하는 방법[5][6] 및 패턴매칭에 의한 방법등 3가지의 대표적인 한글 인식방법을 이용하여 문자인식을 수행하였다. 이 중에서 패턴 매칭에 의한 방법이 가장 우수한 인식률을 나타내었으며 캡션 문자 인식에는 패턴매칭을 이용한 방법이 효과적임을 알 수 있었다. <표 6>에 47×46 크기의 견고딕 문자체 2350 자에 대한 각각의 인식률을 나타내었다.

<표 6> 문자인식기의 성능

인식 방법	노이즈가 없는 영상	실제 캡션 문자 영상
구조적 특징을 이용한 방법	98%	70%
신경망을 이용한 방법	100%(훈련 데이터)	70%
패턴 매칭을 이용한 방법	100%	92.9%

5.1 구조적 특징을 이용한 문자인식

구조적 특징을 이용한 문자인식방법은 자체적으로 개발한 인식기를 사용하였다. 하지만 일반 문서 영상과는 달리 캡션 문자 인식에 저조한 인식률을 보였으며, 이러한 저인식률의 주요 원인은 다음과 같았다.

1. 문자의 경계선부분의 왜곡이 심하다.
2. 이진화시 자소간에 또는 문자 획들 사이에 붙은 경우가 발생
3. 이진화시 획의 유실
4. 획과 비슷하게 나타나는 노이즈

이러한 원인들은 캡션 문자가 삽입될 때 경계선 부분이 배경과 혼합되어 균일하지가 않고 획이 많은 문자의 경우 획의 두께가 두꺼운 견고딕이기 때문에 획사이의 경계선이 유실되어 구조적 특징을 바꾸는 경우가 많기 때문이었다.

5.2 신경망을 이용한 문자인식

신경망은 BP(Backpropagation)를 이용하여 구성하였고 먼저 한글을 여섯가지 유형으로 나누는 유형분류 신경망과 각각의 유형에 맞게 훈련된 자소기반 인식 신경망, 2단으로 구성하였다. 정규화 사이즈는 15×15로 일반화 능력을 키웠으며 훈련 데이터로서 한글을 통해 얻은 깨끗한 견고딕체 한글 2350자를 사용하였다. 먼저 훈련 데이터에 대해서는 100%의 인식률을 얻었지만 실제 데이터에서는 위에 언급한 캡션문자의 특징 때문에 70%를 밑도는 인식률을 얻었다. 이러한 저인식률의 주요원인은 훈련 데이터를 확보하지 못했다는 데 있다. 일반문서의 인식과는 달리 TV의 캡션문자를 2350개의 완성형 한글 전체에 대해 여러 가지의 패턴으로 이미지를 얻는다는 것은 실제로 불가능하였다.

5.3 패턴 매칭을 이용한 문자인식

패턴 매칭을 이용한 문자인식의 가장 큰 장점을 들어보면 알고리즘이 간단하다는 것이다. 하지만 이 경우 많은 수의 글자를 인식하려면 많은 수의 패턴이 메모리 상에 상주해야 되고 또 함께 인식속도도 늦어진다. 본 논문에서 패턴 매칭은 12×12크기로 정규화된 명암영상의 유클리디안(euclidean) 거리를 이용하였으며, <표 7>에 패턴매칭을 이용한 문자인식에 대해 설명하였다.

<표 7> 패턴매칭을 이용한 문자인식

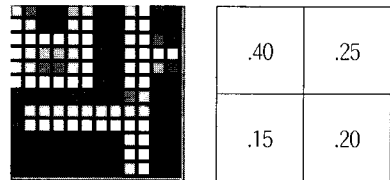
Step 1 : Let $d_i(x, y) = g(x, y) - p_i(x, y)$ $g(x, y)$: gray level of recognizing data $p_i(x, y)$: gray level of i-th pattern data Step 2 : Let $D_i = \sum_{(x,y)} d_i^2(x, y)$ Step 3 : Get Minimum D_i
--

패턴매칭에 사용된 패턴은 완성형 한글 2350자와 영숫자 36자 특수문자 4자를 합쳐 전체 2390자의 견고딕글씨체에서 얻었으며, 12×12 크기로 정규화 하였다.

5.3.1 Normalized 4-Mesh 특징 매칭

N4M(Normalized 4-Mesh) 특징은 정규화된 데이터를 4개의 메쉬(mesh)로 나누어 각 메쉬의 문자 점유율을 퍼센트로 계산한 것으로 (그림 9)에 나타내었다. N4M 특징 매칭은 12×12 메쉬의 패턴 매칭 이전에

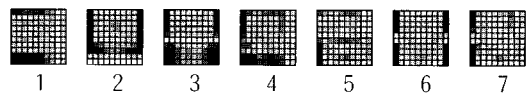
수행되어 먼저 각각의 패턴과 인식될 데이터의 패턴의 유사성의 척도로 사용된다. 인식될 데이터와의 N4M 특징 매칭에서 유사성이 큰 패턴들에 한해서 12×12 크기의 패턴매칭을 시도함으로써 패턴 매칭에 소요되는 시간을 평균 1/10로 줄일 수 있었다.



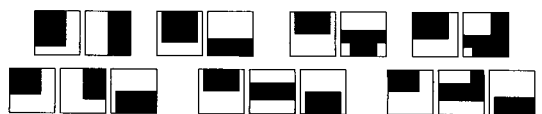
(그림 9) Normalized 4-Mesh 특징

5.3.2 한글의 구조를 이용한 패턴 매칭 오인식 정정

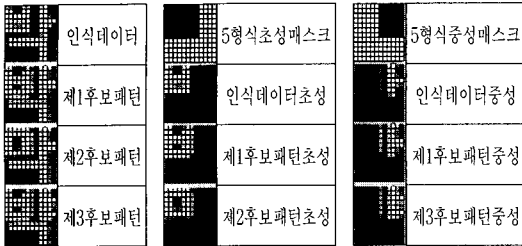
패턴매칭을 이용한 인식에서 나타나는 오인식의 많은 경우가 비슷한 자음이나 모음으로 기인한다. 곧 ‘히’와 ‘히’의 경우나 ‘포’와 ‘프’의 경우는 문자획이 두꺼운 견고딕에서 오인식이 될 가능성이 높다. 이러한 오인식을 정정하기 위해 인식된 패턴을 7가지 유형으로 나누고 각각의 유형의 구조에 맞는 패턴매칭을 이용하는 방법을 제시하였다. 먼저 (그림 12)의 첫 번째 그림을 보면 ‘박’자의 인식될 데이터에 대해 제1, 제2, 제3후보 패턴이 각각 ‘박’, ‘막’, ‘박’과 같이 모두 5형식 문자에 속하게 된다. 제1, 제2후보 패턴은 초성이 다른 경우이므로 (그림 11)의 5형식 마스크중 초성 마스크를 이용하여 초성부분만 다시 패턴 매칭을 시도하였다. 마찬가지로 제1, 제3후보 패턴에 대해서는 5형식 중성 마스크를 이용하여 패턴 매칭을 시도하였으며 최종적으로 제1후보 패턴이 가장 가까운 패턴이므로 ‘박’자로 인식하게 된다. (그림 11)에 나타난 한글의 구조에 따른 7가지 유형의 초, 중, 중성의 마스크는 (그림 10)와 같이 각 유형의 평균 영상에서 얻을 수 있었다.



(그림 10) 7가지 한글 문자 유형의 평균 이미지



(그림 11) 7가지 유형에 맞게 만들어낸 17개의 마스크



(그림 12) 한글 구조를 이용한 패턴매칭 과정

5.3.3 문자 프레임과 문자인식 결과

이제 까지 TV문자영상에서 캡션문자인식을 시도하여 보았다. 시험영상으로는 MPEG-2로 압축된 KBS 스포츠뉴스를 사용하였으며, 먼저 압축영상의 DCT 복호화를 하지 않고 7장의 문자가 있는 프레임을 모두 찾을 수 있었다. 찾아진 7장의 영상에서 문자영역 추출을 시도하였으며 15개의 문자영역을 추출하여 100%의 추출률을 얻을 수 있었다. 추출된 15개의 문자영역에서 이진화, 개별문자분리, 정규화를 거쳐 126개의 문자영상으로 인식을 시도하였다. 여러 가지의 문자인식 방법 중에서는 패턴 매칭을 이용한 인식방법이 가장 인식률이 좋았으며 117개를 인식하여 92.9%의 인식률을 얻을 수 있었다. 오인식이 된 문자들을 살펴보면, 日, 美와 같은 한자 2자, 이진화시 문자유실이나 노이즈에 의한 문자 3자, 기타 문자 4자등이 있었다.

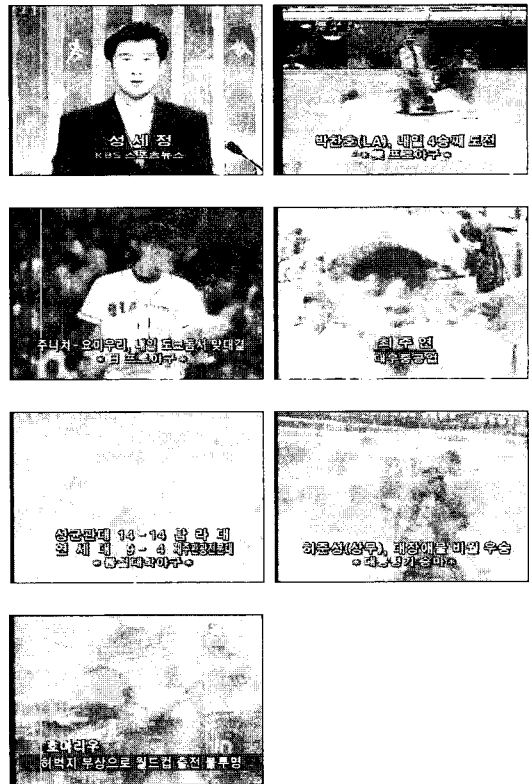
6. 결 론

뉴스영상의 문자를 인식하는데 가장 시간 소요가 많은 부분은 문자가 있는 프레임(frame)을 찾는 부분이다. 1초에 30프레임이나 되는 많은 영상들을 일일이 복호화하여 문자 유무를 판단한다는 것은 많은 시간을 요구하므로 본 논문에서는 DCT계수를 이용하여 빠르게 프레임내의 문자유무를 판단하는 알고리즘을 사용하였다. 또한 뉴스영상의 문자 인식의 특징은 많은 노이즈를 포함하기 때문에 문자영역 추출 및 이진화등의 정교한 전처리 과정을 요구한다. 본 논문에서는 추출된 문자 프레임에서 문자영역 검출은 문자영상에서 나타나는 문자의 특징(흰색, 외곽선, 그림자)을 사용하여 검출하였고, 검출된 영역의 이진화시, 비디오영상에 적합한 연속되는 프레임의 합성, 이중문턱치와 영역확장, 분할과 병합(split and merge) 알고리즘을 이용하였다. 마지막으로 이진화된 영상의 노이즈를 제거하기 위해

수리형태학적인 필터링을 사용하였다. 이진화된 영상에 대해서는 평균 문자폭을 이용한 문자분리에 이어 정규화를 거쳐 문자인식을 시도하였다. 문자인식 방법은 구조적 특징을 이용한 인식, 신경망을 이용한 인식, 패턴매칭을 이용한 방법 중에 가장 인식률이 좋은 패턴매칭을 이용한 방법을 사용하였다.

본 논문에서의 실험영상으로는 NOD들의 서비스를 고려하여 MPEG-2로 압축된 KBS 스포츠 뉴스를 사용하였다. 실험결과 문자가 있는 7개의 영상을 정확히 찾았고, 찾아진 영상의 126개의 문자중 117개의 문자를 인식하여 92.9%의 인식률을 얻었다.

뉴스영상의 캡션 문자인식은 다양한 형태의 노이즈에 불구하고 문자체가 다양하지 않기 때문에 비교적 높은 인식률을 얻을 수 있었는데 앞으로 엑스포체와 같이 스포츠 뉴스에서 가끔 사용하는 문자체의 인식에 대해 고려가 되어야 하겠고, 기울어진 문자, 문자의 명도가 배경보다 낮은 문자 등에 대해서도 고려되어야 하겠다.



(그림 13) 시험 스포츠 뉴스 영상