Ground Color Customization of Soccer Videos

Ilkoo Ahn and Changick Kim
School of Engineering, Information and Communications University
119 Munji street, Yuseong-Gu, Daejeon, 305-714, South Korea
Email : {ahn19, ckim }@icu.ac.kr

In this paper, we present a method for customizing the ground color in outdoor sports video to provide TV viewers with a better viewing experience. This relates to content personalization and the issue is becoming critical with the advent of mobile TV and interactive TV such as IPTV. In the case of watching outdoor sports video such as soccer video, it is sometimes observed that the ground color is not satisfied by viewers. In this work, the proposed algorithm is focused on customizing the ground color to deliver viewers better viewing experiences. The algorithm comprises three modules; ground detection, long-shot detection, and ground color customization. We customize the ground color by considering the difference between ground colors from both input video and target video.

Experimental results show that the proposed scheme offers useful tools to provide a more comfortable viewing environment and is amenable to realtime performance even in a software based implementation.

Key words: Mobile devices, Soccer video analysis, ground color, content personalization, and content customization.

1. INTRODUCTION

Recently, the rapid development of multimedia communication technologies and computing power in user terminals make it possible to provide TV viewers with customized multimedia contents. In particular, the customization of mobile content is driven by several limited resources such as modest processing power, limited battery power, small memory capacity and smaller-sized LCD display. Unlike other limited resources, which are getting better solutions, the small sized display is the peculiar attribute of mobile devices. Therefore it is required that novel customization techniques for better viewing experiences need to be equipped for the sake of mobile users.

There have been several schemes for such customization: extracting and magnifying region of interest (ROI) on small LCD panel for mobile device user [1], magnifying scoreboard for mobile device user [2], and reducing the intensity of shadows cast on the ground in outdoor sports videos [3].

There has been a number of research activities focused on enhancing ground color. However, the most of them are mainly focused on event detection or summarization [4][5], or detecting ground to identify players and ball [6]-[8] and so on [9].

Figure 1 shows various ground colors in soccer videos. In general, it is obvious viewers would prefer fair ground color. 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' satisfactions.

In soccer videos, three types of shots are edited and transmitted as shown in Fig. 2. The long-shot frames are images captured in a long distance, thus most parts of the shot tend to be occupied by ground (see Fig. 2-(a)). In this case, the viewer's viewing experience can be influenced by the ground color. For the other type of frames, which may be either mid-shot frame or close-up frame, the ground color would have little impact on the viewing experience since the viewers' eye gazes are usually drawn by interesting objects (see Fig. 2-(b) and (c)). Thus, for comfortable and efficient display of soccer video, the image frames need to be classified into two categories, long-shot frames and non-long-shot frames. Then the ground color of the long-shot frames are customized if a viewer raise a request.

(a) (b) (c)

Fig. 1. Examples of desolate ground. These make viewers feel uncomfortable.

(a) (b) (c)

Fig. 2. Three kinds of frame in soccer video. (a) Long-shot frame, (b) Mid-shot frame, (c) Close-up shot frame.

In Section 2, we introduce a ground extraction algorithm and shot classification algorithm based on our previous work [1] and then the ground color customization technique is applied to the

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

extracted ground region. Experimental results and conclusion follow in Section 3 and 4 respectively.

2. Proposed Algorithm

2.1. Ground Detection

Since the ground color may vary over different soccer videos, and may vary over shots from different camera angle even in a same soccer video, detecting the ground needs to be performed every frame.

It has been observed that the HSV space is better suited for computing the color changes and separating hue from saturation and brightness adds robustness under most lighting variations [10].

We assign N_H bins for Hue channel, N_S for Saturation channel, and N_V for 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 in (1).

$$Hist(H^i)_k : 0 \le k < N_H,\ Hist(S^i)_k : 0 \le k < N_S,\ Hist(V^i)_k : 0 \le k < N_V \quad (1)$$

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

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

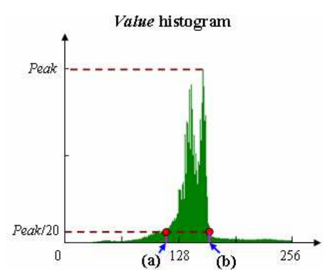


Fig. 3. Finding of (a) *LeftBoundary(Hist(V))* (b) *RightBoundary(Hist(V))*.

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 *NV*. *LeftBoundary(Hist(S))* is obtained from the *Saturation* histogram with the same procedure addressed for *LeftBoundary(Hist(S))*. Finding *LeftBoundary(Hist(V))*, and *RightBoundary(Hist(V))* is illustrated in Fig. 3.

Basically, it is noted that there is a relationship $g > r > b$ on the ground color, which is 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.

The equations to compute Saturation S and Value V in HSV color space from R,G,B values in RGB space is as follows [11].

$$\begin{aligned} S &= \frac{Max(R,G,B) - Min(R,G,B)}{Max(R,G,B)} \quad (0.0 \le S \le 1.0) \\ V &= Max(R,G,B) \qquad (0.0 \le V \le 1.0) \end{aligned} \quad (3)$$

where R, G, B are all normalized values from 0 to 1. By using (3) and the observed relationship, $g > r > b$ for the ground area, we can denote V as $g{=}G \cdot N_V$ since $G{\approx}Max(R,G,B)$ and $N_V{=}256$, whereas $S{=}[(g{-}b)/g] \cdot N_S{=}[(G{-}B)/G] \cdot N_S$ since $G{\approx}Max(R,G,B)$ and $B{\approx}Min(R,G,B)$. Finally, the refined conditions, described in RGB space, to distinguish each pixel whether or not it belongs to the ground are expressed in (4).

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

where r,g,b denote the R, G, B values at spatial location (x, y) respectively, and the values are in the range [0-255]. The video sequences may possess different lighting or ground conditions, thus the first condition in (4) is necessary. If a pixel's g value is between *LeftBoundary(Hist(V))*- θ_1 and *RightBoundary(Hist(V))*+ θ_2 and $[(g-b)/g] \cdot N_S$ is greater than *LeftBoundary(Hist(S))* - θ_3, the pixel is assumed to be in the ground. In this work, we set θ_1, θ_2, and θ_3, to minimize false detection, to be ten, five, and eight respectively. This simple histogram based classification is faster than the methods proposed in [4] and [12].

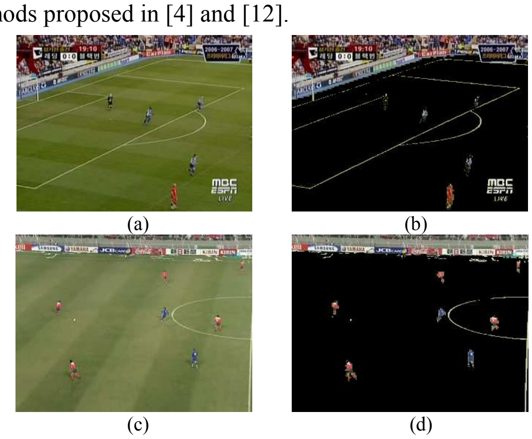


Fig. 4. The proposed detection scheme. (a), (c) Original image (b), (d) Detected ground by using the proposed scheme (regions in black).

2.2. Shot Classification

2.2.1 Constructing Ground Block Map

For shot boundary detection and shot class decision, the whole frame is partitioned into 16×16 blocks to generate a ground block map *GB(i,j)*. Define a set of pixels B_{ij} in the *i-th* and *j-th* block in vertical and horizontal direction, respectively as in (5).

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

Then, the ground block map *GB(i,j)* is computed as

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

where $0 \le i < frame.height/16$ and $0 \le j < frame.width/16$. 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.

2.2.2 Shot Boundary Detection

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 rate in shot class decision. We exploit the temporal block difference (TBD) between ground block maps to make prompt and efficient shot boundary detection.

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

where $\otimes$ denotes Exclusive OR operation. The current frame is determined as a boundary between shots when both $TBD_{i\text{-}1} < \theta_{ShotChange}$ and $TBD_i \ge \theta_{ShotChange}$ are satisfied ($\theta_{ShotChange} = 30$ in this paper). For the fast changing scene, the condition $TBD_i \ge \theta_{ShotChange}$ tends to be often satisfied, thus the second condition $TBD_{i\text{-}1} < \theta_{ShotChange}$ is combined to reduce the false detections. Note that the current ground block map is compared with that of three frames before in temporal order so as to deal with gradual changes due to fade in/out or other special visual effects.

2.2.3 Shot Class Decision

In this work, we use the ground block map to distinguish between long-shot frame and other frame types. Firstly we filter the ground block map to fill the holes inside the ground area as shown in Fig. 5. The detailed process is as follows.

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

This means if there exist one- or two-block gaps in either horizontal or vertical direction, they are treated as a ground block.

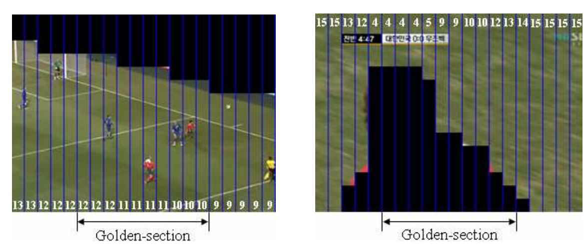
(a) (b)

Fig. 5. (a) Ground block map (GBM) before filling holes. (b) GBM with holes filled.

Once the holes are filled, the longest green segment for the *i-th* block column, LGS_i is searched. LGS_i is shown in Fig. 6. Shot classification is done by measuring the length of LGSs inside the golden section defined in [4]. As shown in Fig. 6, if there is at least one LGS smaller than predefined value, θ_L, then the frame is determined as a non-long-shot frame. Otherwise, the frame is declared as a long-shot frame.

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

where $\theta_L = BlocksInColumn/3$, and $GSLeft = BlocksInRow\times 3/11$, $GSRight = BlocksInRow\times 8/11$, and f_k denotes the current frame.



(a) (b)

Fig. 6. LGS for each column. The values are shown in white on each column. (a) Long-shot. (b) Non-long-shot.

2.3. Ground Color Customization

2.3.1 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 the pixel belonging to ground in the *Binarymap* by average (Y,U,V) obtained from the target ground patch shown in Fig. 9.

Firstly, from the ground pixels in the image, the average *Y*, *U*, and *V* are obtained using their histograms. In more detail, *Peak* and *PeakIndex* are found by the procedure addressed in sub-section 2.1. These values are used to find two indexes *PeakLeftIdx* and *PeakRightIdx* as expressed in (10) and (11), respectively. From the values, *MeanIdx* is calculated using (12). Indeed, this definition is applied to *Y*, *U*, *V* histograms of both current frame and 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))*, *MeanIdx(Hist(TargetV))* for the target image patch.

$$PeakLeftIdx = i, \quad \text{where } Histo_i = Peak/3 \text{ for } PeakIndex \geq i \geq 0 \tag{10}$$

$$PeakRightIdx = i, \quad \text{where } Histo_i = Peak/3 \text{ for from } PeakIndex \leq i < 256 \tag{11}$$

$$MeanIdx = \left(\frac{PeakLeftIdx + PeakRightIdx}{2}\right) \tag{12}$$

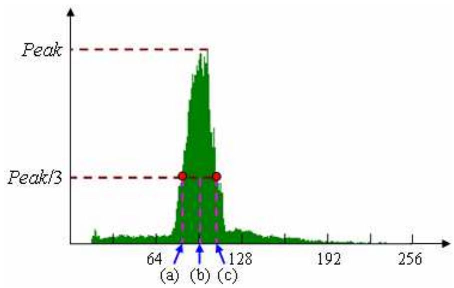


Fig. 7. Example of calculating the average value. (a) *PeakLeftIdx* (b) *MeanIdx* (c) *PeakRightIdx*.

Secondly, the differences of indexes are expressed in (13), respectively and the ground pixel value change is conducted by using (14). Each channel's difference $DiffY$, $DiffU$, and $DiffV$ between target patch and current frame can be expressed as (13).

$$\begin{aligned} DiffY &= MeanIdx(Hist(TargetY)) - MeanIdx(Hist(CurY)) \\ DiffU &= MeanIdx(Hist(TargetU)) - MeanIdx(Hist(CurU)) \\ DiffV &= MeanIdx(Hist(TargetV)) - MeanIdx(Hist(CurV)) \end{aligned} \tag{13}$$

The *Y*, *U*, and *V* values of ground pixel of current frame are added by $DiffY$, $DiffU$, and $DiffV$ respectively as you can see in (14).

$$\begin{aligned} \text{If } BinaryMap(x,y) = 0,\ & CurY(x,y) \mathrel{+}= DiffY \\ & CurU(x,y) \mathrel{+}= DiffU \\ & CurV(x,y) \mathrel{+}= DiffV \end{aligned} \tag{14}$$

Images and histograms of target patch, current frame, and processed image are depicted in Fig. 8. These figures show the *Y*, *U*, and *V* values of experimental image's ground pixels shifted towards the *Y*, *U*, and *V* average values of the target patch.

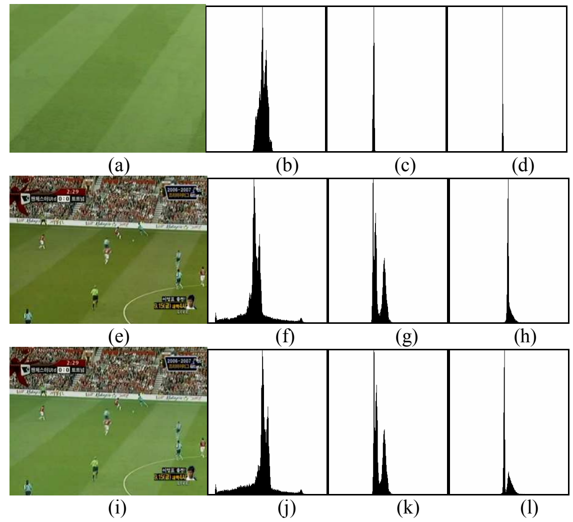


Fig. 8. Comparison of histograms before and after processing. (a), (e), (i) target, current, and customized image respectively. (b), (c), (d) Y, U, V histograms of (a), (f), (g), (h) Y, U, V histograms of (e), (j), (k), (l) Y, U, V histograms of (a).

3. Experimental Results

The proposed system has been implemented by using Visual Studio 2003 (C++) under Win32 environment and FFMpeg library has been utilized for decoding input videos. We used three soccer video sequences which are encoded in MPEG-1 with the image size of 320x240 and frame rate 30fps. The experiments were conducted on a low-end PC (Pentium4 3.0GHz). Each video consists of 9000 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 56.6fps.

Table 1. The processing time with and without the ground color customization module.

	Frame rate without customization (fps)	*Frame rate with customization (fps)*
Test1	68.3	55.9
Test2	69.1	56.6
Test3	70.4	57.4
Average frame rate	69.3	56.6

Figure 9 shows the three target images in our experiments. The *MeanIdx(Hist(TargetY))*, *MeanIdx(Hist(TargetU))*, and *MeanIdx(Hist(TargetV))* of Fig. 9-(a), (b), and (c) are {122,100,118},{122,92,108}, and {84,92,108} respectively. Note that users can choose any target colors whenever they want.

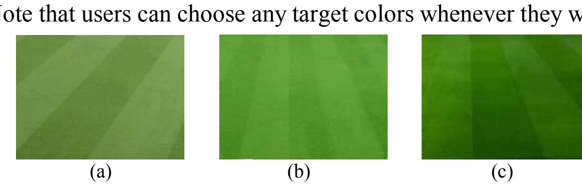


Fig. 9. Three target images used in the experiments. (a) Target image1. (b) Target image2. (c) Target image3.

Fig. 10. Test results from test videos. Leftmost column: original images from four videos, Second column: the result images using target image1 (Fig.9-(a)), Third column: the result images using target image2 (Fig.9-(b)). Rightmost column: the result images using target image3 (Fig.9-(c)).

4. Conclusion

We propose a ground color customization algorithm to deliver a more pleasant viewing environment to the viewers watching outdoor sports video, such as soccer video. The ground color customization is conducted on the ground pixels belonging to long-shot frame. The proposed ground color customization scheme including shot classification can be performed in real-time in software with frame rates around 56.6fps on Pentium-IV 3.0 GHz PC for QVGA video. We believe that this ground color customization algorithm can provide more appropriate viewing environments for watching the soccer video and, especially, can provide better viewing satisfaction to the viewers of mobile devices with small screen. Our on-going work is to construct a framework to deliver viewers better viewing experience with robust and efficient algorithms, and the proposed ground color customization is a part of it.

Reference

[1] Keewon Seo, Jaeseung Ko, Ilkoo Ahn and Changick Kim, "An Intelligent Display Scheme of Soccer Video for Mobile Devices," in *IEEE Transactions on Circuits and Systems for Video Technology(CSVT)*, vol. 17, no. 10, Oct 2007, pp.1395-1401.

[2] Wonjun Kim and Changick Kim, "Scoreboard Extraction from Soccer Videos for Multimedia Mobile Users," in *Conference of the Korean society of broadcast engineers*, Nov. 2006.

[3] Jaeseung Ko, Jaeho Lee, and Changick Kim, "The Intensity Reduction of Ground Shadow to Deliver Better Viewing Experiences of Soccer Videos," in *Proc. SPIE Applications of Digital Image Processing XXIX*, 6696-29, San Diego, Aug. 2007.

[4] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," in *IEEE Tr. on Image Processing*, vol. 12, 2003, pp. 796–807.

[5] Assfalg, J.; Bertini, M.; Colombo, C.; Bimbo, A.D.; Nunziati, W.,"Automatic extraction and annotation of soccer video highlights,"in *International Conference on Image Processing*, Volume 2. 2003, pp14-17.

[6] A. Ekin, and A. Murat Tekalp, "Robust dominant color region detection and color-based applications for sports video", in *International Conference on Image Processing*, Volume 1, 2003, Page(s):I - 21-4 vol.1.

[7] X.F. Tong, H.Q. Lu, Q.S. Liu, "An effective and fast soccer ball detection and tracking method," in *International Conference on Pattern Recognition*, Vol.4, 2004, pp.795 - 798.

[8] 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*, ACM Press, (New York, NY, USA), 2003, pp. 11-20.

[9] C.L Huang, H.C Shih, C.Y Chao, "Semantic analysis of soccer video using dynamic Bayesian network," in *IEEE Tr. on Multimedia*, Volume 8, 2006, pp.749-760.

[10] X.Sun, J.Foote, D.Kimber, and B.S.Manjunath, "Region of Interest Extraction and Virtual Camera Control Based on Panoramic Video Capturing," in *IEEE Tr. On Multimedia*, vol. 7, no. 5, October 2005, pp.981-990.

[11] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, "*Computer Graphics - Principles and Practice"*, Addison-Wesley, 1996, pp. 592-593.

[12] 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*, ACM Press, (New York, NY, USA), 2003, pp. 311–314.