

Customizing Ground Color to Deliver Better Viewing Experience of Soccer Video

Ilkoo Ahn, Youngwoo Kim, and Changick Kim

In this paper, we present a method to customize the ground color in outdoor sports video to provide TV viewers with a better viewing experience or subjective satisfaction. This issue, related to content personalization, is becoming critical with the advent of mobile TV and interactive TV. In outdoor sports video, such as soccer video, it is sometimes observed that the ground color is not satisfactory to viewers. In this work, the proposed algorithm is focused on customizing the ground color to deliver a better viewing experience for viewers. The algorithm comprises three modules: ground detection, shot classification, and ground color customization. We customize the ground color by considering the difference between ground colors from both input video and the target ground patch. Experimental results show that the proposed scheme offers useful tools to provide a more comfortable viewing experience and that it is amenable to real-time performance, even in a software-based implementation.

Keywords: Mobile devices, soccer video analysis, ground color, content personalization, content customization.

I. Introduction

The recent rapid development of multimedia communication technologies and computing power in user terminals makes it possible to provide TV viewers with customized multimedia content. In particular, the customization of mobile content is driven by several limited resources, such as modest processing power, limited battery life, small memory capacity, and increasingly small LCD displays. Unlike other limited resources, for which good solutions have been found, small display is an attribute which is particular to mobile devices. Therefore, novel customization techniques need to be developed to improve viewing experience for mobile users.

The emerging Internet protocol (IP) TV service is expected to allow viewers to personalize the settings of their viewing environment. The personalized and interactive characteristics of IPTV enable customization for individual users.

Customization is meant to fulfill users' satisfaction. Satisfaction can be defined as freedom from discomfort and positive attitudes toward the use of a product. It is a response of users to interaction with the product. Subjective satisfaction comprises subjective responses of users expressing feelings, beliefs, attitudes, or preferences [1].

There have been several approaches to customization: extracting and magnifying the region of interest (ROI) on small LCD panels for mobile device users [2], magnifying the scoreboard for mobile device users [3], and reducing the intensity of shadows cast on the ground in outdoor sports videos [4].

With the advent of multimedia mobile services, users treat mobile devices in an inherently personal way. Examples include a number of different mobile phone models on the market, the wide selection of polyphonic ring tones, personalized ringback tones, and so on. From the viewpoint of

Manuscript received Apr. 08, 2007; revised Nov. 14, 2007.

This research was supported by the MIC, Rep. of Korea, under the ITRC support program supervised by the IITA (IITA-2007-C1090-0701-0017).

Ilkoo Ahn (phone: + 82 42 866 6819, email: ahn19@jcu.ac.kr), Youngwoo Kim (email: ywkim@jcu.ac.kr), and Changick Kim (phone: + 82 42 866 6168, email: ckim@jcu.ac.kr) are with the School of Engineering, Information and Communications University, Daejeon, Rep. of Korea.

video display, color is an essential attribute, which needs to be customized because the viewer's viewing experience or satisfaction may be greatly affected by colors in the image.

Colors are an integral part of our lives, and in some cases, they can influence how we react and feel. For example, colors in the red area of the color spectrum are known as warm colors and include red, orange, and yellow. These warm colors evoke emotions ranging from warmth and comfort to anger and hostility.

Valdez and Mehrabian [5] conducted an experiment regarding the effects of color on the emotions. In their experiments, emotional reactions to color, hue, saturation, and brightness were investigated. The result shows that saturation (S) and brightness (B) have a strong and consistent influence on emotions. Moreover, lighter colors were judged to be friendlier, brighter, and more cultured. They seemed to make life easier, more pleasant and more beautiful, according to [6]. Nonetheless, human feelings are all subjective. Therefore, to improve viewing experience, it should be possible to allow viewers to personalize or customize their viewing conditions.

There have been several studies focused on detecting ground color; however, most of them mainly focus on event detection or summarization [7], [8], or on ground detection to identify players and the ball [9]-[11] and so on [12], [13]. Studies related to color enhancement are generally on natural images rather than sports images.

Many image enhancement algorithms involve histogram techniques. In [14], the successive mean quantization transform called weighted histogram separation (WHS) is used to enhance color images. However, this method changes an entire image, rather than a local area. These approaches are not suitable for our goal, which is to customize ground color. In [15], a combined global/local enhancement technique was proposed to enhance the dark areas of an image with reasonable complexity while maintaining the natural look of the images. The goal of that study, to modify a selected part of an image using its histogram, is similar to our goal. However, the method proposed in [15] may affect neighboring regions as well as the selected region of an image. Therefore, the approach is not suitable for color customization of specific regions. Unlike the previous methods, which are applied to still images, the algorithm proposed in [16] is implemented on video. In [16], a color transformation scheme was proposed, which performs an example-based color stylization of images using perceptual color categories. It adapts the colors of input images to those of a target image. Since its adaptation is carried out on the overall image, this method is not suitable for our goal to customize specific regions in the images.

Figure 1 shows various ground colors in soccer videos, which may result from different ground conditions, light

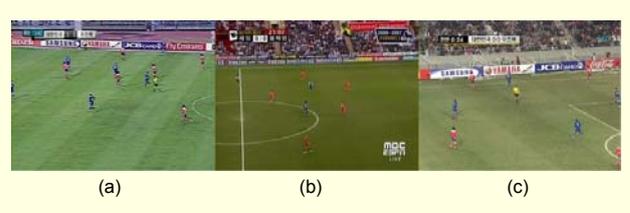


Fig. 1. Examples of ground. The ground colors may not be pleasant to viewers.



Fig. 2. Four kinds of frames observed in soccer video: (a) long-shot frame, (b) mid-shot frame, (c) close-up shot frame, and (d) frame with no ground.

conditions, or even different parameter settings of video encoders. The ground color ranges from yellowish-green to reddish-brown, and it may or may not be pleasant to viewers. In this paper, we propose an efficient algorithm, which provides ground color customization or personalization for soccer video viewers. We believe such customization would raise the viewers' satisfaction.

In this paper, frames are classified into four types of shots according to the amount and location of ground regions as shown in Fig. 2. Long-shot frames are images captured at a long distance; therefore, most parts of a shot tend to be occupied by the ground. For the middle-shot frames, the lower parts of a frame are often taken up by ground. We also divide close-up shots into two groups: close-up with ground and close-up without ground. These four types of shots should be treated appropriately according to ground layout in the image frame.

This paper is organized as follows. In section II, we introduce a ground extraction algorithm and a shot classification algorithm based on our previous work [2]. Then the ground color customization technique is applied to the extracted ground region. Experimental results and the conclusion follow in sections III and IV, respectively.

II. Proposed Algorithm

As mentioned in the preceding section, soccer video contains four different types of shots from the view point of ground layout. For example, in the case of long-shot frames, the ground regions occupy most parts of the shot. This characteristic makes long-shots reliably distinguishable from other shots. Once long-shot detection is performed, the other three types of shots are further classified. After classification of an input frame, the next step is to perform ground region detection. Then, the ground color customization scheme is applied on the detected regions. Figure 3 illustrates the overall process to customize the ground color in a soccer video.

1. Ground Detection

In soccer video analysis, ground color plays an important role, especially in distinguishing shot classes. Because the ground color may vary over different soccer videos and may be affected by camera angles and the existence of shadows, ground detection must be performed for every frame.

Unlike hardware-oriented color models such as RGB, CMY, or YIQ, the HSV color model relates to the perceptual concepts of hue, saturation, and brightness. Moreover, it has been observed that HSV space is better suited for computing color changes and separating hue from saturation, and brightness adds robustness under most lighting variations [17].

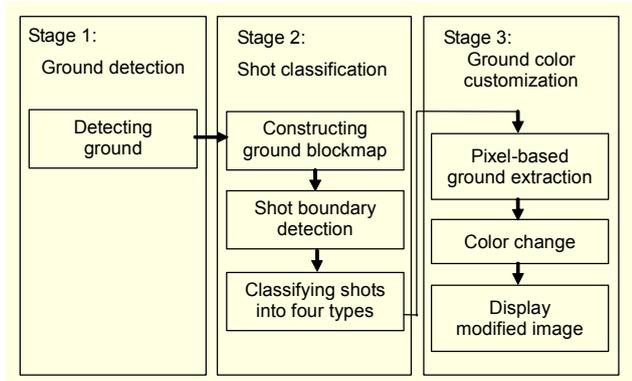


Fig. 3. Block diagram of the proposed algorithm.

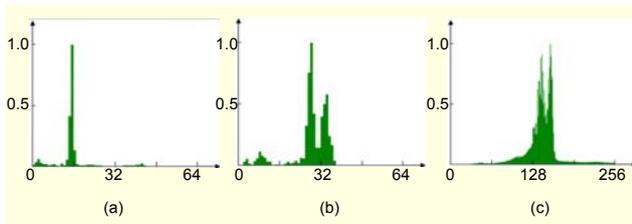


Fig. 4. Histograms for HSV color channels: (a) hue, (b) saturation, and (c) value.

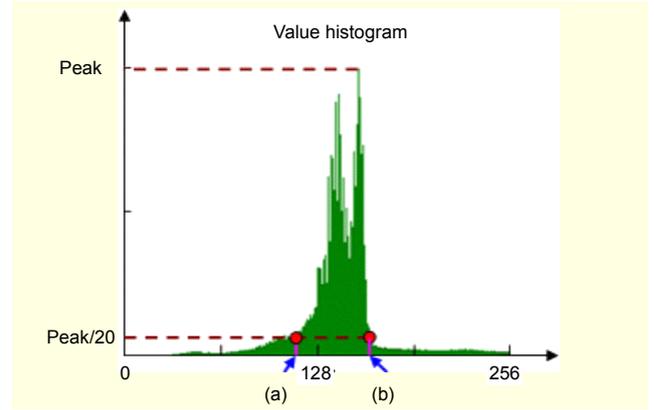


Fig. 5. Finding (a) $LeftBoundary(Hist(V))$ and (b) $RightBoundary(Hist(V))$.

Figure 4 shows the HSV histogram obtained from a long-shot frame in a soccer video. We see that the hue histogram is dominated by yellow-green bins, while histograms for saturation and value are fairly well spread out. With these observations in mind, we propose a ground color detection technique.

We assign N_H bins for the hue channel, N_S for the saturation channel, and N_V for the value channel. In this work, we set $N_H = 64$, $N_S = 64$, and $N_V = 256$; therefore, each histogram for the i -th frame is defined as

$$\begin{aligned} Hist(H^i)_k &: 0 \leq k < N_H, \\ Hist(S^i)_k &: 0 \leq k < N_S, \\ Hist(V^i)_k &: 0 \leq k < N_V. \end{aligned} \quad (1)$$

By utilizing the definitions in (1), some variables are defined as follows:

$$\begin{aligned} PeakIndex(Hist(V)) &= i, \\ &\text{where } Hist(V)_i \geq Hist(V)_p \\ &\text{for all } 0 \leq p < N_V, \\ LeftBoundary(Hist(V)) &= j, \\ &\text{where } Hist(V)_j = Peak(Hist(V)) / 20 \\ &\text{for } PeakIndex(Hist(V)) > j \geq 0, \\ RightBoundary(Hist(V)) &= k, \\ &\text{where } Hist(V)_k = Peak(Hist(V)) / 20 \\ &\text{for } PeakIndex(Hist(V)) \leq k < N_V, \text{ and} \\ LeftBoundary(Hist(S)) &= l, \\ &\text{where } Hist(S)_l = Peak(Hist(S)) / 20 \\ &\text{for } PeakIndex(Hist(S)) > l \geq 0. \end{aligned} \quad (2)$$

$Peak(Hist(V))$ denotes the peak of the value histogram and $PeakIndex(Hist(V))$ denotes the bin index of the $Peak(Hist(V))$. $LeftBoundary(Hist(V))$ is the index of the bin ranging from

$PeakIndex(Hist(V))$ to 0, the value of which corresponds to $Peak(Hist(V))/20$ in the value histogram. Finding $RightBoundary(Hist(V))$ is same as finding $LeftBoundary(Hist(V))$, except the search direction is from $PeakIndex(Hist(V))$ to N_V . $LeftBoundary(Hist(S))$ is obtained from the Saturation histogram with the same procedure as that used to obtain $LeftBoundary(Hist(V))$. Finding $LeftBoundary(Hist(V))$, and $RightBoundary(Hist(V))$ is carried out as shown in Fig. 5.

There is a relationship $g > r > b$ for the ground color, which we obtained by observing many soccer videos. We need to refine the conditions so as to minimize the false inclusion of a player's body or ground line into the ground detection results.

As given in [18], we can obtain the saturation S and value V from R , G , and B values as follows,

$$S = \frac{\text{Max}(R, G, B) - \text{Min}(R, G, B)}{\text{Max}(R, G, B)},$$

$$V = \text{Max}(R, G, B). \quad (3)$$

From (3) and the observed relationship, the refined conditions to decide if a pixel belongs to the ground are expressed as

$$Ground(x, y) = \begin{cases} 1 & \text{if } \begin{cases} g > 0.95 \cdot r \text{ and } r > 0.95 \cdot b \text{ and} \\ LeftBoundary(Hist(V)) - \theta_1 < g \text{ and} \\ g < RightBoundary(Hist(V)) + \theta_2 \text{ and} \\ \frac{g-b}{g} \cdot N_S > LeftBoundary(Hist(S)) - \theta_3 \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where r , g , and b denote the R , G , and B values at spatial location (x, y) , respectively, and the values are in the range [0-255]. Because there is a variety of soccer videos, some extreme cases also need to be handled for better performance. For instance, the video sequences may possess different lighting or ground conditions; therefore, the first condition in (4) is necessary to allow such cases. Regarding the value and saturation of each pixel, we observe that most of ground pixels satisfy conditions two to four. If a pixel's g value is between $LeftBoundary(Hist(V)) - \theta_1$ and $RightBoundary(Hist(V)) + \theta_2$ and $[(g-b)/g] \cdot N_S$ is greater than $LeftBoundary(Hist(S)) - \theta_3$, the pixel is assumed to be part of the ground. In this work, we set θ_1 , θ_2 , and θ_3 , to minimize false detection, to be 10, 5, and 8, respectively. This simple histogram-based classification is faster than the methods proposed in [7] and [19] and is more robust to the case where a part of the ground is shadowed, as shown in Fig. 6(c).

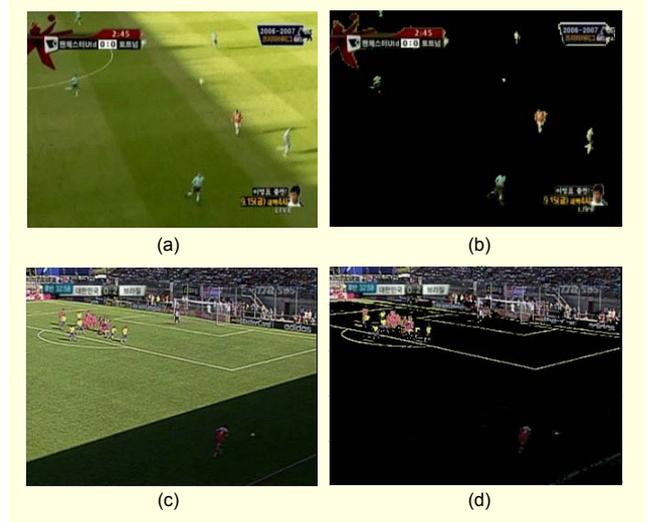


Fig. 6. Proposed detection scheme can even effectively handle shadowed ground areas: (a) and (c), ground with shadows; and (b) and (d), detected ground by the proposed scheme (regions in black).

2. Shot Classification

A. Constructing Ground Block Map

In the previous section, we developed a pixel classification scheme to identify the pixels in a video frame belonging to the ground region. For shot boundary detection and shot class decision, the whole frame is partitioned into 16×16 blocks to generate a ground block map (GBM), $GB(i, j)$, which is computed as follows. Define a set of pixels B_{ij} in the i -th block vertically and in the j -th block horizontally as in (5).

$$B_{ij} = \{(x, y) | i \times 16 \leq x < (i+1) \times 16, j \times 16 \leq y < (j+1) \times 16\} \quad (5)$$

Then, the GBM is computed as

$$GB(i, j) = \begin{cases} 1 & \text{if } \frac{\sum_{(x,y) \in B_{ij}} Ground(x, y)}{16 \times 16} \geq T_{Ground}, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $0 \leq i < frame.height/16$ and $0 \leq j < frame.width/16$. As shown in (6), a block is regarded as a ground block if the number of ground pixels with $Ground(x, y)=1$ is greater than or equal to T_{Ground} of the number of pixels in the block. In this paper, the threshold value T_{Ground} in (6) is set to 0.5 to determine the ground blocks.

B. Shot Boundary Detection

Shot boundary detection is a fundamental process in the video analysis. By providing a decision of shot classes at shot boundaries only, we can reduce the computation complexity

and the number of false decisions and, thus, improve the accuracy in shot class decision. Many approaches have been proposed to detect the shot boundaries [20]. We exploit the temporal block difference (TBD) between GBMs to make prompt and efficient shot boundary detection.

$$TBD_i = \sum_x \sum_y \{GB_{i-3}(x, y) \otimes GB_i(x, y)\}, \quad (7)$$

where \otimes denotes exclusive OR operation. The current frame is determined as a boundary between shots when both $TBD_{i-1} < \theta_{ShotChange}$ and $TBD_i \geq \theta_{ShotChange}$ are satisfied ($\theta_{ShotChange} = 30$ in this paper). For fast changing scenes, the condition $TBD_i \geq \theta_{ShotChange}$ is often satisfied; thus, the second condition $TBD_{i-1} < \theta_{ShotChange}$ is combined with first condition to reduce the number of false detections. Note that the current GBM is compared with that of the three previous frames in temporal order to deal with gradual changes due to fade in/out or other special visual effects.

C. Shot Class Decision

After the shot boundary is detected, the shot class decision has to be made. We use the GBM to classify four kinds of shots. First, we distinguish long-shot frames from others. Next, we classify non-long-shot frames into three categories: no-ground-shot, half-ground-shot, and full-ground-shot.

For long-shot detection, we filter the GBM to fill the holes inside the ground area as shown in Fig. 7. The detailed process is carried out as follows:

$$GB(i, j)=1 \text{ if } \begin{cases} [GB(i-1, j)=1 \text{ and} \\ \{GB(i+1, j)=1 \text{ or } GB(i+2, j)=1\}] \text{ or} \\ [\{GB(i-2, j)=1 \text{ or } GB(i-1, j)=1\} \\ \text{and } GB(i+1, j)=1] \text{ OR} \\ [GB(i, j-1)=1 \text{ and} \\ \{GB(i, j+1)=1 \text{ or } GB(i, j+2)=1\}] \text{ or} \\ [\{GB(i, j-2)=1 \text{ or } GB(i, j-1)=1\} \\ \text{and } GB(i, j+1)=1]. \end{cases} \quad (8)$$

This means that, if there are any one- or two-block gaps in either the horizontal or vertical direction, they are treated as ground blocks.

Once the holes are filled, the longest green segment for the i -th block column LGS_i is found as shown in Fig. 8. Shot classification is done by measuring the length of LGSs. As shown in Fig. 8, if there is at least one LGS smaller than the predefined value θ_L , then the frame is determined to be a non-long-shot frame. Otherwise, the frame is declared a long-shot frame.



Fig. 7. (a) Ground block map (GBM) before filling holes. The black blocks are non-ground blocks. (b) GBM with holes filled.



Fig. 8. LGS for each column. Values are shown in white on each column. (a) Long-shot and (b) non-long-shot.

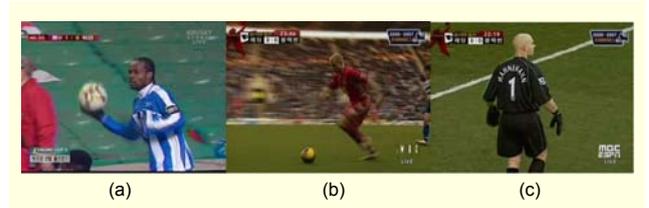


Fig. 9. Further classification of non-long shot frames: (a) no-ground-shot, (b) half-ground-shot, and (c) full-ground-shot.

$$Class(f_k) = \begin{cases} \text{Non-long-shot,} \\ \text{if } |LGS_i| < \theta_L \text{ for } 0 \leq i < BlocksInRow \\ \text{Long-shot, otherwise,} \end{cases} \quad (9)$$

where $\theta_L = BlocksInColumn/3$, and f_k denotes the current frame.

Unlike long-shot frames, which are full of ground regions, non-long-shot frames are further categorized for efficient processing. As shown in Fig. 9, non-long-shot frames can be grouped into three different categories: no-ground-shot, half-ground-shot, and full-ground-shot. In a no-ground-shot frame, no ground regions are observed. If the lower part of the frame is occupied by ground regions, it is categorized as a half-ground-shot frame. A frame is determined to be full-ground-shot if a close-up object is surrounded by ground regions.

By dividing non-long-shot frames into three types, we can develop a system that is more robust to false positives.

First, no-ground-shot frames are detected by counting the



Fig. 10. (a) Ground blocks in the lowest block row, and (b) upper half of a block map. Both are marked in gray.

number of ground blocks located in the lowest block row as shown in Fig. 10(a). If the number of ground blocks in a frame is less than a pre-determined threshold Th_1 and the number of ground blocks in the lowest block row is less than another threshold Th_2 , the frame is determined to be a no-ground-shot frame. Let N_W and N_H represent the number of blocks in the horizontal and vertical directions, respectively. In our system, Th_1 and Th_2 are set to $0.1 \cdot (N_W \times N_H)$ and $0.5 \cdot N_W$, respectively. Distinguishing a half-ground-shot frame from a full-ground-shot is conducted by counting the number of ground blocks located in the upper half of a frame (see gray part in Fig. 10(b)). If the number of ground blocks in the upper half is less than $N_W \cdot N_H \cdot 2/15$, the frame is determined to be a half-ground-shot frame.

Ground detection or computation of (4) in non-long-shot frames is performed using histogram information taken from the preceding long-shot frame, specifically, the first frame of the preceding long shot. This is because the amount of ground color may not be enough in the non-long-shot case.

3. Ground Color Customization

A. Ground Extraction in Pixel Accuracy

Our purpose is to customize ground color, thus accurate ground pixel detection is critical to achieving our goal. To have a good treatment of ground edges in ground pixel detection in long-shot frames, the GBM needs to be extended as follows. The extended GBM is defined as *EGB* and expressed as

$$EGB = \{(i, j) \mid (k, l) \in GB \text{ and } \|(i, j) - (k, l)\| \leq 1\}. \quad (10)$$

Figure 11(b) shows the original GBM, and Fig. 11(c) shows the extended GBM (*EGB*). We use a binary map, which is the inverse of the $Ground(x, y)$ expressed in (4), to detect the ground regions:

$$BinaryMap(x, y) = \begin{cases} 0, & \text{if } \begin{cases} Ground(x, y) = 1 \text{ and} \\ EGB\left(\text{round}\left(\frac{x}{16}\right), \text{round}\left(\frac{y}{16}\right)\right) = 1, \end{cases} \\ 1, & \text{otherwise.} \end{cases} \quad (11)$$

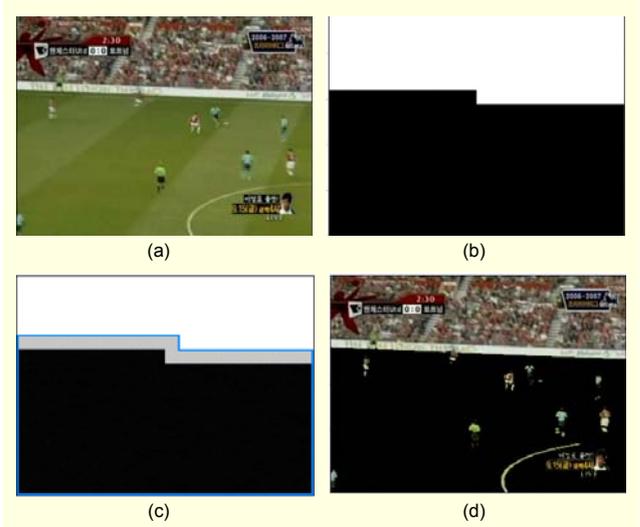


Fig. 11. (a) Long-shot image, (b) GB map, (c) EGB map (extended areas are denoted in gray), and (d) binary image with non-ground pixels in gray.

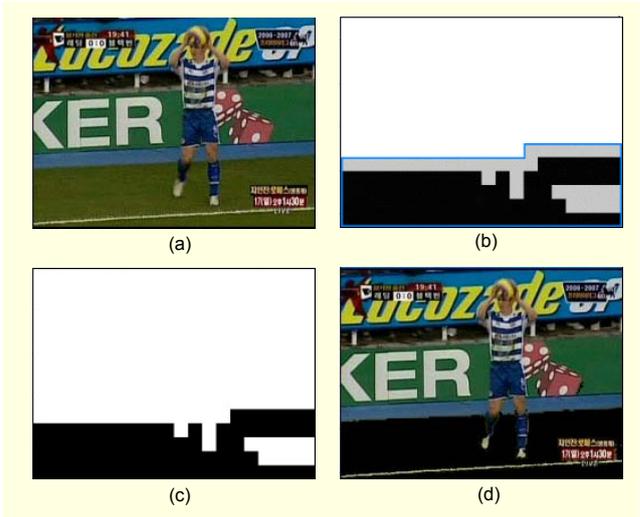


Fig. 12. (a) Original image, (b) original GBM, (c) EGB map (extended areas are denoted in gray), and (d) binary image with non-ground pixels in gray. Note that ground pixel detection is not conducted in white regions.

Figure 11(d) shows the binary map. Note that the non-ground blocks' neighboring ground blocks are not used in this process.

Once a frame is classified as a no-ground-shot frame, ground detection using (4) is not performed:

$$BinaryMap(x, y) = 0, \text{ if } Class(f_k) = \text{No-ground-shot}. \quad (12)$$

In half-ground-shot frames, ground detection or computation of (4) is conducted on extended ground blocks only. If there are scattered ground blocks in the white area, those are regarded as falsely detected ground blocks and are ignored. This prevents pixels in non-ground areas from being detected as ground pixels.

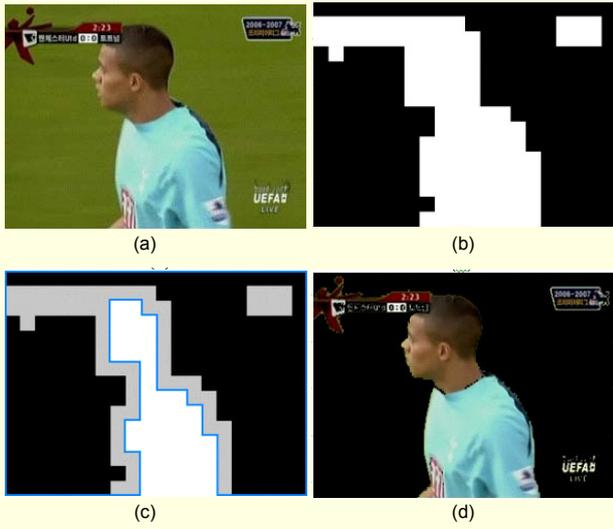


Fig. 13. (a) Original image, (b) original GBM, (c) EGB map (extended areas in gray), and (d) binary image with non-ground pixels in gray.

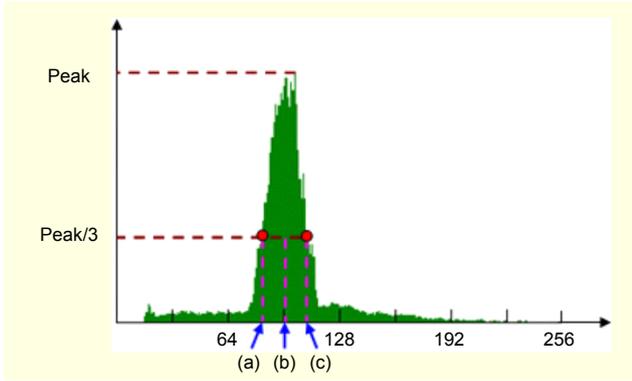


Fig. 14. Example of calculating the average value: (a) $PeakLeftIdx$, (b) $MeanIdx$, and (c) $PeakRightIdx$.

In full-ground-shot frames, ground pixels are similarly obtained from extended ground blocks only.

$$BinaryMap(x, y) = \begin{cases} 1, & \text{if } Ground(x, y) = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

B. Color Customization

Ground color customization is conducted in YUV space using $BinaryMap$ obtained in the previous sub-section. The objective of this sub-section is to adjust (Y, U, V) of pixels belonging to the ground in the $Binarymap$ using the average Y, U, V obtained from the target ground patch shown in Fig. 19. The detailed approach is as follows.

First, from the ground pixels in the image, the average $Y, U,$ and V are obtained using their histograms. In more detail, $Peak$

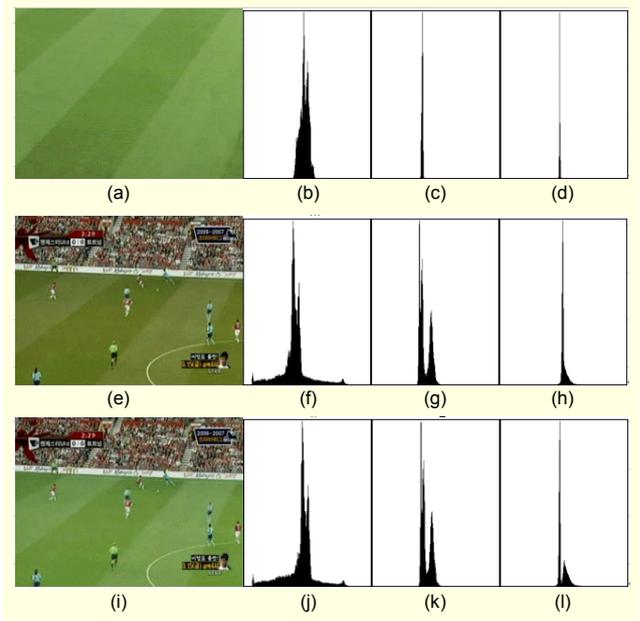


Fig. 15. Comparison of histograms before and after processing: (a), (e), (i) target, current, and customized image, respectively, where (b), (c), and (d) are $Y, U,$ and V histograms of (a); (f), (g), and (h) are $Y, U,$ and V histograms of (e); (j), (k), and (l) are $Y, U,$ and V histograms of (i).

and $PeakIndex$ are found by the procedure described in sub-section II.1. These values are used to find two indexes $PeakLeftIdx$ and $PeakRightIdx$ as expressed in (14) and (15) (respectively). From the values, $MeanIdx$ is calculated using (16). Indeed, this definition is applied to $Y, U,$ and V histograms of both the current frame and the target image patch, and they are named $MeanIdx(Hist(CurY)), MeanIdx(Hist(CurU)), MeanIdx(Hist(CurV))$ for the current frame, and $MeanIdx(Hist(TargetY)), MeanIdx(Hist(TargetU)),$ and $MeanIdx(Hist(TargetV))$ for the target image patch.

$$PeakLeftIdx = i, \quad (14)$$

where $Histo_i = Peak / 3$ for $PeakIndex \geq i \geq 0$

$$\quad (15)$$

$$MeanIdx = \left(\frac{PeakLeftIdx + PeakRightIdx}{2} \right) \quad (16)$$

Second, the differences between indexes are expressed in (17), and the ground pixel value change is conducted using (18). Each channel's difference $DiffY, DiffU,$ and $DiffV$ between the target patch and the current frame can be expressed as

$$\begin{aligned} DiffY &= MeanIdx(Hist(TargetY)) - MeanIdx(Hist(CurY)), \\ DiffU &= MeanIdx(Hist(TargetU)) - MeanIdx(Hist(CurU)), \\ DiffV &= MeanIdx(Hist(TargetV)) - MeanIdx(Hist(CurV)). \end{aligned} \quad (17)$$



Fig. 16. Screen shots of our demo system. Original image frame (left) and the customized frame (right). Users can select their favorite ground color from the patch list (the selected patch is indicated by a yellow border).

The Y , U , and V values of ground pixels in the current frame are added to by $DiffY$, $DiffU$, and $DiffV$, respectively, as follows:

$$\begin{aligned}
 & \text{If } BinaryMap(x, y) = 0 \\
 & \quad CurY(x, y) += DiffY, \\
 & \quad CurU(x, y) += DiffU, \\
 & \quad CurV(x, y) += DiffV.
 \end{aligned} \tag{18}$$

Images and histograms of the target patch, current frame, and processed image are shown in Fig. 15. These figures show the Y , U , and V values of the experimental image's ground pixels shifted towards the Y , U , and V average values of the target patch.

III. Experimental Results

The proposed system was implemented using Visual Studio 2003 (C++) under the Win32 environment and the FFmpeg library was utilized for decoding input videos. We used four soccer video sequences (Test 1: Reading FC vs. Blackburn FC, 2006-2007 England Premier League; Test 2: Manchester Utd. vs. Wigan Athletic FC, 2006 Carling Cup Final; Test 3: Korea vs. Uzbekistan, 2006 World Cup, Asian last preliminary match, A group, third match; Test 4: Korea vs. Uzbekistan, 2006 World Cup, Asian last preliminary match, A group, fourth match) which are encoded in MPEG-1 with an image size of 320×240 and a frame rate of 30 fps. The experiments were conducted on a low-end PC (Pentium 4, 3.0 GHz).

1. Performance Evaluation

Each video consists of 9,000 frames. The entire process of ground map construction, shot boundary detection, shot classification, and the ground color customization was conducted in real-time, resulting in an average processing speed of approximately 56.8 fps.

Table 1. Processing times with and without the ground color customization module.

	Frame rate without customization (fps)	Frame rate with customization (fps)
Test 1	68.3	55.9
Test 2	69.1	56.6
Test 3	70.4	57.4
Test 4	70.5	57.1
Average frame rate	69.6	56.8

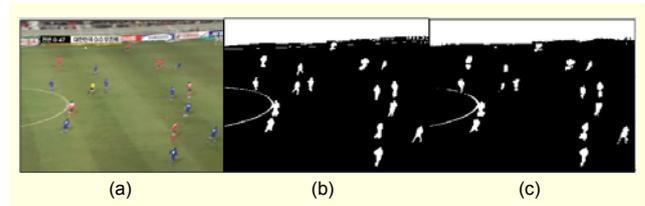


Fig. 17. (a) Original image, (b) ground detected manually, and (c) ground detected by the proposed algorithm.

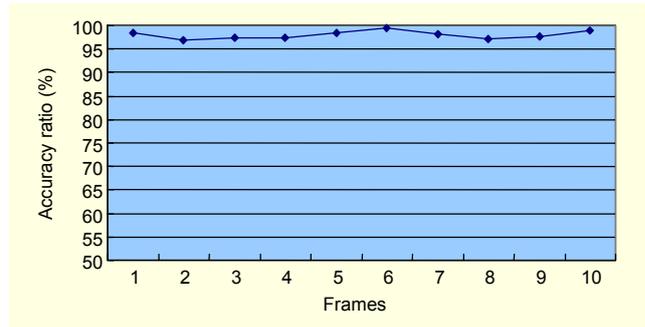


Fig. 18. Evaluation of ground detection.



Fig. 19. Target images used in the experiments.

Table 1 shows the processing speed with and without the ground color customization process.

We conducted the following procedure to evaluate the proposed ground detection algorithm. First, we extract 10 frames including long-shot, middle-shot, and close-up-shot frames from the four test videos. Figure 17(a) shows one of extracted frames. Then, the detection error was computed by comparing the manually generated ground truth map and the result obtained from the proposed algorithm (see Fig. 17).

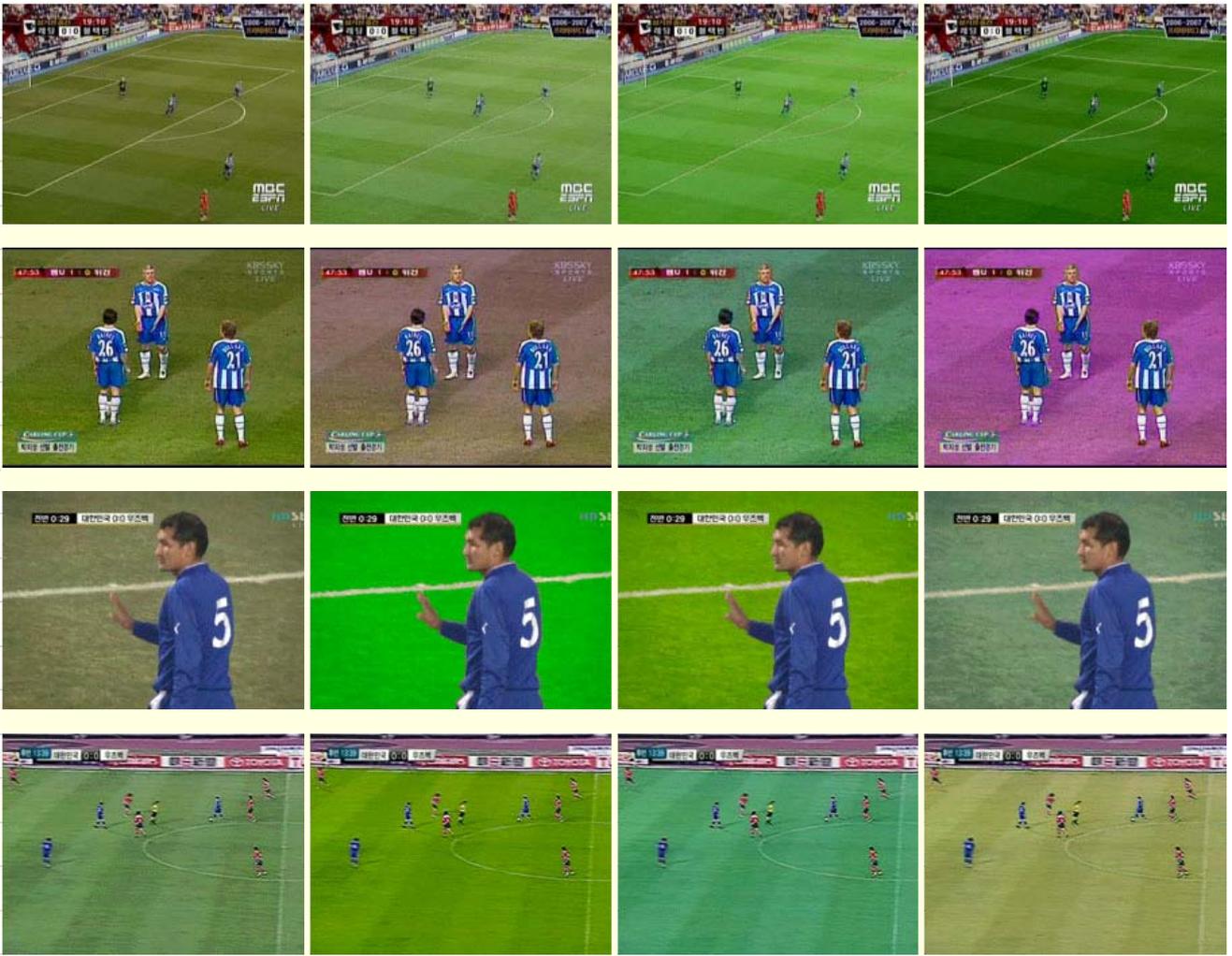


Fig. 20. Test results from the test videos. First column: original images from four videos. Second column: the result images using 1st, 4th, 7th, and 10th target images. Third column: the result images using 2nd, 5th, 8th, and 11th target images. Fourth column: the result images using 3rd, 6th, 9th, and 12th target images.

$$err(x, y) = \begin{cases} 1, & \text{if } RES(x, y) \neq TR(x, y), \\ 0, & \text{otherwise,} \end{cases} \quad (19)$$

where $RES(x, y)$ denotes the result obtained by the proposed algorithm and $TR(x, y)$ is the ground truth map. We finally calculated the accuracy as

$$Accuracy (\%) = \left\{ 1 - \frac{\sum_x \sum_y err(x, y)}{frame.height \times frame.width} \right\} \times 100. \quad (20)$$

Figure 18 shows the accuracy of the ground detection for 10 sample frames. The proposed algorithm has an average accuracy of 97.9%.

Figure 19 shows the twelve target images used in our system. The $MeanIdx(Hist(TargetY))$, $MeanIdx(Hist(TargetU))$, and

$MeanIdx(Hist(TargetV))$ of the first three patches from the top-left to top-right are $\{122, 100, 118\}$, $\{122, 92, 108\}$, and $\{84, 92, 108\}$, respectively. Figure 20 shows the test results obtained from the four test videos. The ground color is customized to that of the target patch, providing various viewing experiences. Note that users can choose any target colors whenever they want.

The experimental results for the frames with ground shadow are shown in Fig. 21. Note that the customization is affected by the amount of shadow cast on the ground. Shadow regions are customized to the selected ground color if the ground is mostly occupied by shadow. Thus the customized ground becomes a little brighter outside the shadow (see Figs. 21(a) and (b)). On the contrary, the shadow region becomes a little darker if the shadow takes up small regions because the normal ground regions are customized to the selected ground color in this case (see Figs. 21(c) and (d)).

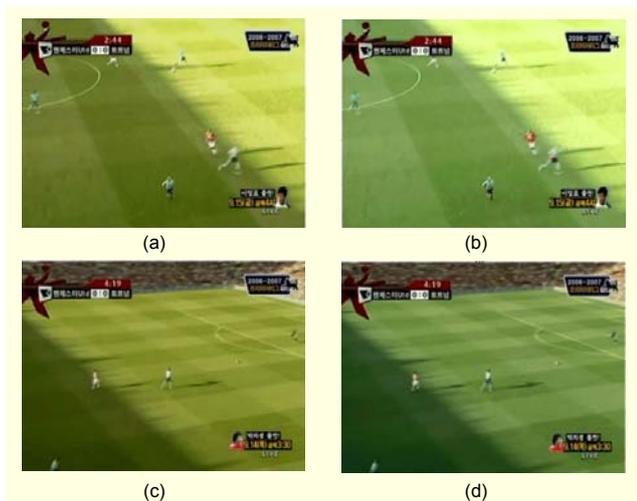


Fig. 21. Customization of shots that have shadow regions: (a) and (c) original images, (b) and (d) customized image of (a) and (c), respectively. The top-left target patch in Fig. 19 is used for customization.

Table 2. Test environment for PDA porting (hardware).

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

2. Subjective Evaluation

According to [1], subjective satisfaction can be evaluated by subjective or objective measures. Objective measures can be based on observation of the behavior or physiological responses of the user. Subjective measures of satisfaction quantifying the strength of a user's subjectively expressed reactions, attitudes, or opinions. These can be quantified in a number of ways, including the use of surveys and questionnaires.

We conducted subjective evaluation to see if the proposed system could improve the viewing experience of viewers. The algorithm was implemented on PDA. Its specifications are summarized in Table 2. Then, we conducted a survey, where test 3 and 4 video clips were used. The video clips had a resolution of 320×240 and were encoded in H.264/AVC format. The videos were displayed over the whole display panel (640×480). Each clip was 900 frames long and composed of long-shot, middle-shot, and close-up-shot frames.

There were 74 participants, and most of them were college or high school students who were familiar with multimedia mobile devices. Their ages ranged from 15 to 49, with an



Fig. 22. (a) PDA captured image with ground patches. When the LCD panel is touched, the patches appear. (b) Customized image when lower-left patch is selected. The original video is displayed when the user touches anywhere outside the patches.

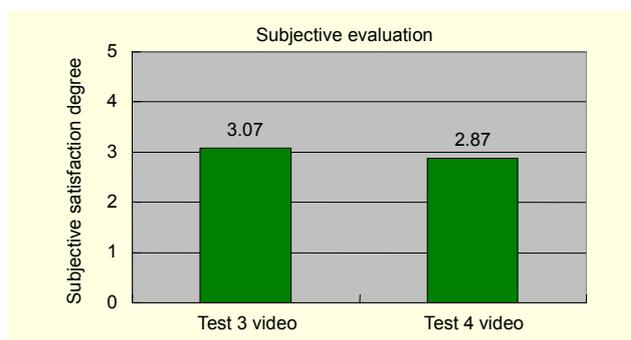


Fig. 23. Effect of ground customization on satisfaction.

average of 25.6 years. The average playback frame-rate, including decoding and displaying time was 10.16 fps.

The participants were told the purpose of the study before the experiments were conducted. They were provided with PDAs and were asked to view video clips at a comfortable distance. When the participants were comfortable with the procedure, the experiments began and the participants watched original video clips and customized video clips successively. This evaluation method is very similar to the DSIS [21] which is a popular method to measure video quality. While the participants watched the video clips, they were allowed to select their favorite target patch as shown in Fig. 22. Of course, the playback frame rates of the original clips and the customized clips were the same.

The participants were asked to evaluate their degree of satisfaction when viewing the customized scene. They were asked to evaluate how much their visual experience or satisfaction was raised, compared to that of viewing the original videos. The assessment was done over a six-step scale ranging from 0 (no difference) to 5 (much better) for each clip.

Figure 23 shows the satisfaction improvement associated with ground customization. The score obtained from the test 3 video is a little higher than that from test 4 video, and both are around three. We believe that users will be able to enjoy this customization scheme with great satisfaction.

IV. Conclusion

We propose a ground color customization algorithm to deliver a more pleasant viewing experience to the viewers of outdoor sports video, such as soccer video. The ground color customization is conducted on the ground pixels. The proposed ground color customization scheme, including shot classification, can be performed in real-time using software with frame rates around 56.8 fps on a Pentium-IV 3.0 GHz PC for QVGA video. We believe that this ground color customization algorithm can provide better viewing environments for watching soccer video. It can particularly improve viewing satisfaction for the viewers of mobile devices with small screens.

The goal of our on-going work is to construct a framework to deliver viewers a better viewing experience with robust and efficient algorithms. The framework is not necessarily limited to soccer videos and can be applied to other types of field sports videos.

References

- [1] ISO 9241-11, Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) Part 11: Guidance on Usability, 1998.
- [2] K. Seo, J. Ko, I. Ahn, and C. Kim, "An Intelligent Display Scheme of Soccer Video for Mobile Devices," *IEEE Trans. Circuits and Systems for Video Technology(CSVT)*, vol. 17, no. 10, Oct. 2007, pp. 1395-1401.
- [3] W. Kim and C. Kim, "Scoreboard Extraction from Soccer Videos for Multimedia Mobile Users," *Conf. the Korean Society of Broadcast Engineers*, Nov. 2006.
- [4] J. Ko, J. Lee, C. Kim, and V. Bhaskaran, "The Intensity Reduction of Ground Shadow to Deliver Better Viewing Experiences of Soccer Videos," *Proc. SPIE Applications of Digital Image Processing XXIX*, San Diego, Aug. 2007.
- [5] P. Valdez and A. Mehrabian, "Effects of Color on Emotion," *Journal of Environmental Psychology*, vol. 123, no. 4, 1994, pp. 394-409.
- [6] E. Grandjean, *Ergonomics of the Home*, New York: Wiley, 1973.
- [7] A. Ekin, A.M. Tekalp, and R. Mehrotra, "Automatic Soccer Video Analysis and Summarization," *IEEE Trans. on Image Processing*, vol. 12, 2003, pp. 796-807.
- [8] J. Assfalg, M. Bertini, C. Colombo, A.D. Bimbo, and W. Nunziati, "Automatic Extraction and Annotation of Soccer Video Highlights," *Int'l Conference on Image Processing*, vol. 2, 2003, pp. 14-17.
- [9] A. Ekin and A. Murat Tekalp, "Robust Dominant Color Region Detection and Color-Based Applications for Sports Video," *Int'l Conf. Image Processing*, vol. 1, 2003, pp. 21-24.
- [10] X.F. Tong, H.Q. Lu, and Q.S. Liu, "An Effective and Fast Soccer Ball Detection and Tracking Method," *Int'l Conference on Pattern Recognition*, vol. 4, 2004, pp. 795-798.
- [11] 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," *MULTIMEDIA 3: Proc. the Eleventh ACM Int'l Conference on Multimedia*, ACM Press, 2003, pp. 11-20.
- [12] C.L. Huang, H.C. Shih, and C.Y. Chao, "Semantic Analysis of Soccer Video Using Dynamic Bayesian Network," *IEEE Trans. on Multimedia*, vol. 8, 2006, pp. 749-760.
- [13] H.S. Yoon, Y. J. Bae, and Y.K. Yang, "A Soccer Image Sequence Mosaicking and Analysis Method Using Line and Advertisement Board Detection," *ETRI Journal*, vol. 24, no. 6, Dec. 2002, pp. 443-454.
- [14] S.C. Pei, Y.C. Zeng, and J.J. Ding, "Color Images Enhancement Using Weighted Histogram Separation," *IEEE Int'l Conference on Image Processing*, 2006, pp. 2889-2892.
- [15] J. Zhao and S. Lei, "Automatic Digital Image Enhancement for Dark Pictures," *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 2, 2006, pp. II.953-II.956.
- [16] Y. Chang, S. Saito, and M. Nakajima, "Example-Based Color Transformation for Image and Video," *Proc. the 3rd Int'l Conf. Computer Graphics and Interactive Techniques in Australasia and South East Asia*, 2005, pp. 347-353.
- [17] X. Sun, J. Foote, D. Kimber, and B.S. Manjunath, "Region of Interest Extraction and Virtual Camera Control Based on Panoramic Video Capturing," *IEEE Trans. Multimedia*, vol. 7, no. 5, Oct. 2005, pp. 981-990.
- [18] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice*, Addison-Wesley, 1996, pp. 592-593.
- [19] K. Wan, X. Yan, X. Yu, and C. Xu, "Real-Time Goal-Mouth Detection in mpeg Soccer Video," *MULTIMEDIA: Proc. the Eleventh ACM Int'l Conf. Multimedia*, ACM Press, 2003, pp. 311-314.
- [20] R.W. Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms," *Storage and Retrieval for Image and Video Databases VII*, vol. 3656, no. 1, SPIE, 1998, pp. 290-301.
- [21] ITU-R BT.500-11, "Recommendation: Methodology for the Subjective Assessment of the Quality of Television Pictures," 2002.



Ilkoo Ahn received the BS degree in electronics and information engineering from Chonbuk National University, Jeonju, Rep. of Korea, in 2005. He is currently pursuing an MS degree with the School of Engineering, Information and Communications University (ICU), Daejeon, Rep. of Korea. His research interests include intelligent display, 3D video processing, and machine learning.



Youngwoo Kim received the BS degree in computer engineering from Kwangwoon University, Seoul, Rep. of Korea, in 2007. He is an MS candidate with the School of Engineering, Information and Communications University (ICU), Daejeon, Korea. His research interests include intelligent display, 2D-to-3D conversion, and machine learning.



Changick Kim received the BS degree in electrical engineering from Yonsei University, Seoul; the MS degree in electronics and electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, Rep. of Korea; and the PhD degree in electrical engineering from the University of Washington, Seattle; in 1989, 1991, and 2000, respectively. From 2000 to 2005, he was a senior member of technical staff at Epson Research and Development, Inc., Palo Alto, CA. Since February 2005, he has been with the School of Engineering, Information and Communications University (ICU), Daejeon, Korea, where he is currently an assistant professor. His research interests include multimedia communication, 3-D video processing, image/video understanding, intelligent media processing, and video coding for IPTV.