

멀티미디어 단말기 사용자를 위한 축구 경기 비디오의 점수상자 추출

*김원준 *김창익
한국정보통신대학교
*jazznova@icu.ac.kr

Scoreboard Extraction from Soccer Videos for Multimedia Mobile Users

*Kim, Won-Jun *Kim, Changick
Information and Communications University

요약

최근 정보통신 기술의 급속한 발전으로 소형 이동형 단말기를 이용한 각종 스포츠 경기 시청이 두드러지게 증가하고 있다. 그럼에도 불구하고 이동형 단말기를 통해 제공되는 영상은 일반 TV나 HDTV용으로 제작되어 소형 이동형 단말기의 사용자가 화면을 통해 스포츠 경기의 상황을 인식하는데 많은 불편함을 주고 있다. 특히, 경기 진행 시간이나 점수를 포함하는 점수상자(scoreboard)는 경기의 상황을 파악하는데 매우 중요한 역할을 하나, 소형 이동형 단말기의 작은 화면에서는 점수상자의 내용을 정확히 인식하기가 쉽지 않다. 이에 본 논문은 많은 사람들이 즐겨보는 축구 경기에 대하여 짧은 학습 기간을 갖는 효율적인 점수상자 추출 방법을 제안하고자 한다. 제안하는 알고리즘은 점수상자와 주변 환경의 밝기 정보를 이용한 점수상자 경계 좌표 추출, 학습을 통한 최적의 경계 좌표 결정, 점수상자 영역 추출 및 확대의 세 단계로 구성된다. 제안하는 알고리즘은 점수상자가 없는 프레임에서도 몇 프레임 앞서 표시된 점수상자의 저장을 통해 디스플레이가 가능하도록 하였다. 다양한 축구 경기 비디오에 대한 실험을 통해 제안된 알고리즘이 소형 이동형 단말기 상에서 점수상자를 추출하고 이를 사용자가 쉽게 인식할 수 있도록 확대하여 디스플레이 하는 좋은 해결책을 보이고자 한다.

1. 서론

IT 기술발전과 더불어 DMB(Digital Multimedia Broadcasting)와 같은 새로운 모바일 TV 서비스의 등장으로 소형 이동형 단말기 사용이 현저히 증가하고 있다. 이와 같은 추세에 힘입어 소형 이동형 단말기를 이용한 축구 경기 시청 또한 두드러지게 증가하고 있다. 축구 경기에서 점수상자는 경기 상황을 파악하는데 매우 중요한 요소인 경기 진행 시간과 점수를 포함하고 있다. 따라서 사용자는 경기에 대한 정보를 얻기 위해 점수상자를 보기 원한다. 하지만 서비스되는 영상이 단지 소형 이동형 단말기의 화면 크기에 맞게 축소되어 디스플레이 되기 때문에 영상정보의 손실이 발생하게 된다. 예를 들면, 점수상자가 작아서 현재 상황을 알 수 없는 경우나 현재 영상 내에 점수상자가 없는 경우 등 사용자가 불편을 느끼는 경우가 발생한다. 이러한 문제점을 극복하기 위해 관심영역만을 확대하여 디스플레이 하는 지능형 디스플레이 방법에 대한 많은 연구가 이루어졌다[1]. 그 결과 축구 경기를 시청함에 있어 사용자가 느끼는 불편함이 어느 정도 해결되었으나 관심영역 안에 점수상자가 포함되지 않는 경우가 발생하기 때문에 점수상자를 별도로 추출할 필요가 있다.

그동안 광고판 영역에 대한 연구는 몇몇 있어 왔으나 점수상자 추출에 대한 관련 연구는 활발히 이루어지지 않았다. Okihisa[2] 등은 에지(Edge) 정보를 이용하여 그라운드 영역과 캡션(Caption) 영역을 구별하고 있지만 캡션 영역 내에서 점수상자와 광고판을 구별하기 쉽지 않은 문제점 있다. Yoon[3] 등은 광고판의 수평방향으로의 밝기 변화가 그라운드와 다르다는 사실을 이용하여 광고판 영역을 구별하고 있지만 다양한 디자인의 광고판과 점수상자가 존재하기 때문에 밝기의

차이를 이용해 이 둘을 구별하기는 쉽지 않다. 따라서 점수상자의 특성만을 이용하여 점수상자를 추출하는 것이 바람직하다.

본 논문에서는 짧은 학습 기간을 이용한 효율적인 점수상자 추출 방법을 제시하고자 한다. 추출된 점수상자는 작은 화면에서도 알아보기 쉽도록 확대하여 디스플레이 한다. 논문의 구성은 다음과 같다. 먼저, 점수상자와 주변 환경의 밝기 정보를 이용하여 점수상자 경계 좌표 후보군을 추출한다. 둘째, 일정 프레임 동안의 학습을 통해 후보군 중에서 최적의 좌표를 결정한다. 셋째, 결정된 좌표를 이용하여 점수상자를 추출하고 이를 확대하여 디스플레이 한다. 마지막으로 결론 및 향후 과제를 도출한다.

2. 점수상자 경계 좌표 추출

2.1. 점수상자의 특징

점수상자는 축구 경기에서 일반적으로 화면의 왼쪽 상단에 위치하며, 주로 일정한 밝기의 경계선을 가진 사각형 형태로 디스플레이 된다. 또한, 자연 영상과는 달리 그래픽이며, 영상 내에서 디스플레이 되는 위치가 일정하다. 좀더 구체적으로 살펴보면 클로즈업 샷[1]일 경우 점수상자의 주변 환경이 정적(static) 상태인 경우가 많다. 이러한 경우 좀더 정확한 점수상자 추출이 가능하다는 장점이 있다. 다양한 축구 경기를 관찰한 결과 대부분의 점수상자가 화면의 시작점으로부터 가로 길이의 1/2, 세로 길이의 1/3 크기 영역 안에 포함되기 때문에 탐색 범위를 이 영역 안으로 제한하여 점수상자를 효율적으로 추출 할 수 있도록 한다(그림 1 참조).

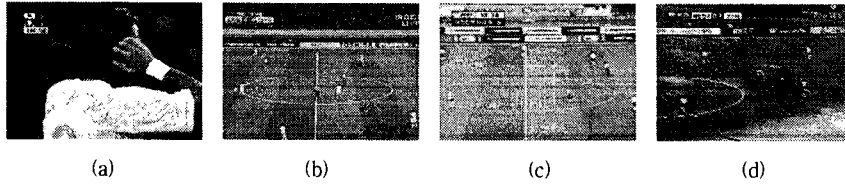


그림 1. (a), (b), (c), (d) 축구 경기의 다양한 점수상자. 작은 화면에서 점수상자의 내용을 확인하기가 쉽지 않음.

2. 2. 한 프레임 내에서 점수상자 경계 좌표 추출

점수상자는 일정한 밝기를 가진 경계선을 가지고 그 안에 점수나 시간을 표현하는 문자를 포함하기 때문에 Markov source의 한 형태인 자연영상과 달리 이웃 픽셀간의 밝기 차이가 크게 변한다. 따라서 밝기 차이들의 합이 지정된 임계값 보다 커지는 순간을 점수상자의 경계선으로 인식할 수 있다. 점수상자의 왼쪽 상단 x 좌표를 추출하기 위해서 먼저, 그림 2와 같이 탐색 영역의 왼쪽 경계에서 오른쪽 방향으로 한 열 단위로 각 픽셀과 이웃하는 픽셀의 밝기 차이를 구하고 절대값을 취하여 더한다. y 좌표를 추출하기 위해서는 탐색 영역의 위쪽 경계에서 아래쪽 방향으로 한 행 단위로 각 픽셀과 이웃하는 픽셀의 밝기 차이를 구하고 절대값을 취하여 더한다. 점수상자의 오른쪽 하단 x 좌표를 추출하기 위해서는 탐색 영역의 오른쪽 경계에서 왼쪽 방향으로, y 좌표를 추출하기 위해서는 탐색 영역의 아래쪽 경계에서 위쪽 방향으로 위와 같은 과정을 수행한다. 즉, 수평, 수직방향 측면에 대해 sum of pixel difference (SPD)를 다음과 같이 정의한다.

$$SPD_v(y) = \sum_{x=0}^{width/2} |f(x,y) - f(x+1,y)| \quad (0 \leq y < \frac{height}{3})$$

$$SPD_h(x) = \sum_{y=0}^{height/3} |f(x,y) - f(x,y+1)| \quad (0 \leq x < \frac{width}{2}) \quad (1)$$

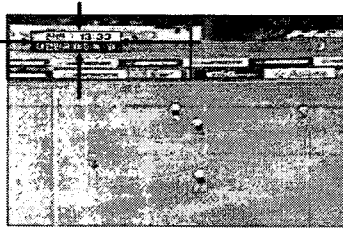


그림 2. 굵은 선으로 표시한 사각형 내부가 탐색영역이고 화살표 방향이 각 측면에 대해 SPD를 구하는 방향이다.

식(1)에서 x, y 는 탐색 영역안의 수평, 수직 방향의 좌표를 나타내고, $f(x, y)$ 는 (x, y) 좌표에서의 밝기를 나타낸다. SPD를 이용하여 그 값이 지정된 임계값 보다 커지는 순간을 점수상자의 경계로 인식할 수도 있으나 영상에 따라 SPD 값이 다양하기 때문에 임계값을 지정하기 쉽지 않다. 따라서 SPD 값의 변화량을 이용해야 할 필요가 있다. SPD 값의 변화량은 식(1)을 이용하여 행(또는 열) 단위로 구한 SPD 값의 차이로 알 수 있고 이 차이가 지정된 임계값 보다 커지는 순간을 점수상자의 경계로 인식하게 된다. 즉, 수평, 수직 방향 측면에 대해 difference of SPD (DSPD)를 다음과 같이 정의한다.

$$DSPD_v(y) = |SPD_v(y) - SPD_v(y+1)| \quad (0 \leq y < \frac{height}{3})$$

$$DSPD_h(x) = |SPD_h(x) - SPD_h(x+1)| \quad (0 \leq x < \frac{width}{2}) \quad (2)$$

점수상자를 사각형 형태로 추출하기 위해선 점수상자의 왼쪽 상단 좌표와 오른쪽 하단의 좌표가 필요하기 때문에 위의 방법을 점수상자의 4개의 측면에 대해 적용한다. $DSPD_v, DSPD_h$ 이 지정된 임계값 보다 커지는 순간의 (x, y) 좌표를 점수상자의 경계 좌표로 결정한다.

본 논문에서는 실험에 사용한 모든 비디오에 대해 임계값으로 200을 사용하였으나 각 영상에 따라 [200, 300] 범위의 값 중 적절한 값을 선택 할 수 있다. 그림 3은 그림 1. (a)에 대하여 왼쪽 상단의 x, y 의 좌표를 추출하기 위한 $DSPD_v, DSPD_h$ 의 변화를 그래프로 나타낸 것이다. 그래프를 통해 $DSPD_v, DSPD_h$ 값이 지정된 임계값 보다 갑자기 커지는 순간을 알 수 있다. 여기서는 임계값 200에 대하여 수평, 수직방향에 대한 각각의 x, y 좌표가 (29, 21)로 정해짐을 알 수 있다. 따라서 점수상자 추출을 위한 왼쪽 상단의 좌표는 (29, 21)이 된다. 동일한 방법으로 오른쪽 하단의 좌표도 추출한다.

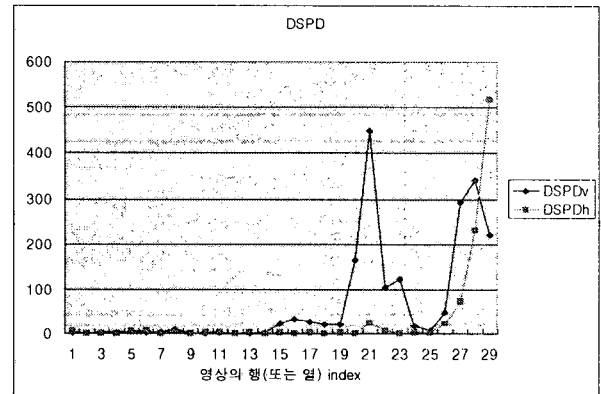


그림 3. 수평, 수직 방향에 대한 DSPD

그림 4는 기술한 알고리즘을 실제 영상에 적용한 결과를 보여주고 있다. 일반적으로 축구 경기에서 점수상자의 주변 환경은 역동적으로 변하기 때문에 한 프레임만을 이용하여 정확한 점수상자 경계 좌표를 결정하기는 매우 어렵다. 그림 4에서와 같이 주변 픽셀들을 포함한 영역을 점수상자로 인식하는 경우가 발생한다. 따라서 일정 프레임 구간에 대한 학습을 통해 최적의 경계 좌표를 결정할 필요가 있다.

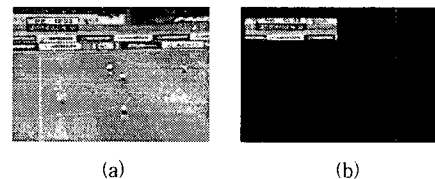


그림 4. (a) 축구 경기 영상. (b) 제안하는 알고리즘을 이용하여 한 프레임에 대해 추출된 점수상자 영역. 점수상자 주변 픽셀의 변화가 크기 때문에 한 프레임 안에서 정확한 점수상자 추출이 이루어지지 않고 있다.

3. 학습을 이용한 최적의 경계 좌표 결정

3. 1. 최적의 경계 좌표에 대한 정의

최적의 경계 좌표란 추출된 점수상자 영역이 주변 픽셀들을 포함하지 않고 점수상자 영역만을 포함하는 좌표를 말한다. 역동적으로 변하는 주변 환경 때문에 일반적으로 한 프레임 안에서 추출된 점수상자 영역은 정확하지 않다. 점수상자 주변에 광고판이나 관중석이 존재할 때 SPD와 DSPD 값만을 이용하여 점수상자인지 아닌지 구별하기가 쉽지 않기 때문에 추출된 점수상자 영역은 더 많은 주변 픽셀들이 포함 되게 된다. 그림 4.(b)와 같이 실제 점수상자는 일반적으로 추출된 영역 안에 포함되기 때문에 최적의 점수상자 영역을 추출하기 위해서는 일정한 학습 기간 동안 점수상자의 왼쪽 상단 경계 좌표를 위해 추출된 후보군 중 최대 x , 최대 y 값을 가지는 좌표가 필요하고, 점수상자의 오른쪽 하단 경계 좌표를 위해 추출된 후보군 중 최소 x , 최소 y 값을 가지는 좌표가 필요하다는 사실을 알 수 있다. 따라서 일정 프레임 구간에 대한 학습을 통해 위의 조건을 만족하는 경계 좌표를 결정한다.

3. 2. 최적의 경계 좌표 결정

일정 프레임 구간 동안 비교를 위해 지정된 초기값을 설정한 후, 매 프레임마다 현재 프레임에서 구한 경계 좌표와 이전 프레임까지의 최적의 경계 좌표를 비교한다. 점수상자 왼쪽 상단의 좌표를 위해서는 최대 x , 최대 y 가 필요하므로 x, y 값이 각각 더 큰 좌표를, 점수상자 오른쪽 하단의 좌표를 위해서는 최소 x , 최소 y 가 필요하므로 x, y 값이 각각 더 작은 좌표를 저장한다. 기본적인 알고리즘은 아래와 같다.

```

i : frame number
learning_period : 100 frames

temp_max_x = temp_max_y = -1;
temp_min_x = temp_min_y = 1000;

for(i = 0; i < learning_period; i++)
    if(temp_max_x < top_left_x(i))
        temp_max_x = top_left_x(i);
    if(temp_max_y < top_left_y(i))
        temp_max_y = top_left_y(i);
    if(temp_min_x > bottom_right_x(i))
        temp_min_x = bottom_right_x(i);
    if(temp_min_y > bottom_right_y(i))
        temp_min_y = bottom_right_y(i);
    
```

$temp_{max}$ 와 $temp_{min}$ 의 초기값으로 각각 -1과 1000을 사용하였다. i 는 현재 프레임에서의 추출된 좌표를 나타낸다. 또한 학습 구간으로 100 프레임을 사용하였다. 100 프레임 구간 동안 $temp_{max}$ 와 $temp_{min}$ 에 저장된 값과 현재 프레임에서 추출한 왼쪽 상단 좌표와 오른쪽 하단 좌표 값을 비교하는 과정을 반복하여 최종 값을 도출한다.

학습이 끝나면 더 이상 좌표 추출을 위한 처리과정을 실행하지 않고 $temp_{max}$ 와 $temp_{min}$ 에 저장된 최적의 경계 좌표 x, y 를 이용하여 스포어보드 영역을 디스플레이 하게 된다.

4. 점수상자 디스플레이

이동형 단말기의 작은 화면에서도 알아보기 쉽게 3장에서 도출한 최적의 점수상자 영역을 선형 보간법을 사용하여 2배로 확대한다. 점수상자가 없는 프레임에서도 점수상자를 디스플레이 하기 위해서 몇 프레임 앞서 표시된 점수상자를 저장한다. 본 논문에서는 점수상자의 주변 환경이 정적인 클로즈업 샷[1] 프레임의 점수상자 영역을 저장한다. 저장한 점수상자 영역과 3장의 알고리즘을 통해 결정한 최적의 경계 좌표를 이용하여 추출한 영역의 밝기 차이가 지정된 임계값 보다 크면 현재 프레임에 점수상자가 없는 것으로 판단하고 저장한 점수상자 영역을 디스플레이 한다.

5. 실험결과

이 실험은 Win32 환경에서 Visual Studio 2003 (C++)을 이용하여 개발하였으며, MPEG 복호화를 위해 FFMpeg 라이브러리를 이용하였다. 우리는 5분짜리 축구 경기 영상을 352x240, 29.97fps의 MPEG-1 포맷으로 부호화하여 실험에 이용하였다. 모든 과정은 Pentium - 4 3.00GHz PC에서 실시간으로 수행되었다.

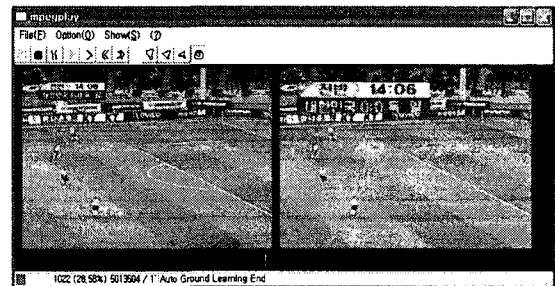


그림 5. 실험 결과. 축구 경기 영상(왼쪽). 일정 프레임 구간 동안의 학습이 끝나고 최적의 점수상자 영역을 추출한다. 사용자가 알아보기 쉽게 2배 확대하여 디스플레이 한다(오른쪽).

다양한 영상에 대해 적용한 결과는 그림 6과 같다. 100 프레임 구간의 학습을 통해 추출한 점수상자 영역이 2배 확대되어 디스플레이 되고 있다. 점수상자 추출을 위해 두개의 좌표만을 이용하기 때문에 사각형 형태의 점수상자에 비해 돌출된 영역을 포함하고 있거나 사각형 형태가 아닌 점수상자는 다소 부정확하게 추출되고 있음을 볼 수 있다(그림 6 참조).

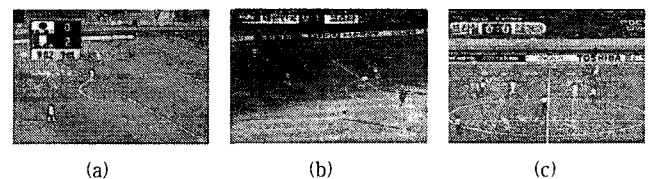


그림 6. (a) 사각형 형태의 점수상자. (b), (c) 사각형 형태가 아닌 점수상자. 점수상자 추출을 위해 두개의 좌표만을 이용하기 때문에 추출된 영역이 다소 부정확하다.

추출의 정확성을 측정하기 위해 본 논문에서는 다음과 같이 정의되는 Recall과 Precision을 사용하였다.

$$\text{Recall} = \frac{\text{추출된 점수상자 영역 중 실제 점수상자에 속하는 픽셀 수}}{\text{실제 점수상자에 속하는 픽셀 수}} \times 100\%$$

$$\text{Precision} = \frac{\text{추출된 점수상자 영역 중 실제 점수상자에 속하는 픽셀 수}}{\text{추출된 점수상자 영역에 속하는 모든 픽셀 수}} \times 100\%$$

정확성에 관계없이 추출된 점수상자 영역이 실제 점수상자 영역 보다 크다면 Recall 값이 커지고 추출된 점수상자 영역이 실제 점수상자 영역 보다 작다면 Precision 값이 커지기 때문에 Recall과 Precision을 동시에 살펴야 한다. 4개의 축구 경기 비디오를 이용하여 Recall과 Precision을 구하였으며, 그 결과는 그림 7과 표 1, 2에 자세히 나타내었다.(영상 1 : 브라질 대 크로아티아, 영상 2 : 일본 대 이탈리아, 영상 3 : 대한민국 대 독일, 영상 4 : 대한민국 대 코스타리카.)

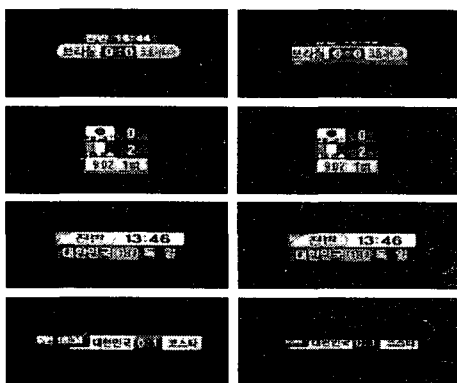


그림 7. 실제 점수상자 영역(왼쪽 : 위에서부터 아래로 각각 영상 1, 영상 2, 영상 3, 영상 4). 제안된 알고리즘에 의해 추출된 점수상자 영역(오른쪽).

표 1. 각 영역에 해당하는 픽셀 수

	실제 점수상자 영역	추출된 점수상자 영역	추출된 영역 중 실제 점수상자에 속하는 영역
영상 1	1501	1509	1263
영상 2	1788	1610	1610
영상 3	2232	2232	2232
영상 4	1678	866	866

표 2. Recall과 Precision을 측정한 결과

	Recall(%)	Precision(%)
영상 1	84.14	83.69
영상 2	90.04	100
영상 3	100	100
영상 4	51.61	100

그림 7에서 사각형 형태를 가지는 점수상자에 비해 영상 1과 4와 같이 점수상자가 사각형 형태가 아닌 경우 Recall과 Precision값이 떨어짐을 확인할 수 있었다.

우리는 실제 모바일 기기의 응용을 위해서 PDA용으로도 개발하였다. Microsoft Embedded Visual Studio 4.0을 이용하여 개발하였으며, 실험에는 HP iPAQ hx4700을 사용하였다. 성능 측정을 위해 h.264 포맷으로 재생시간 2분을 갖는 축구 경기 비디오를 이용하였다.

(대한민국 대 독일 : 2004년 친선 축구 경기, 이탈리아 대 일본 : 2004년 아테네 올림픽 축구 경기.) 제안하는 알고리즘을 적용한 영상 두 개에 대해 [1]에서 사용한 식을 이용하여 성능 측정된 결과를 표 3에 정리하였다.

표 3. PDA에서의 성능 평가

	대한민국 vs. 독일	이탈리아 vs. 일본
점수상자 추출 없을 시 평균 재생 속도	31.045 frame/sec	30.166 frame/sec
점수상자 추출 시 평균 재생 속도	29.251 frame/sec	28.667 frame/sec
시간 증가율(%)	5.78	4.97

6. 결론 및 토의

우리는 이동형 단말기의 작은 화면에서도 점수상자를 쉽게 볼 수 있도록 점수상자를 추출한 후 확대하여 디스플레이 하는 시스템을 제안하였다. 5장 실험결과에서 보인 것처럼 제안한 시스템은 일반 PC에서 실시간으로 동작하였으며 PDA에서 전체 디스플레이에 약 29 frame/sec의 속도를 보이고 있으며 사용자의 취향에 따라 추출 기능의 on/off가 가능하다.

그러나 현재의 시스템에서는 그림 6의 (b), (c)와 같이 사각형 형태가 아닌 점수상자는 시간을 표시하는 불출된 영역 등에 의해서 정확히 추출되지 않고 있다. 또한, 점수상자가 화면의 왼쪽 상단에 위치하지 않은 경우나 그래픽으로 구성된 영역이 영상 내에서 여러 개 존재할 때 제안한 알고리즘으로 점수상자를 추출 할 수 없다. 따라서 위와 같은 문제점을 해결하기 위해 현재 우리는 영상 내 모든 영역에서 그래픽 영역을 추출 할 수 있는 실험을 진행 중에 있다. 이렇게 추출한 그래픽 영역에서 점수상자를 판단 할 수 있는 알고리즘 또한 개발 중에 있다. 이를 통해 축구 경기 이외에도 다양한 스포츠 경기나 뉴스의 헤드라인 박스 추출 등에 응용 될 수 있을 것으로 기대된다.

참고 문헌

- [1] C. Kim, et al, "Intelligent video display to raise quality of experience on mobile devices", in *Proc. Applications of Digital Image Processing XXIX*, SPIE 6312, San Diego, August 2006.
- [2] O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka, "An object detection method for describing soccer games from video", in *Proc. Multimedia and Expo*, vol. 1, Aug. 2002, pp. 45-48.
- [3] Ho-Sub Yoon, Young-lae J. Bae, and Young-kyu Yang, "A soccer image sequence mosaicking and analysis method using line and advertisement board detection", *ETRI Journal*, vol. 24, Dec. 2002, pp. 443-454.