

이동형 단말기 사용자를 위한 축구경기 비디오의 시청경험 향상 방법

(Raising Visual Experience of Soccer Video for Mobile Viewers)

안 일 구 [†] 고 재 승 [†] 김 원 준 [†] 김 창 익 ^{**}
 (Ilkoo Ahn) (Jaeseung Ko) (Wonjun Kim) (Changick Kim)

요약 최근 멀티미디어 신호처리 및 통신 기술의 발전에 힘입어 작은 LCD 패널을 통한 스포츠경기 시청이 증가하고 있다. 하지만 멀티미디어 단말에 방송되는 대부분의 영상들은 주로 경제적인 이유로 일반 TV나 HDTV 용으로 제작되어 단순히 크기만을 변환하거나 추가적인 편집 없이 녹화되고 있는 실정이다. 이는 작은 이동형 단말 사용자들이 경기화면 내의 상황을 인식하는데 있어서 많은 불편함을 겪는다. 예를 들어, 원거리 샷 카메라 기법으로 찍힌 축구 경기 동영상의 경우, 운동장 내의 공과 선수들은 매우 작아서 알아보기가 힘든 경우가 발생한다. 또한 경기 진행 시간이나 점수를 포함하는 점수상자(scorebox)의 내용 역시 시청자가 쉽게 알아보기 힘들다. 따라서 소형 디스플레이 시청자들의 원활한 이해를 위한 지능형 디스플레이 기술이 필요하다. 이를 위한 핵심기술의 하나가 관심 영역을 자동으로 결정하고 확대하여 보여주는 일이다. 여기서 관심영역이란 시청자들이 화면 내에서 다른 부분에 비해 더욱 관심을 갖게 되는 영역을 말하며 축구경기 비디오의 경우, 주로 상단 모서리에 존재하는 점수상자나 원거리 샷에서의 공을 둘러싼 주변영역 등이 해당된다.

본 논문에서 우리는 이동형 단말기 시청자들을 위한 시청경험 향상을 위한 방법을 제안한다. 경기장면에서 관심영역의 추출을 위해 화면 내에서 시각적으로 현저한 부분의 검출에 관심을 갖는 방법 대신, 축구 경기 비디오 고유의 특징을 이용하는 도메인 한정적인 접근법을 이용한다. 제안된 시스템은 크게 두 가지 모듈 관심영역 결정, 점수상자 추출로 구성된다. 실험 결과를 통해 제안된 알고리즘이 이동형 단말기 상에서 지능형 디스플레이를 위한 좋은 해결책임을 보이고자 한다.

키워드 : 지능형 디스플레이, 이동형 기기, 관심영역, 축구비디오 분석, 점수상자 추출

Abstract The recent progress in multimedia signal processing and transmission technologies has contributed to the extensive use of multimedia devices to watch sports games with small LCD panel. However, the most of video sequences are captured for normal viewing on standard TV or HDTV, for cost reasons, merely resized and delivered without additional editing. This may give the small-display-viewers uncomfortable experiences in understanding what is happening in a scene. For instance, in a soccer video sequence taken by a long-shot camera techniques, the tiny objects (e.g., soccer ball and players) may not be clearly viewed on the small LCD panel. Moreover, it is also difficult to recognize the contents of the scorebox which contains the elapsed time and scores. This requires intelligent display technique to provide small-display-viewers with better experience. To this end, one of the key technologies is to determine region of interest (ROI) and display the magnified ROI on the screen, where ROI is a part of the scene that viewers pay more attention to than other regions. Examples include a region surrounding a ball in long-shot and a scorebox located in the corner of each frame.

In this paper, we propose a scheme for raising viewing experiences of multimedia mobile device users. Instead of taking generic approaches utilizing visually salient features for extraction of ROI in a scene, we take domain-specific approach to exploit unique attributes of the soccer video. The proposed

[†] 학생회원 : 한국정보통신대학교 전자공학과
 ahn19@icu.ac.kr
 jsko0117@icu.ac.kr
 jazznova@icu.ac.kr

^{**} 정 회원 : 한국정보통신대학교 전자공학과 교수
 ckim@icu.ac.kr
 논문접수 : 2007년 1월 22일
 심사완료 : 2007년 5월 2일

scheme consists of two modules: ROI determination and scoreboard extraction. The experimental results show that the proposed scheme offers useful tools for intelligent video display on multimedia mobile devices.

Key words : Intelligent display, Mobile devices, Region of interest, Soccer video analysis, scoreboard extraction

1. 서론

최근 DVB-H(Digital Video Broadcasting-Handheld) 및 T-DMB(Terrestrial-Digital Multimedia Broadcasting), S-DMB(Satellite-DMB) 등과 같은 이동형 비디오 서비스의 등장으로 언제 어디서나 TV를 즐길 수 있는 시대가 도래하였다[1]. 하지만 경제적인 이유 등으로 인하여 대부분의 서비스들은 기존의 화면 크기를 단지 축소하여 이동형 서비스에 제공하고 있는 실정이다. 이동형 단말에서 이러한 축소된 영상을 재생하는데 있어 필요한 해상도와 재생률을 측정한 [2]에서 보고된 실험에 의하면, 해상도가 큰 영상의 직접적인 축소는 시각적으로 상세한 부분에 대한 손실을 가져옴을 알 수 있었다. [2]에서 제시된 실험 결과에 따르면, 이러한 손실은 축구 비디오와 같은 필드스포츠인 경우 더욱 크고 뮤직비디오, 뉴스, 애니메이션의 순으로 조금씩 줄어들었다. 즉, 시청 경험을 높이기 위해서는 다양한 단말기에 따른 상황 인지 기반의 내재 컴퓨팅 환경이 제공되어야 할 필요가 있다.

필드스포츠와 같은 특정 내용을 담은 비디오의 경우, 해상도의 직접적인 축소에 따른 문제를 해결하기 위해서는 소형 이동 단말기 시청자들이 관심 영역(Region of interest 또는 ROI)이 확대된 화면을 시청할 수 있도록 해주는 지능형 디스플레이 기술의 개발이 필요하다. 여기서 관심영역이란 시청자들이 화면상에서 가장 관심을 갖게 되는 영역 또는 시청자들이 다른 영역에 비해 더욱 주의를 기울이는 영역으로 정의될 수 있다. [3]에서는 필드스포츠 중에서도 가장 인기 있는 콘텐츠 중 하나인 축구경기에 대해 구체적인 조사를 실시하였다. 640×480, 480×360의 높은 해상도의 영상에 대해서는 관심 영역을 적용한 것보다 원본 영상을 그대로 보는 것이 높은 만족도를 보였던 반면, 320×240 이하의 해상도를 가진 영상에 대해서는 관심 영역만을 확대한 영상을 보는 것이 원본 영상을 보는 것 보다 높은 만족도를 보이고 영상을 시청 할 때의 불만 사항 건 수 역시 큰 폭으로 줄어드는 것을 확인하였다. 관심 영역 추출 기법을 적용함으로써, 즉 상황 인지 기반의 내재 컴퓨팅 환경을 제안함으로써 시청 만족도를 높인 것이다. 상황인지 콘텐츠 적용, 큰 영상의 자동 브라우징[4], 트랜스 코딩, 지능형 정보 관리 등에 활용 될 수 있는 관심영역의 추

출은 비디오 장면(scene)을 의미적으로 분석하는 첫 단계가 될 수 있으므로 영상 분석을 위해서도 매우 중요한 기술이라 할 수 있다.

그 동안 관심영역을 결정하기 위한 여러 방법들이 연구되어 왔다. [5]에서 Itti 등은 국부적으로 존재하는 현저함(saliency)을 측정하기 위해 여러 개의 공간적 시각적 특징들을 하나의 특징점 맵(saliency map)으로 나타낼 수 있는 시각적 주의 모델을 제안하였으나 동영상으로의 확장은 제대로 고려하지 않았다. Cheng [6]등은 그들의 논문에서 명암, 색상 및 모션 정보에 기초하여 비디오에서 관심영역을 결정하는 방법을 연구하였다. 그 논문에서, 중요한 물체는 높은 색상 또는 명암의 대비를 갖는다는 가정 하에 비디오의 각 영상 프레임에서 가장 눈에 띄는 특징점을 가리키는 방법을 제안하였다. 하지만 중요한 물체가 어둡거나 낮은 대비를 가질 수 있으므로 이 가정은 항상 옳은 것은 아니다. 비디오 스킴밍(Skimming)과 요약을 위해 Ma 등이 제안한 또 다른 비디오 분석 방법에서는 모션, 스피치, 카메라 동작 및 비디오 편집 정보 등을 이용하여 사용자의 관심 모델이 제시되었다[7]. Fan 등은 일반적인 비디오의 이러한 특징점들을 이용하여 작은 디스플레이에 자동-확대축소 기능을 구현하였다[8]. 위에서 언급한 비디오 화면내의 현저한 특징들을 이용하여 관심영역들을 추출하는 방법들이 효과적인 해결책으로 보이나 필드 스포츠 비디오의 경우에는 이러한 접근 방법이 적절치 않다. 왜냐하면 스포츠 비디오의 경우, 확대가 필요하지 않은 샷도 존재하며 화면 안에 현저한 특징점을 갖는 작은 물체들이 동시에 여러 개가 존재 할 수 있어 위의 특징점에 기반한 방법을 적용 시 다수의 관심영역이 발견될 수 있기 때문이다.

관심영역 지정에 대한 많은 연구들이 존재하지만 소형 이동 단말기를 이용한 콘텐츠 시청과 같은 특정상황을 고려한 관심영역 확대를 고려한 연구는 없었다. 최근 IPTV에서는 시청자를 위한 개인화 및 주문화로 시청 만족도를 높이려는 연구가 활발하다. 그 중 관심영역을 기반으로 한 계층화 영상코딩(Scalable Video Coding)에 대한 연구가 활발한데 이 역시 관심영역 확대를 고려하지는 않는다.

본 논문에서 우리는 이전 연구[9]의 연장으로서, 축구 영상의 향상된 성능의 지능적 디스플레이 기법을 제안

한다. 축구 비디오의 독특한 특성들을 이용한 도메인 한정적인 (domain-specific) 접근 방법을 이용하기로 한다. 이러한 관점으로 볼 때, 축구 경기에는 두 종류의 프레임이 - 원거리샷 프레임과 비원거리샷 프레임이 - 존재함을 알 수 있다. 원거리샷 프레임은 원거리에서 촬영되었으므로 작은 LCD 화면으로 시청할 때 공과 선수, 점수상자의 크기가 작아 알아보기 어렵다(그림 1(a) 참조). 이 경우 관심영역을 추출하여 확대 재생하여 시청하면 사물을 인식하기가 보다 쉽다. 중거리 샷이나 근거리 샷 같은 비원거리 샷의 경우 원 영상을 그대로 보여주어도 공과 선수 등의 물체를 시청에는 문제가 없다(그림 1(b), (c)). 한편, 점수상자의 경우에는 샷의 종류에 관계없이 경기 진행 시간이나 점수를 확인하는데 있어 작은 크기로 인해 불편함을 겪을 수 있다. 본 논문이 제안하는 이동형 단말기 시청자를 위한 축구경기 비디오의 디스플레이 방법은 크게 다음 두 가지 모듈로 나누어 진다: 1) 관심영역 추출 및 확대 2) 점수상자 추출 및 확대.

이 중, 우리의 이전 시스템[9]의 연장이기도 한 관심영역 추출 및 확대 모듈에서, 우리는 이전의 시스템[9]에서와 달리 영상의 시작에서 추가적인 시간이 드는 그라운드 색상 학습 단계를 제외하였다. 그리하여 입력 비디오에 대해 PDA에서 일반 영상의 프레임율인 30fps에 근접하는 처리속도를 낼 수 있었다. 또한 이전 시스템[9]에서는 MPEG영상에 기반한 실험을 수행하였으나 본 연구에서는 실제 DMB영상 규격인 H.264/AVC에 기반하여 시스템을 구현하였다.

이 논문은 다음과 같이 구성되어 있다. 2장에서는 본 논문에서 제안하는 관심영역과 점수상자의 추출 및 확대 방법에 대해 기술한다. 3장에서는 실험 결과를 제시하며 이에 대한 결론을 4장에 기술한다.

2. 제안 알고리즘

2.1 관심영역 추출 및 디스플레이



그림 1 (a) 원거리 샷 영상. 관심영역을 추출하여 확대 재생할 필요가 있음. (b) 중거리 샷 영상. (c) 근거리 샷 영상

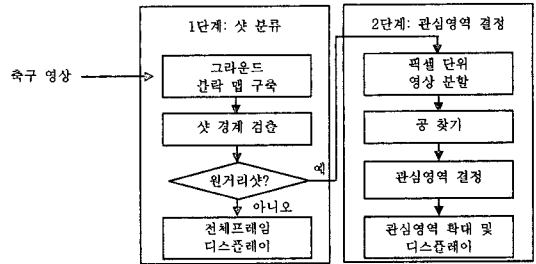


그림 2 제안된 알고리즘의 Block diagram.

제안된 알고리즘 과정을 그림 2에 나타내었다.

2.1.1 그라운드 픽셀 검출

그라운드 픽셀 검출에서의 우리 목표는 i 번째 프레임

I^i 를 그라운드 영역과 나머지 영역으로 나누는 것이다. 이진 배열 $Ground(x,y)$ 는 I 에 의해 두 가지 경우로 나누어 정의된다. 여기서 I 는 r, g, b 값의 합의 평균값이다.

Case 1: $I(x,y) > 50, Ground(x,y) =$

$$\begin{cases} 1, \text{ ground, if } \left\{ \begin{array}{l} ((85g > 95b) \text{ AND } (95g > 85r)) \text{ AND} \\ (g-r) + b > 30 \text{ AND} \\ (I < 150) \end{array} \right. \\ 0, \text{ non-ground, otherwise,} \end{cases} \quad (1)$$

Case 2: Otherwise, $Ground(x,y) =$

$$\begin{cases} 1 \text{ if } ((85g > 95b) \text{ AND } (95g > 85r)), \\ 0 \text{ otherwise.} \end{cases} \quad (2)$$

여기서 r, g, b 와 I 의 인덱스 (x, y) 는 편의상 생략한다. (1)와 (2)에 나타나 있듯이 그라운드 픽셀 검출 알고리즘을 두 가지 경우로 나누었다. 하나는 일반적인 그라운드 색상을 다루고 나머지 하나는 그림자가 진 그라운드 색상을 다룬다. 일반적인 그라운드 색상을 다룬 식 (1)은 $I(x,y)$ 가 본 논문에서 지정한 값 50보다 큰 경우를 말한다. 식 (2)는 그림자가 진 그라운드 영역을 검

출하는데 적용된다. (1)의 첫 번째 조건식은 그라운드 픽셀에서는 $g > r$ 그리고 $g > b$ 의 관계가 있음을 나타낸다. 충분한 관찰을 수행한 뒤, 관계식이 약간 수정되었다. (1)의 두 번째 조건식은 그라운드 색상과 비슷한 유니폼 색상을 구별하기 위한 조건식이다. 마지막으로 세 번째 조건식은 볼과 라인이 그라운드 픽셀로 검출되는 것을 방지해준다. 그라운드 색상이 영상마다 조금씩 다르지만 수많은 영상을 관찰한 결과를 토대로 도출해 낸 위 조건식은 그림 3에서 보는 바와 같이 그라운드 픽셀을 잘 검출해낸다.

첫 단계에서 그라운드 색상 학습이 필요했던 우리의 이전 논문[9]과 달리, 이 새로운 개념은 그라운드 색상 학습 단계를 필요로 하지 않고 보다 강건하고 빠른 성능을 보여주고 있다. 또한 제안된 개념은 Ekin[10] 또는 Wan[11]이 제안한 개념들보다 빠르고 그라운드의 일부분이 그림자가 드리워졌더라도 잘 작동한다.

2.1.2 그라운드 블록 맵 구축

이전 절에서 우리는 각 픽셀이 그라운드에 포함되는지의 여부를 결정할 수 있었다. 빠른 샷 경계 검출과 샷 구분을 위해, 전체 프레임을 16x16 블록으로 나누어 그라운드 블록 맵 $GB(i,j)$ 를 만들었다. 어느 한 블록 $GB(i,j)$ 이 그라운드 블록인지를 판단하기 위해 [9]에서는 블록 내부의 모든 픽셀을 사용하였으나 빠른 검출을 위해 아래 식 (3)과 같이 블록 내의 대표점만을 이용한다.

$$GB(i, j) = \begin{cases} 1 \text{ (ground block)} & \text{if } Ground(x_a, y_b) = 1 \\ 0 \text{ (non-ground block)} & \text{Otherwise.} \end{cases} \quad (3)$$

여기서 (x_a, y_b) 는 그 블록을 대표하는 대표점으로서

각 블록의 왼쪽 위 점으로부터 (a, b)만큼 떨어진 곳에 위치한 한 픽셀을 나타낸다.

$$a = \begin{cases} \frac{BLOCK_WIDTH}{4}, & \text{if } j = \text{odd} \\ \frac{BLOCK_WIDTH \times 3}{4}, & \text{if } j = \text{even} \end{cases},$$

$$b = \frac{BLOCK_HEIGHT}{2} \quad (4)$$

이와 같이 대표점을 블록의 정 가운데 점으로 정하지 않고 홀수번째 행과 짝수번째 행 블록에서 다르게 설정한 이유는, 정 가운데로 설정할 경우 그라운드에 존재하는 가운데 흰색 라인이 영상에서 수직으로 잡혀 한 블록열(block-column) 모두가 non-ground 블록으로 판단되는 것을 방지하기 위함이다. 블록열(block-column) 모두가 non-ground 블록으로 판단되면 식 (7)에 의해 샷종류 판정에서 오류가 날 수 있다.

그림 4에서 보듯이 블록 내부의 모든 픽셀을 조사하여 블록의 속성을 구하는 방법[9]에 비해, 식 (4)에서 결정되는 블록 내부의 한점의 그라운드 속성으로 블록의 그라운드 속성을 구하는 것은 소요되는 시간을 대폭 줄일 수 있는 반면, 판정오류는 증가시킬 수 있다(그림 4(a)와 4(b) 비교). 그러나 식 (5)을 이용한 홀 채우기 과정을 통해 그라운드 블록 맵 내부에 고립되어 존재하는 홀을 채우고 나면 최종결과는 거의 대등함을 알 수 있다(그림 4(c)와 (d) 비교).

$$GB(i, j) = 1 \text{ if } [GB(i-1, j) = 1 \text{ and } \{GB(i+1, j) = 1 \text{ or } GB(i+2, j) = 1\}] \text{ or } [\{GB(i-2, j) = 1 \text{ or } GB(i-1, j) = 1\} \text{ and } GB(i+1, j) = 1] \quad (5)$$

위 식은 수직방향으로 1개 또는 2개의 홀이 있을 경

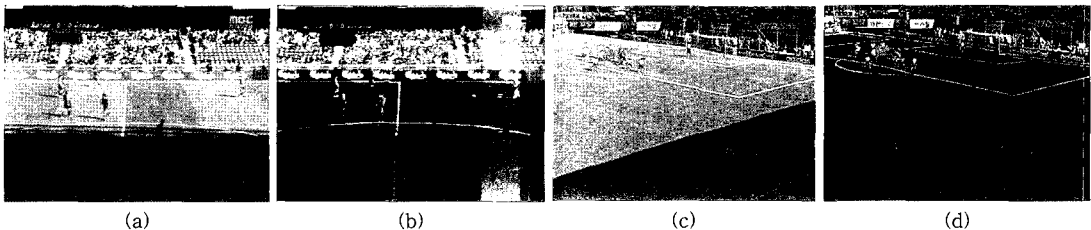


그림 3(a), (c) 그림자영역이 있는 프레임. (b), (d) 제안된 개념을 이용하여 검출된 그라운드 영역 (검정)

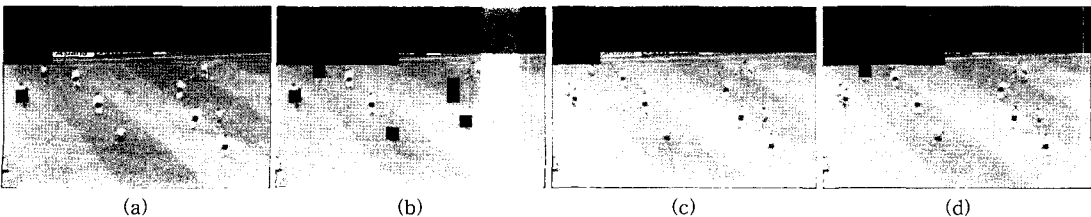


그림 4 초기 그라운드 블록 맵 비교. (a) [9]로부터의 초기 GBM. (b) 제안된 알고리즘에 의한 초기 GBM. (c)와 (d) (a)와 (b) 각각으로부터 홀이 채워진 GBM

우 그라운드 블락으로 채워주는 것을 나타낸다.

2.1.3 샷 경계 검출

샷 경계 검출은 샷 경계에서만 샷을 분류함으로써 계산량을 줄이고, 일시적인 화면 상태 변화로 인한 판정 오류를 줄임으로써 샷 종류 판정의 정확성을 높일 수 있다. 샷 경계를 검출하는 방법으로는 여러 가지 방법이 제시되고 있는데[12], 우리는 그라운드 블락 맵의 시간적 불연속성(Temporal Block Difference, TBD)을 체크하여 효율적이고 빠르게 샷 경계를 검출하였다.

$$TBD_i = \sum_x \sum_y \{GB_{i-3}(x, y) \otimes GB_i(x, y)\} \quad (6)$$

여기서 \otimes 는 XOR연산을 뜻한다. 우리는 $TBD_{i-1} < \theta_{ShotChange}$ 이고 $TBD_i \geq \theta_{ShotChange}$ 인 조건을 만족할 때 샷의 경계로 지정하였다($\theta_{ShotChange} = 30$ 사용). 빠르게 변화하는 장면의 경우 $TBD_i \geq \theta_{ShotChange}$ 인 상황이 많이 발생하기 때문에 $TBD_{i-1} < \theta_{ShotChange}$ 를 함께 두어 경계 검출 오류를 줄였다. 현재의 그라운드 블락 맵을 세 프레임 이전의 것과 비교하였는데, 이는 페이드 인/아웃이나 시각적 효과(visual effect)로 인해 샷이 점진적으로 변하는 경우에 대응하기 위해서이다.

2.1.4 샷 종류 판정

샷 경계가 검출되면 그 지점에서 샷 종류 판정이 수행되어야 한다. 원거리 샷과 그렇지 않은 샷을 구분해내기 위해 우리는 그라운드 블락 맵을 이용하였다.

먼저 i 번째 블락열에 대해 가장 긴 녹색 세그먼트(Longest Green Segment), LGS_i 를 찾아낸다. 이제 각 영상 프레임에서 [8]에서 말한 황금 영역 내 LGS의 길이를 측정하여 샷의 종류를 알아낼 수 있다. θ_L 보다 작은 LGS가 하나라도 존재하면 그 샷을 원거리 샷이 아닌 것으로 판정한다. 반대의 경우에는 원거리 샷으로 판단한다.

$$Class(f_k) = \begin{cases} \text{Non-long-shot, if } |LGS_i| < \theta_L \text{ for } GSLeft \leq i \leq GSRight \\ \text{Long-shot, otherwise} \end{cases} \quad (7)$$

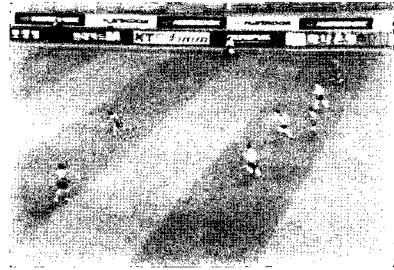
여기서 $\theta_L = BlocksInColumn/3$, $GSLeft = BlocksInRow \times 3/11$, and $GSRight = BlocksInRow \times 8/11$ 로 정하였다.

2.1.5 픽셀 단위 영상 분할

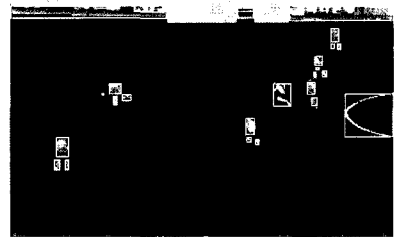
관심영역 창의 지정은 공의 위치와 관계된다. 공을 찾기 위해서 우리는 식 (8)을 이용하여 픽셀 단위 이진화 영상을 구축하였다.

$$BinaryMap(x, y) = \begin{cases} 0, \text{ if } Ground(x, y) = 1 \\ 1, \text{ otherwise} \end{cases} \quad (8)$$

그림 5(b)는 이진화 영상을 보여주고 있다. 이 과정에서는 그라운드 블락이 아닌 블락은 제외시켰다. 그런 후



(a)



(b)

그림 5(a) 이진화 영상. (b) 영역 채색 기법을 통해 검출한 물체의 최소 경계 사각형을 표시하였음

프레임 내에 존재하는 물체들의 목록을 얻기 위해 8방향의 영역 채색 기법을 이용하였다.

2.1.6 물체 타입 결정

물체들은 식 (9)에서와 같이 'BALL_CANDIDATE' 또는 'OTHERS'로 명명된다.

$$\begin{cases} \text{object_type} = \text{BALL_CANDIDATE,} \\ \quad \text{if } \left\{ \begin{array}{l} 3 < M < 20 \text{ AND} \\ \left(1.5 < \frac{\text{width} \times \text{height}}{M} \text{ OR } \text{tb_diff1} \leq \frac{\text{width}}{2} \text{ AND } \text{tb_diff2} \leq \frac{\text{width}}{2} \right) \text{ AND} \\ 0.6 \leq \text{aspect_ratio} \leq 1.7 \text{ AND} \\ \text{Intensity} > \text{ValuePeakIndex} \end{array} \right\} \\ \text{object_type} = \text{OTHERS, otherwise.} \end{cases} \quad (9)$$

여기서 M 은 물체의 픽셀 수이고 $width$ 와 $height$ 는 각각 물체의 가로와 세로를 나타낸다. $aspect_ratio$ 는 가로를 세로로 나눈 비율이며, $Intensity$ 는 물체의 평균 명도를 나타낸다. 두 번째와 세 번째 식은 물체가 둥근 물체인지 아닌지를 나타내는데, tb_diff1 는 물체의 왼쪽 위 점의 x 좌표와 오른쪽 아래 점의 x 좌표의 차의 절대값을 나타낸다. tb_diff2 는 물체의 오른쪽 위 점의 x 좌표와 왼쪽 아래 점의 x 좌표의 차의 절대값을 말한다. 그림 5(b)는 물체들의 최소 영역 사각형(Minimum Bounding Rectangle, MBR)을 통해 얻어낸 물체들을 노란색을 사용해 나타난 이진화 영상이다.

2.1.7 공 찾기

물체목록을 얻고 물체들을 구별할 수 있지만 한 프레임 안에서 어느 물체가 공인지 구별하는 것은 쉬운 일

이 아니다. 그 이유는 한 프레임 내에는 공과 비슷한 물체들 즉, 선수들의 양말이나 그라운드 라인 조작 등이 많기 때문이다. 게다가 공 모양은 찌그러지거나 선수들에 의해 가려지기도 한다. 때로는 공이 영상 내에 존재하지 않기도 한다. 공과 선수를 찾는 많은 연구들이 있어왔다[13-15]. 하지만 [13]에서 제시한 방법은 공의 처음 위치를 수동으로 지정해야 하는 단점이 있었다. [14]에서는 고정된 카메라로 촬영한 축구 영상을 이용하였기 때문에 우리의 경우에는 맞지 않다. [15]는 공의 궤적을 얻기 위해 이후 프레임의 정보를 필요로 하므로 실시간으로 동작할 수 없다는 단점이 있다. 우리는 간단하고 이전 값에만 의존하여 방송 영상을 실시간으로 처리할 수 있는 방법을 제안한다.

비터비(Viterbi) 알고리즘은 최적의 경로를 찾는 데 가장 많이 쓰이는 알고리즘이다[16]. 우리는 이 알고리즘을 볼의 자취를 찾는 데 이용한다. 이 과정은 그림 6으로 나타낼 수 있고 다음과 같이 설명될 수 있다.

$P_t(j)$ 를 frame t 에서의 j 번째 후보에서 끝나는 최적의 경로라고 하자. $\delta_t(j)$ 는 $P_t(j)$ 을 따라 노드와 가장자리 가중치를 합한 값이고, N_t 는 frame t 에서의 후보의 총 개수이고, $\{q_t\}_{t=1, \dots, T}$ 은 하나의 최적의 상태열이다. 시간 $t-1$ 에서의 $\delta_{t-1}(i)$ 에서 시간 t 에서의 $\delta_t(j)$ 로의 이동은 가지거리(branch metric) $d(\mathbf{z}_i^{t-1}, \mathbf{z}_j^t)$ 로 나타내어지는 비용이다. 여기서 $\mathbf{z}_j^t = (x_j^t, y_j^t)$ 으로 프레임 t 에서의 j 번째 공의 후보의 한 가운데 점의 좌표이다. 가지거리는 city block 거리[17] 즉, $d(\mathbf{z}_i^{t-1}, \mathbf{z}_j^t) = |x_i^{t-1} - x_j^t| + |y_i^{t-1} - y_j^t|$ 으로 나타낼 수 있다. 공은 인접한 프레임간에 많은 거리를 이동하지 않는다는 가정 아래 $d(\mathbf{z}_i^{t-1}, \mathbf{z}_j^t)$ 가 지정된 값 d_{max} 보다 작도록 하였다. 우리 실험에서는 352x240크기 영상에서 d_{max} 를 20으로 정했다. 영상내의 점수상자나 로고 같은 물체를 공으로 오인하는 경우를 방지하기 위해 T 개의 연속된 프레임동안 물체의 위치가 변하지 않을 때 $\delta_T(j)$ 를 $5 \times d_{max}$ 로 정했다. 이전 T 개의 연속프레임을 이용해 매 프레임마다 공의 경로를 계산하며, 우리의 실험에서는 T 를 6으로 설정하였다. 이전 절에서 얻은 물체들 중 오직 타입이 'ball candidate', 즉, object type = BALL_CANDIDATE인 물체들만 공 후보 노드로 설정된다. 만약 프레임 내에 공의 후보가 없다면 볼의 위치는 이전 프레임의 공의 위치로 설정되며, 세 프레임 이상 공이 찾아지지 않을 경우 영상의 정 중앙을 공의 위치로 설정했다.

1. $\delta_t(i) = 0, 1 \leq i \leq N_t$
2. $\delta_t(j) = \min_{1 \leq i \leq N_{t-1}, d(\mathbf{z}_i^{t-1}, \mathbf{z}_j^t) \leq d_{max}} [\delta_{t-1}(i) + d(\mathbf{z}_i^{t-1}, \mathbf{z}_j^t)], 2 \leq t \leq T$
3. $q_T = \arg \min_{1 \leq i \leq N_T} [\delta_T(i)]$

그림 6 비터비 알고리즘을 기반한 볼 위치추정

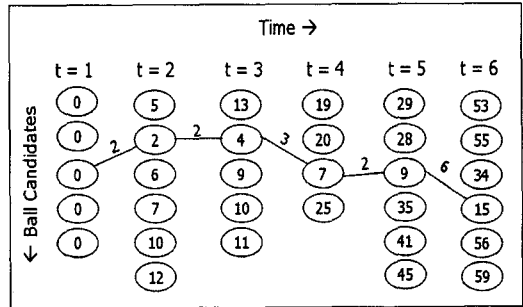


그림 7 가지와 상태거리의 격자구조의 예. 상태거리와 최적의 경로가 실선으로 표시되어 있다.

볼의 위치가 인접한 프레임에서 큰 변화를 보이지 않는 것은 쉽게 알 수 있다. 그래서 2단계 탐색을 실행한다. 즉, 이전 프레임의 공의 위치를 중심으로 한 일정 영역에서 후보를 탐색하고, 후보가 찾아지지 않을 경우 나머지 영역을 탐색하는 것이다. 이 2단계 탐색은 전체 프레임 탐색대비 19.31%의 시간 향상효과를 가져왔다. 실험 결과 1단계 탐색영역의 크기는 프레임의 가로 세로 길이의 각 1/3 크기가 가장 효율적인 것으로 조사됐다.

2.1.8 관심영역 창 지정

관심영역 창을 배치하는 가장 간단한 방법은 관심영역 창의 중심을 공의 위치에 놓는 것이다. 하지만 여러 프레임에 걸쳐 관심영역 창의 중심이 공을 따라가는 경우 화면이 흔들리거나 사용자가 보기에 너무 빠르게 움직일 가능성이 있어 창이 부드럽게 움직이도록 해야 한다. 또한 공이 빠르게 움직이는 경우를 감안하여 디스플레이 창을 움직이는 방법에는 가속도의 개념이 포함되어야 한다.

관심영역 창은 공의 위치를 창의 중심에 두므로 공이 프레임의 가장자리에 위치할 경우 관심영역 창이 프레임을 벗어날 수 있다. 관심영역 창이 프레임을 벗어나지 않게 제한시키기 위해, 관심영역 창이 프레임을 벗어나더라도 디스플레이 되는 것은 프레임 내로 제한시켰다. 더 구체적인 내용은 [9]를 참고 바란다.

만약 공과 비슷한 물체로 인하여 공의 위치가 오판될 경우, 관심영역이 잘못 지정될 수 있다. 또한 사용자가 공의 주변이 아닌 화면의 다른 부분을 보고자 할 경우도 있다. 이러한 경우 사용자는 직접 관심 영역을 지정하고 싶을 것이다. 즉, 공 찾기 오류를 보완하고 사용자

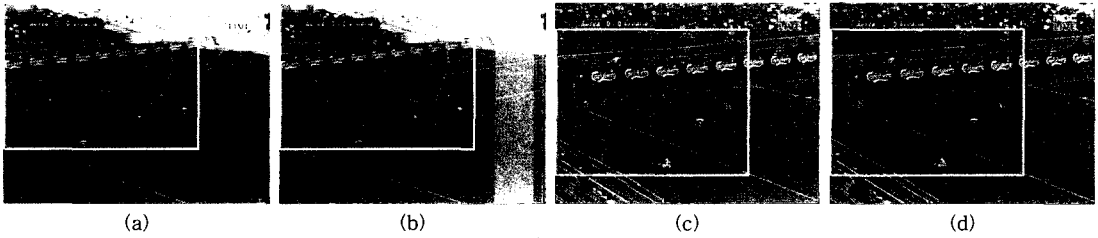


그림 8 공의 위치가 분홍색 X자 표시되어있다. (a) 220th 프레임. 공 찾기에서 오류를 보이고 있음. (b) 221st 프레임. 사용자가 직접 공의 위치를 지정함. (c) 320th 프레임. 100프레임 동안 지정된 위치가 공의 위치로 지정됨. (d) 321st 프레임. 100프레임 이후 다시 공 찾기 프로세스가 시작됨

의 직접적인 관심영역 지정이 필요하다. 이를 위해 사용자가 원하는 지점을 찍을 경우 찍은 시점부터 100 프레임 동안은 관심영역이 그 지점을 가리키도록 했다. 물론 100프레임 이후에는 다시 공 찾기 프로세스가 시작된다.

2.2 점수상자 추출 및 디스플레이

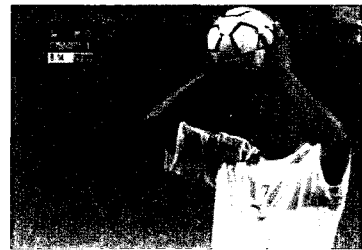
이동형 단말기로 축구경기 영상을 시청할 때, 사용자가 느끼는 불편함은 위에서 설명한 작은 물체들을 포함하는 원거리 샷의 경우 외에도 주로 화면의 상단 귀퉁이에 위치하는 작은 점수상자에도 기인하는 바가 크다.

축구 경기에서 점수상자는 경기 상황을 파악하는 데 매우 중요한 요소인 경기 진행 시간과 점수를 포함하고 있다. 따라서 사용자는 경기에 대한 정보를 얻기 위해 점수상자를 보기 원한다. 하지만 앞에서 언급했던 바와 같이 서비스되는 영상이 단지 소형 이동형 단말기의 화면 크기에 맞게 축소되어 디스플레이 되기 때문에 영상 정보의 손실이 발생하게 된다. 예를 들면, 그림 9에서 볼 수 있는 바와 같이 이동형 단말기의 경우에는, 점수상자가 작아서 현재 상황을 알 수 없는 경우나 현재 영상 내에 점수상자가 없는 경우 등 사용자가 불편을 느끼는 경우가 발생한다. 이러한 문제점을 극복하기 위해 효율적인 점수상자 추출 및 확대가 필요하다.

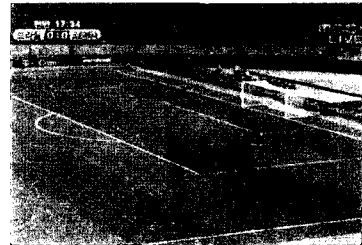
그동안 광고판 영역에 대한 연구는 몇몇 있어 왔으나 점수상자 추출에 대한 관련 연구는 활발히 이루어지지 않았다. Okihisa[18] 등은 에지(Edge) 정보를 이용하여 그라운드 영역과 캡션(Caption) 영역을 구별하고 있지만 캡션 영역 내에서 점수상자와 광고판을 구별하기 쉽지 않은 문제점 있다. Yoon[19] 등은 광고판의 수평방향으로의 밝기 변화가 그라운드와 다르다는 사실을 이용하여 광고판 영역을 구별하고 있지만 다양한 디자인의 광고판과 점수상자가 존재하기 때문에 밝기의 차이를 이용해 이 둘을 구별하기는 쉽지 않다. 따라서 점수상자의 특성만을 이용하여 점수상자를 추출하는 것이 바람직하다.

2.2.1 점수상자 추출

2.2.1.1 한 프레임 내에서 점수상자 경계 좌표 추출



(a)



(b)

그림 9 (a), (b) 축구 경기의 다양한 점수상자. 점수상자의 내용을 확인하기가 쉽지 않음

점수상자는 일정한 밝기를 가진 경계선을 가지고 그 안에 점수나 시간을 표현하는 문자를 포함하기 때문에 Markov source의 한 형태인 자연영상과 달리 이웃 픽셀간의 밝기 차이가 급격히 변한다. 따라서 수평, 수직 방향으로 밝기 차이들의 합이 지정된 임계값(threshold) 보다 커지는 순간을 점수상자의 경계 좌표로 인식할 수 있다. 즉, 수평, 수직방향 측면에 대해 *sum of pixel difference (SPD)*를 수평방향으로는 (10), 수직방향으로는 (11)와 같이 정의한다.

$$SPD_v(y) = \sum_{x=0}^{width/2} |f(x,y) - f(x+1,y)| \quad (0 \leq y < \frac{height}{3}) \quad (10)$$

$$SPD_h(x) = \sum_{y=0}^{height/3} |f(x,y) - f(x,y+1)| \quad (0 \leq x < \frac{width}{2}) \quad (11)$$

위 식에서 우리는 추출 성능 향상을 위해 점수상자는 화면가로의 1/2, 세로의 1/3 안에 해당하는 영역에 위치

하는 것으로 가정한다. 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. 즉, 수평, 수직 방향 측면에 대해 *difference of SPD(DSPD)*를 각각 식 (12), (13)와 같이 정의한다.

$$DSPD_h(x) = |SPD_h(x) - SPD_h(x+1)| \quad (0 \leq x < \frac{width}{2}) \quad (12)$$

$$DSPD_v(y) = |SPD_v(y) - SPD_v(y+1)| \quad (0 \leq y < \frac{height}{3}) \quad (13)$$

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. 실험에 사용한 모든 비디오에 대해 임계값으로 200을 사용하였다. 그림 10은 그림 9(a)에 대하여 왼쪽 상단의 x, y 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, y좌표가 (29, 21)로 정해짐을 알 수 있다. 따라서 점수상자 추출을 위한 왼쪽 상단의 좌표는 (29, 21)이 된다. 동일한 방법으로 오른쪽 하단의 좌표도 추출한다.

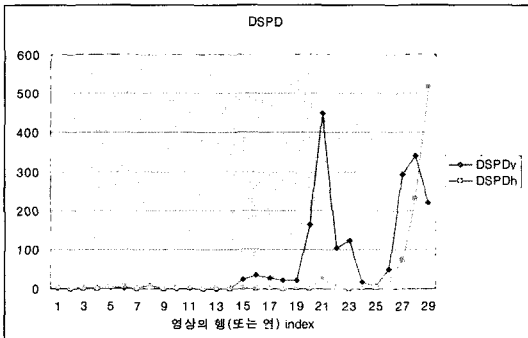
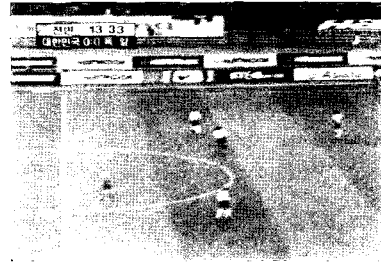
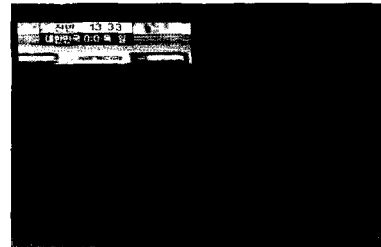


그림 10 수평, 수직 방향에 대한 DSPD

그림 11은 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. 따라서 일정 프레임 구간에 대한 학습을 통해 최적의 경계 좌표를 결정할 필요가 있다.



(a)



(b)

그림 11 (a) 축구 경기 영상. (b) 제안하는 알고리즘을 이용하여 한 프레임에 대해 추출된 점수상자 후보 영역

2.2.1.2 학습을 이용한 최적의 경계 좌표 결정

축구 경기에서 역동적으로 변하는 주변환경 때문에 일반적으로 한 프레임 안에서 추출된 점수상자 영역은 정확하지 않다. 점수상자 주변에 광고판이나 관중석이 존재 할 때 SPD와 DSPD 값만을 이용하여 점수상자 인지 아닌지 구별하기가 쉽지 않기 때문에 추출된 점수상자 영역은 더 많은 주변 픽셀들이 포함 되게 된다. 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 y값을 가지는 좌표가 필요하고, 점수상자의 오른쪽 하단 경계 좌표를 위해 추출된 후보군 중 최소 x, 최소 y값을 가지는 좌표가 필요하다는 사실을 알 수 있다. 따라서 일정 프레임 구간에 대한 학습을 통해 위의 조건을 만족하는 경계 좌표를 결정한다. 이를 위해 식 (14)와 같은 간단한 알고리즘을 사용하였다.

```

(i : frame number, learning_period : 200 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);
    
```

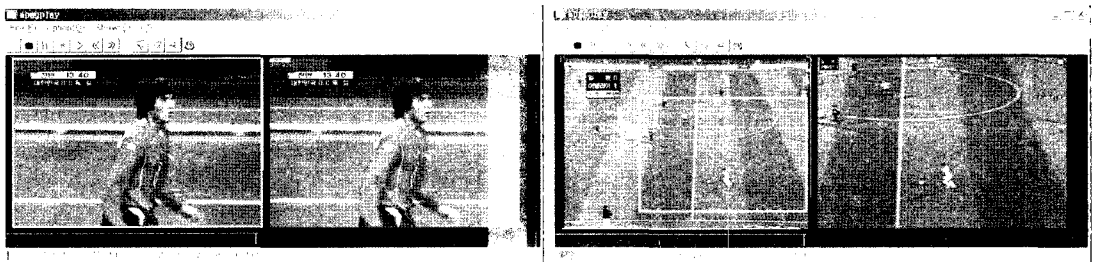
(14)

학습 구간으로 200 프레임을 사용하였다. 200 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 점수상자 영역을 디스플레이 하게 된다.

3. 실험 결과

이 시스템은 Win32 환경에서 Visual Studio 2003 (C++)를 이용하여 개발하였으며 MPEG 복호화를 위해 FFMpeg 라이브러리를 이용하였다. 우리는 352×240 해상도, 29.97fps의 MPEG1 포맷으로 부호화된 2분 길이 축구 영상 네 개(한국 대 독일: 2004년 평가전, 일본 대 이탈리아: 2004 아테네 올림픽, 브라질 대 크로아티아 : 2006 월드컵, 한국 대 코스타리카: 2006년 국가대표평가전)를 이용하여 실험하였다. 모든 실험은 Pentium-4 3.0GHz PC에서 실시간으로 수행되었다. 관심영역 결정을 위한 파라미터는 $\alpha, \alpha_1, \alpha_2$ 에 대해 각각 0.85, 0.05, 0.15로 설정하였다([9] 참조). 또한 관심영역 창 크기는 [3]에서의 실험결과를 참조하여 352240의 크기를 갖는 원래 영상에 대해 수직, 수평방향에 대해 1:0.68의 비율로 축소된 240×164 크기의 관심영역을 추출하였다. 아래 그림 12는 우리 실험 프로그램의 스크린 샷이다.

3.1 관심 영역의 결정



(a) non-long shot frame

(b) long shot frame

그림 12 실험 결과. 모든 과정은 자동적이며 실시간으로 수행된다. 원 영상 위에 노란색의 관심영역 창이 놓여있고 관심영역이 디스플레이 크기에 맞게 확대 되어있다. (a) 비원거리 샷 (b) 원거리 샷

3.1.1 그라운드 검출 성능의 평가

제 3장에서 제안된 그라운드 검출 성능을 검증하기 위해 10개의 서로 다른 영상을 선택하였다. 영상의 그라운드 색상은 비디오마다 충분히 다르다. 각 영상마다 임의적으로 2개의 프레임을 선택하여 각 프레임으로부터 두 개의 조각을 추출한다. 두 조각 중 하나는 그라운드 영역으로부터, 나머지 조각은 비그라운드 영역으로부터 추출한다. 조각의 크기는 160×80이며 한 조각에는 16×16크기의 50개의 블록이 들어있게 된다. 따라서 16×16크기의 그라운드 블록 1000개(=50×2×10)와 1000개의 비그라운드 블록을 가지고 실험하였다. 실험 결과, 1000개의 그라운드 블록 중 970개가 그라운드 블록으로 검출되었으며 1000개의 비그라운드블록 중 42개가 그라운드로 검출되었다. 다시 말하면, 거짓 음성률(false-negative rate)은 0.030이고 거짓 양성률(false positive rate)은 0.042이며 식 (15)와 같이 나타낼 수 있다.

$$\begin{aligned}
 \text{false negative rate} &= \frac{\text{number of false negatives}}{\text{number of positives}} \\
 \text{false positive rate} &= \frac{\text{number of false positives}}{\text{number of negatives}} \quad (15)
 \end{aligned}$$

3.1.2 샷 분류 과정에 대한 평가

실험 결과를 표 1에 나타내었다. 표에서 알 수 있는 바와 같이 평균 90%가 넘는 정확도를 보이고 있다. 표에는 우리의 이전 알고리즘인 [9]과 비교되어 있다. 제안된 알고리즘에서는 블록의 대표점만을 이용하여 블록의 그라운드 속성을 구함에도 불구하고, 이전결과에 못지 않은 샷 분류성능을 보이고 있다. 개선된 그라운드

표 1 제안된 샷 분류 기법에 대한 정확도

| | Prior Algorithm [9] | | | Proposed Algorithm | | |
|----------------------|---------------------|--------------|-------|--------------------|--------------|-------|
| | Long shot | Non-longshot | Total | Long shot | Non-longshot | Total |
| Japan vs. Italy | 93.4% | 83.1% | 90.7% | 95.2% | 94.3% | 95.0% |
| Korea vs. Germany | 97.8% | 76.1% | 88.4% | 96.8% | 73.4% | 86.7% |
| Brazil vs. Croatia | 99.6% | 83.7% | 93.9% | 99.6% | 83.7% | 93.9% |
| Korea vs. Costa Rica | 98.0% | 63.7% | 83.6% | 98.9% | 74.6% | 88.7% |

픽셀 검출 개념(식 (1)과 (2)참조)과 2.1.2절의 홀 채우기 기법으로 16×16블락의 모든 픽셀을 계산하는 대신 한 가운데 점만을 계산하는 개념이 성능저하를 보상할 수 있었다.

샷 분류와 그라운드 블락 맵 구축, 그리고 샷 경계검출 과정에 소요되는 시간을 비교해 보면, 이전 알고리즘 [9]이 8,396μsec 이 소요되는 반면 새로운 알고리즘은 58μsec이 소요되었다. 새로운 알고리즘이 약 145배 더 빠른 것을 알 수 있다.

3.1.3 관심영역 결정 과정에 대한 평가

결정된 관심영역을 평가하는 객관적인 기준에 대해서는 말하기 힘들다. 하지만 원거리샷일 동안에는 공은 대부분 관심영역 안에 존재해야 한다고 생각할 수 있다. 공이 오랫동안 관심영역 안에 있지 않을 경우 시청자들은 이 관심영역 기반의 디스플레이를 더 이상 사용하기 원치 않을 것이다. 그래서 바로 이 점이 관심영역 결정 성능의 평가의 중요한 요소가 될 수 있다. 그래서 관심영역이 결정된 프레임 수를 세고 그 중 불이 관심영역 안에 존재하는 프레임이 얼마나 존재하는지를 세었다. 제안된 시스템에서의 관심영역 창 크기는 352×240영상의 경우 240×164이다. 표 2에서 볼 수 있듯이, 공이 관심영역 창 안에 상당히 많이 존재함을 알 수 있다. 공이 선수들에 의해 가려지거나 프레임 안에 아예 존재하지 않는 경우는 배제했다. 그림 12에서 볼 수 있듯이, 관심영역 창은 원영상이 근거리 샷이거나 중거리 샷일 경우는 원영상과 같게 설정되고, 원거리 샷일 때는 원영상의 일부분이 된다.

표 3은 공의 위치와 관심영역 창과의 관계를 더욱 구체적으로 보여준다. 우리는 같은 영상을 사용하였다. 'Ball in ROI'은 공이 관심영역 창안에 존재하는 경우이고 'Ball out of ROI but in the frame'은 제안한 알고

리즘이 공을 찾지 못한 경우이다. 'Ball out of frame'은 원영상 안에 공이 존재하지 않는 경우로 관심영역을 공에 두는 것이 불가능한 경우이다. 이 데이터 셋은 관심영역 디스플레이 모드가 실행되는 원거리 샷 프레임의 경우에만 실행되었다.

3.1.4 PDA에서의 성능 평가

우리는 실제 모바일 기기에서 이 시스템의 성능을 측정하기 위해 PDA용으로도 개발하였다. 개발에는 Microsoft Embedded Visual Studio 4.0을 이용하였다. PDA의 스펙은 표 4에 나타내었다. 성능 측정을 위해 H.264/AVC 포맷의 2004년 아테네 일본 대 이탈리아 올림픽 경기 120초 길이의 축구 영상을 사용하였다. 세 번 반복 실험하여 걸리는 시간의 평균을 측정하였다. 표 5는 원거리샷과 비원거리샷을 실행하는데 걸리는 시간을 나타낸다. 비원거리 샷일 때보다 원거리 샷일 때 시간이 더 걸리는 것은 당연한데, 원거리샷일 경우에는 추가적인 관심영역 결정과정이 필요하기 때문이다.

표 4 테스트에 이용한 PDA의 사양

| | |
|-----------|-------------------------|
| Model No. | HP iPAQ hx4700 |
| CPU | Intel PXA270 624MHz |
| RAM | 64 MB |
| screen | VGA (640×480), 4.0 inch |

이전의 알고리즘과 제안된 알고리즘 적용 시의 관심영역 분석 및 평균 재생 속도 비교를 표 6에 나타내었다.

그림 13은 이전의 알고리즘[9]에 대해서 새로운 알고리즘의 속도 개선 정도를 보여주는 그래프이다. 전체 프로그램 수행 시간을 비교해보았을 때 약 45%의 속도 개선이 나타난 것을 확인 할 수 있다.

표 2 관심영역 결정 성능 평가

| | Hit-rate (the ball inside ROI) | |
|----------------------|--------------------------------|--------------------|
| | Prior Algorithm [9] | Proposed Algorithm |
| Japan vs. Italy | 70.47% | 91.35% |
| Korea vs. Germany | 80.13% | 87.59% |
| Brazil vs. Croatia | 64.80% | 70.59% |
| Korea vs. Costa Rica | 77.83% | 81.32% |

표 3 공의 위치와 관심영역 창의 관계

| | Japan Vs. Italy | Korea Vs. Germany | Brazil vs. Croatia | Korea vs. Costa Rica |
|--|-----------------|-------------------|--------------------|----------------------|
| Ball In ROI | 2,113 (83.29%) | 2,428 (79.04%) | 1,680 (70.44%) | 914 (76.29%) |
| Ball out of ROI but in the frame | 200 (7.88%) | 344 (11.19%) | 700 (29.35%) | 210 (17.53%) |
| Ball out of frame | 224 (8.83%) | 300 (9.77%) | 5 (0.21%) | 74 (6.18%) |
| Total number of Long shot frames in the sequence | 2,537 (100%) | 3,072 (100%) | 2,385 (100%) | 1,198 (100%) |

표 5 PDA에서 지능형 디스플레이를 재생할 때 걸리는 시간

| | | # of frames, N | Elapsed time(μs), T | T/N (μs) |
|---|--------------------------------|----------------|---------------------|----------|
| Analysis for Long shot | Total | 2562 | 11,884,840(=TLS) | 4639 |
| | Ball detection | 2562 | | |
| | 2 nd ball detection | 498 | | |
| | Shot classification | 2562 | | |
| Analysis for Non-long shot | Total | 1038 | 1,292,450(=TNL) | 1245 |
| | Shot classification | 1038 | | |
| Entire processing time (including H.264 decoding and display) | | 3600 | 137,498,267 | 38194 |

표 6 관심영역 분석에 걸린 시간 및 평균재생시간

| | Prior Algorithm [9] (fps) | Proposed Algorithm (fps) |
|--|---------------------------|--------------------------|
| Average processing rate for analysis (=1000,000/((T _{LS} +T _{NL})/N)) | 33.9 | 273.29 |
| Average playing rate | 14.4 | 26.355 |

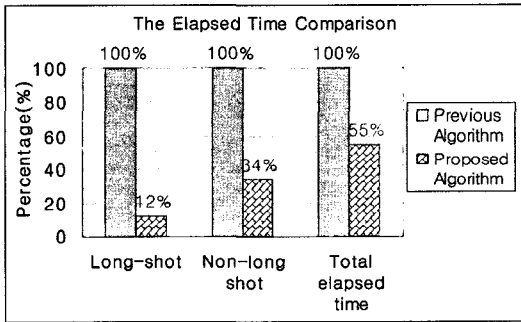


그림 13 이전 알고리즘[9]과 제안된 알고리즘의 실행시간 비교

3.2 점수상자 추출

다양한 영상에 대해 적용한 결과는 그림 15와 같다. 200 프레임 구간의 학습을 통해 추출된 점수상자 영역이 2배 확대되어 디스플레이 되고 있다.

추출의 정확성을 측정하기 위해 다음과 같이 정의되

는 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 영역보다 작다면 Precision 값이 커지기 때문에 Recall과 Precision을 동시에 살펴야 한다. 4개의 축구 경기 비디오를 이용하여 Recall과 Precision을 구하였으며, 그 결과는 그림 16과 표 7, 8에 자세히 나타내었다(영상 1 : 브라질 대 크로아티아, 영상 2 : 일본 대 이탈리아, 영상 3 : 대한민국 대 독일, 영상 4 : 대한민국 대 코스타리카).

그림 16에서 사각형 형태를 가지는 점수상자에 비해 영상 1과 4와 같이 점수상자가 사각형 형태가 아닌 경우 Recall과 Precision값이 떨어짐을 확인 할 수 있었다.

이 프로그램 역시 PDA용으로 개발하여 3.1.4에서 사

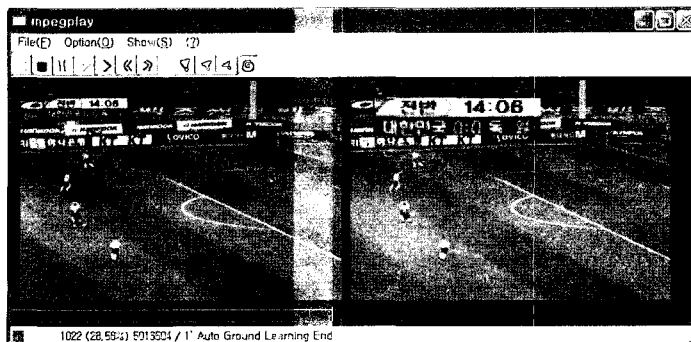


그림 14 실험 결과. 축구 경기 영상(왼쪽 프레임샷). 일정 프레임 구간 동안의 학습이 끝나고 최적의 점수상자 영역을 추출한다. 사용자가 알아보기 쉽게 2배 확대하여 디스플레이 할 수 있다(오른쪽 프레임샷).

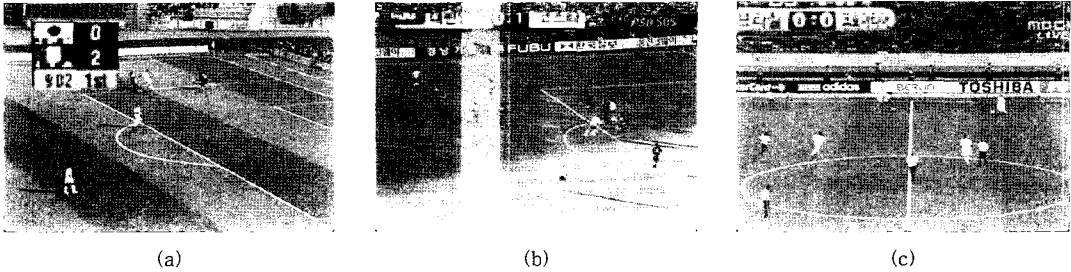


그림 15 (a) 사각형 형태의 점수상자. (b), (c) 사각형 형태가 아닌 점수상자. 점수상자 추출을 위해 두 개의 좌표만을 이용하기 때문에 추출된 영역이 다소 부정확하다.

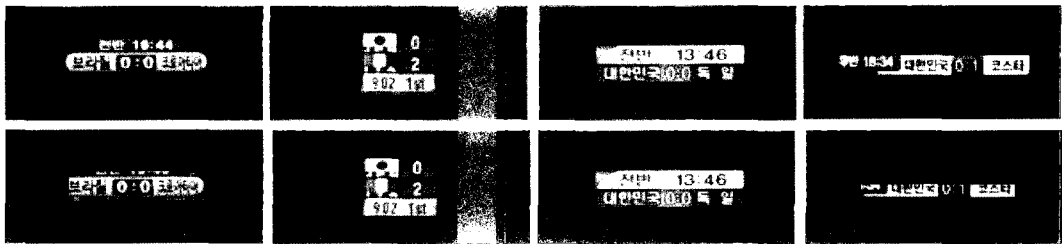


그림 16 실제 점수상자 영역(위쪽 : 왼쪽에서부터 오른쪽으로 각각 영상 1, 영상 2, 영상 3, 영상 4). 제안된 알고리즘에 의해 추출된 점수상자 영역(아래쪽)

표 7 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 8 Recall과 Precision을 측정된 결과

| | Recall | Precision |
|------|--------|-----------|
| 영상 1 | 84.14 | 83.69 |
| 영상 2 | 90.04 | 100 |
| 영상 3 | 100 | 100 |
| 영상 4 | 51.61 | 100 |

용한 PDA를 이용하여 성능을 측정하였다. 성능 측정을 위해 H.264/AVC 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 [9]에서 사용한 식을 이용하여 성능 측정된 결과를 표 9에 정리하였다.

점수상자가 없는 프레임에서도 점수상자를 디스플레이 하기 위해서 몇 프레임 앞서 표시된 점수상자를 저장한다. 본 논문에서는 점수상자의 주변 환경이 정적인 근거리 샷 프레임의 점수상자 영역을 저장한다. 저장한

표 9 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 |

점수상자 영역과 3.2절의 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. 그림 17(b)는 저장한 점수상자를 확대하여 디스플레이하고 있고 그림 17(c)에서는 3.1절의 관심영역 확대와 함께 점수상자를 확대하여 디스플레이하고 있다.

4. 결론

우리는 축구 비디오의 원거리 샷을 검출하고 관심영역을 지정하여 확대 재생하는 지능형 디스플레이 시스템을 제안하였다. 제안한 알고리즘이 제대로 동작하는 것을 보이기 위해 우리는 여러 개의 국가간 축구 경기 비디오 클립을 이용하여 테스트 하였다. 이전 장에서 보인 것처럼 제안한 시스템은 일반 PC 상에서 실시간으로 동작하였으며 PDA에서 평균 25.8 fps의 속도를 보였다.

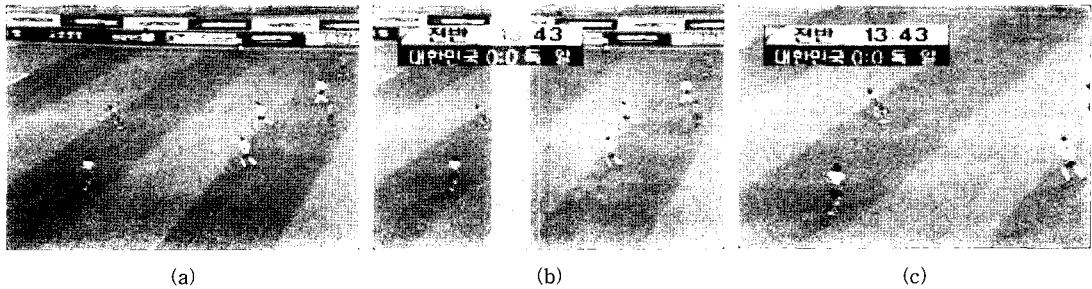


그림 17 (a) 점수상자가 없는 프레임. (b) 저장한 점수상자를 확대하여 디스플레이. (c) 관심영역과 저장한 점수상자를 동시에 확대하여 디스플레이

또한, 이동형 단말기 시청자의 시청편리성을 향상시키기 위한 또 다른 노력으로서 점수상자를 자동으로 추출하여 확대하는 방법을 제안하였다.

H.264/AVC 포맷의 영상에서는 디코딩 시 디코딩 정보를 제공하므로 이 정보를 이용하여 현재와 같이 디코딩 후 픽셀 도메인에서 접근하는 것이 아니라 압축 도메인에서 접근하는 방법도 더욱 신속하고 효율적인 처리를 위해 연구할 가치가 있다.

제안하는 시스템은 [20-23]에서와 같이 H.264/SVC 또는 MPEG-21 DIA에서 관심영역 기반의 인코딩과 연계될 수 있어 디코더 측에서만 아니라 인코더 측에서도 결합될 수 있을 것이다.

참 고 문 헌

- [1] <http://en.wikipedia.org/wiki/DMB>
- [2] H. Knoche, J. D. McCarthy, and M. A. Sasse, "Can small be beautiful?: assessing image resolution requirements for mobile TV," in *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 829-838, ACM Press, (New York, NY, USA), 2005.
- [3] 김창익, 고재승, 안일구, 이재호, 서기원, "이동형 단말기에서의 축구경기 시청을 위한 해상도 및 관심 영역 크기에 관한 사용자 만족도 조사," *Journal of the Korean society of broadcast engineers(한국방송공학회)*, vol.11, no.3, pp.336-348, Sep. 2006.
- [4] Hao Liu, Xing Xie, Wei-Ying Ma, and Hong-Jiang Zhang, "Automatic browsing of large pictures on mobile devices," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pp.148-155, ACM Press, (New York, NY, USA), 2003.
- [5] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no.11, pp.1254-1259, Nov. 1998.
- [6] W. H. Cheng, W. T. Chu, and J. L. Wu, "A visual attention based region-of-interest determination framework for video sequences," *IEICE Transactions on Information and Systems*, E-88D, pp.1578-1586, 2005.
- [7] Y. F. Ma and H. J. Zhang, "A model of motion attention for video skimming," in *Proc. ICIP*, pp. 129-132, 2002.
- [8] X. Fan, X. Xie, H. Q. Zhou, and WY Ma, "Looking into Video Frames on Small Displays," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pp. 247-250, ACM Press, (New York, NY, USA), 2003.
- [9] Keewon Seo and Changick Kim, "A Context-Aware Video Display Scheme for Mobile Devices," in *Proc. SPIE Vol.6074, pp.278-287, Multimedia on Mobile Devices II*, San Jose, Jan. 2006.
- [10] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image Processing*, vol. 12, pp.796-807, 2003.
- [11] K. Wan, X. Yan, X. Yu, and C. Xu, "Real-time goal-mouth detection in mpeg soccer video," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pp. 311-314, ACM Press, (New York, NY, USA), 2003.
- [12] R. W. Lienhart, "Comparison of automatic shot boundary detection algorithms," *Storage and Retrieval for Image and Video Databases VII* 3656(1), pp.290-301, SPIE, 1998.
- [13] Y. Seo, S. Choi, H. Kim, and K. Hong, "Where Are the Ball and Players? Soccer Game Analysis with Color Based Tracking and Image Mosaick," in *ICIAP '97: Proceedings of the 9th International Conference on Image Analysis and Processing-Volume II*, pp.196-203, Springer-Verlag, London, UK, 1997.
- [14] Y. Ohno, J. Miura, and Y. Shirai, "Tracking Players and a Ball in Soccer Games," in *Int. Conf. On Multisensor Fusion and Integration for Intelligent Sys.*, Taipei, Taiwan, 1999.

- [15] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. Wan, "Trajectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pp.11-20, ACM Press, (New York, NY, USA), 2003.
- [16] Rabiner L.R., "A Tutorial on Hidden Markov Model and Selected Applications in Speech Recognition," in *Proceedings of the IEEE*, Vol. 77, No. 2, 1989.
- [17] http://en.wikipedia.org/wiki/Taxicab_geometry
- [18] 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, pp.45-48, 2002.
- [19] 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, pp.443-454, 2002.
- [20] J. Lim, M. Kim, J. Kim, and K. Kim, "Semantic transcoding of video based on regions of interest," *Visual Communications and Image Processing 2003*, 5150(1), pp.1232-1243, SPIE, 2003.
- [21] A. Jerbi, Jian Wang, and S. Shirani, "Error-Resilient Region-of-Interest Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 15, pp.1175-1181, 2005.
- [22] N. Doulamis, A. Doulamis, D. Kalogeras, and S. Kollias, "Low Bit-Rate Coding of Image Sequences Using Adaptive Regions of Interest," *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 8, pp.928-934, 1998.
- [23] K. Park, H. Park, "Region-of-interest coding based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 12, pp.106-113, 2002.



김원준

2006년 8월 서강대학교 전자공학과(학사). 2006년 8월~현재 한국정보통신대학교(ICU) 공학부 석사과정. 관심분야는 Object segmentation, Intelligent Display, Color compensation



김창익

1989년 2월 연세대학교 전기공학과(학사) 1991년 2월 포항공과대학교(POSTECH) 전자전기공학과(석사). 1991년 1월~1997년 7월 SKC Ltd. R&D 센터 선임 연구원. 2000년 12월 워싱턴주립대학교 전기공학과(박사). 2000년 12월~2005년 1월

Senior Member of Technical Staff, Epson Palo Alto Laboratory, Epson R&D Inc. 2005년 2월~현재 한국정보통신대학교(ICU) 공학부 조교수



안일구

2005년 2월 전북대학교 전자정보공학부(학사). 2006년 2월~현재 한국정보통신대학교(ICU) 공학부 석/박사 통합과정 관심분야는 Intelligent Display, Color Customization, 3DTV



고재승

2006년 2월 서울시립대학교 전자전기컴퓨터 공학부(학사). 2006년 2월~현재 한국정보통신대학교(ICU) 공학부 석사과정 관심분야는 2D To 3D Conversion