

A Soccer Image Sequence Mosaicking and Analysis Method Using Line and Advertisement Board Detection

Ho-Sub Yoon, Young-lae J. Bae, and Young-kyu Yang

This paper introduces a system for mosaicking sequences of soccer images in a panoramic view for soccer game analysis. The continuous mosaic images of the soccer ground field allow the user to view a wide picture of the players' actions. The initial component of our algorithm automatically detects and traces the players and some lines. The next component of our algorithm finds the parameters of the captured image coordinates and transforms them into ground model coordinates for automatic soccer game analysis. The results of our experimentations indicate that the proposed system offers a promising method for segmenting, mosaicking, and analyzing soccer image sequences.

I. INTRODUCTION

The recent advances in multimedia technology have accelerated the technical development of image processing. Image processing techniques have been expanding in the field of sports, as well as in applications for facial and gesture recognition in the human body. Studies on the analysis of soccer games received an increased impetus from the 2002 World Cup games. However, the extraction and tracking of moving players from a video image presents complex challenges that have yet to be solved. A panoramic scene presents a larger view of the soccer field and allows the user to view a wider picture of the action. We developed and introduce in this paper an algorithm for mosaicking and generating panoramic view images from sequences of soccer game images.

Gong [1] proposed a method of parsing soccer games to determine the position of the field, and Intille [2] worked out a technique for visual tracking of American football under the “closed world” assumption. Gong’s system classified a sequence of soccer frames into various play categories—such as shot at the left goalpost, top left corner kick, play in right penalty area, and play in mid field—based on an *a priori* model comprising four major components: a soccer field, a ball, the players, and the motion vectors. Intille’s [2] approach was a focused tracking method in the American football domain, but the same techniques, such as closed-world analysis and context-specific tracking, can be applied to the soccer domain. Methods for generating a panoramic view have been proposed by Teodosio and Bender [3]; they spatially and temporally superimposed individual frames on top of each other, taking into account the global camera actions of panning and zooming.

Manuscript received Feb.5, 2002; revised Aug. 8, 2002.

Ho-Sub Yoon (phone: +82 42 860 5233, e-mail: yoonhs@etri.re.kr), and Young-kyu Yang (e-mail: ykyang@etri.re.kr) are with Computer & Software Research Laboratory, ETRI, Daejeon, Korea.

Young-lae J. Bae (e-mail: yjlb1@bclinc.com) is a professor at Chungbuk Provincial University of Science & Technology.

Some similar previous studies have been reported by Yow et al. [4], Irani et al. [5] and Seo et al. [6] in the field of soccer game analysis. Yow's system detected and extracted soccer highlights by analyzing the image contents and presented the shots of these actions by the panoramic reconstruction of the selected events. Next, Irani et al. determined an image-to-model transform and computed the trajectory of each player. These trajectories showed the moving pattern of a player or group of players and could also be used in a game analysis after the game or put into a video annotation. Seo et al. [6], who have continued to work on this project since 1998, used color-based tracking and image mosaicking for soccer game analysis. Taki et al. [7] analyzed a soccer game to quantitatively evaluate teamwork using captured images by cameras at strategic positions. Matsui et al. [8] and Bebie et al. [9] developed image synthesis systems that calculated the player's position from the soccer image and synthesized the scene from any viewpoint using both Computer Vision and Computer Graphics techniques. The paper by Nicolas [10] proposed a new method for creating dynamic mosaic representations of scenes and using the object-based manipulations of the original video data including tennis players. Sun and Tekulp [11] introduced a trifocal transfer, which is an image-based scene representation, as a motion compensation method that uses three frames at a time. They proposed approximating the dense correspondence between two of three views by a parametric model for background mosaic generation. Xu et al. [12] classified a sample soccer frame into 3 kinds of view using a unique domain-specific feature, the grass-area ratio. They developed effective rules for segmenting plays or breaks from a label sequence. Ekin and Takalp [13] proposed shot boundary detection using the ratio of grass-colored pixels in a frame and the size and number of soccer objects detected in a shot. Finally, Won [14] proposed the edge histogram descriptor and Kim [15] developed a multi-modal approach for summarizing and indexing news videos. These techniques can be used for shot boundary detection in MPEG-7 soccer video streams.

Some of the above mentioned research activities attempted to analyze the behavior of the players as individuals or as groups and the others tried to make mosaic images. Some of these studies produced better results than our research. For example, Nicolas's paper contains mathematical derivations, which seem to be more advanced than our research, but none of the reviewed research detected the advertisement board in a soccer game. Detecting the advertisement board region provides significant information for making mosaic images without the complex parameter estimation used by Nicolas and by Sun and Tekulp. Some general algorithms have high performance, but they do not produce optimal results in some specific cases, such as a soccer game.

In this paper, we develop a system for mosaicking sequences

of soccer images in a panoramic view for soccer game analysis. The first important component of the mosaicking and soccer game analysis automatically detects and tracks players and some lines, such as the center circle, sideline, and penalty line. The second component consists of a region-matching algorithm for mosaicking, which is based on the advertisement board. The last component finds an image to model a transformation for analyzing plays or team strategy. The rest of the paper is organized as follows. In section II, the soccer field segmentation algorithm is introduced by two steps: ground field extraction and players and lines detection. Section III presents the advertisement region-matching algorithm for mosaicking. Section IV describes the soccer game analysis using a perspective transformation and explains a reverse perspective transform method for deleting hole areas in the target image. Finally, the experimental results and conclusions are summarized in section V.

II. SOCCER FIELD SEGMENTATION

The soccer field provides us with the most significant and reliable information for soccer scene analysis. Since the cameras usually focus on the players carrying the ball, most of the video frames capture only a portion of the field. Therefore, for the first step of soccer game analysis, we extracted the soccer field region and then as the next stage, developed and combined several tools to automatically extract players and lines from the ground region for a soccer sequence. In our approach, we selected soccer frames with over 65% of field pixels rather than the full frame resolution.

1. Field Extraction

The soccer field is predominantly green and occupies the major portion of video sequences. From this knowledge, we make the R, G, and B color histogram for satisfying condition $I_G(x, y) > I_R(x, y)$ and $I_G(x, y) > I_B(x, y)$ to detect ground region information. The peak values R_{peak} , G_{peak} , and B_{peak} of the color histogram represent the field color information on the basis of the assumption that the dominant green color from the input image is the field area. The following equation shows the field extraction method.

$$O(x, y) = \begin{cases} 0, & \text{if } \left\{ \begin{array}{l} I_G(x, y) > I_R(x, y) \text{ AND} \\ I_G(x, y) > I_B(x, y) \text{ AND} \\ |I_R(x, y) - R_{peak}| < R_t \text{ AND} \\ |I_G(x, y) - G_{peak}| < G_t \text{ AND} \\ |I_B(x, y) - B_{peak}| < B_t \text{ AND} \\ G(x, y) < G_{th} \end{array} \right. \\ 255, & \text{otherwise} \end{cases} \quad (1)$$

where $O(x,y)$ is the binarized output image and I_R , I_G , and I_B indicate the R, G, and B values in one pixel, and R_t , G_t , and B_t are the threshold values for R, G, and B. We set these threshold values to 10, 15, 10. These pre-set threshold values will be changed by a large variance of color values because of each frame's illumination condition. Therefore, we control these threshold values according to a deviation variance from R_{peak} , G_{peak} , and B_{peak} of the color histogram. If these deviation values are larger than the pre-set color histogram deviation, then these threshold values can change to higher values than the pre-set threshold values. The $G(x,y)$ is the gray level information and G_{th} refers to the threshold value for $G(x,y)$. The conditions from the 3rd to the 5th are checked as the ground color and the 1st and 2nd conditions are checked as the dominant green color. The final condition discriminates between a line and the ground using gray level information. The line in the ground field should have higher gray level values than the field area threshold value ($G_{th} = 150$). Figure 1(a) shows an example of an input image and (b) shows the result of this equation.

2. Player and Line Component Detection

In the field extraction result shown in Fig. 1(b), the black region indicates the ground region and the white regions include two categories, the player and the line segment. To

separate the two categories, we use gray level information using the *a priori* knowledge that line segments have a higher average gray level distribution than player segments. Figures 2(a) and (b) show the candidate regions of each category using gray level thresholding values.

From these results, we initially process the 2-pass labeling algorithm and Hough transform. These results give some information on the isolated white regions that have centroid, compactness, minimum bounding rectangle (MBR), area (# of pixel), average gray level, average color level of R, G, B, Hough value, and so on. From this information, we can distinguish the player and line segments. The following player rules (Player-Rule) are used to separate player regions.

- Player-Rule 1 (about size): check the size of the MBR and area
- Player-Rule 2 (about shape): check compactness
- Player-Rule 3 (about color): check the R, G, B distribution
- Player-Rule 4 (about elongateness): check the vertical vs. horizontal length

These rules present the characteristic information of the player region in the soccer game image sequence. In addition, the following line rules (Line-Rules) are used to separate line regions.

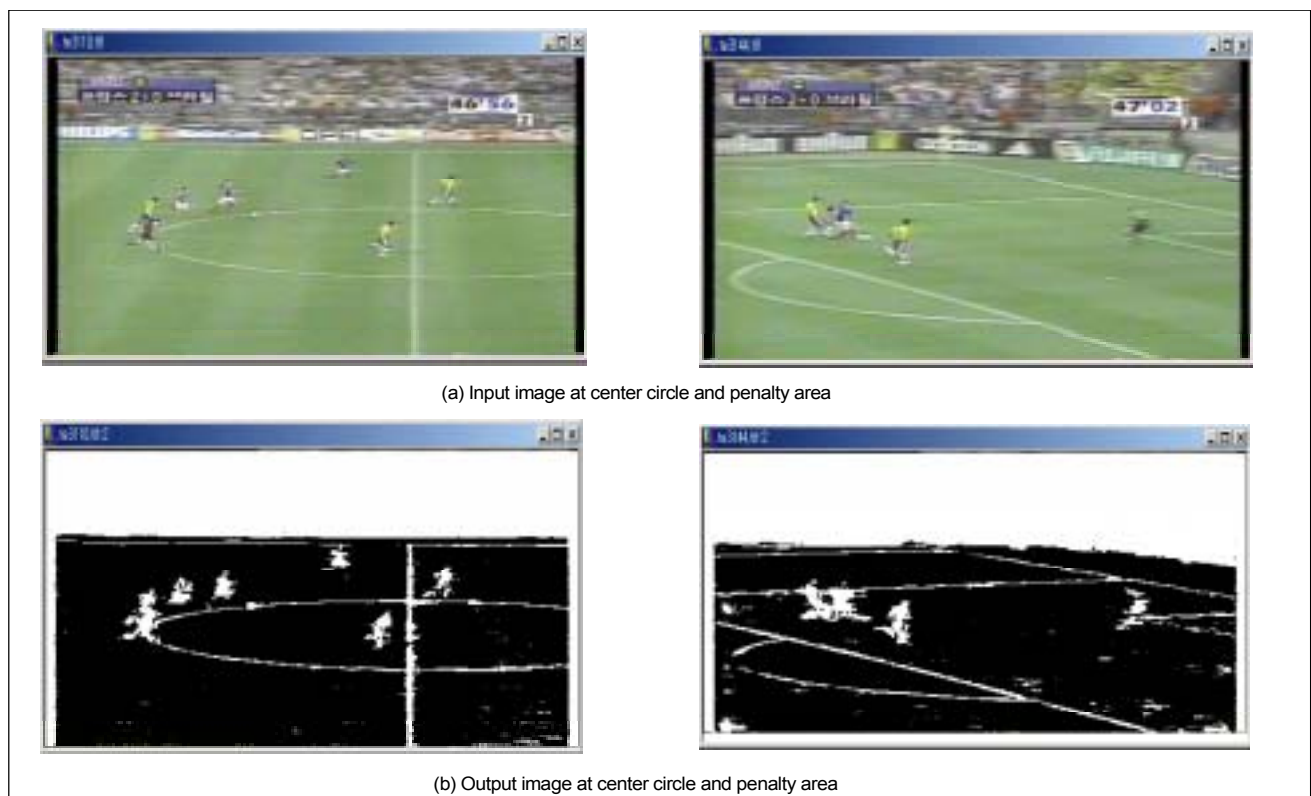


Fig. 1. The result of field extraction algorithm.

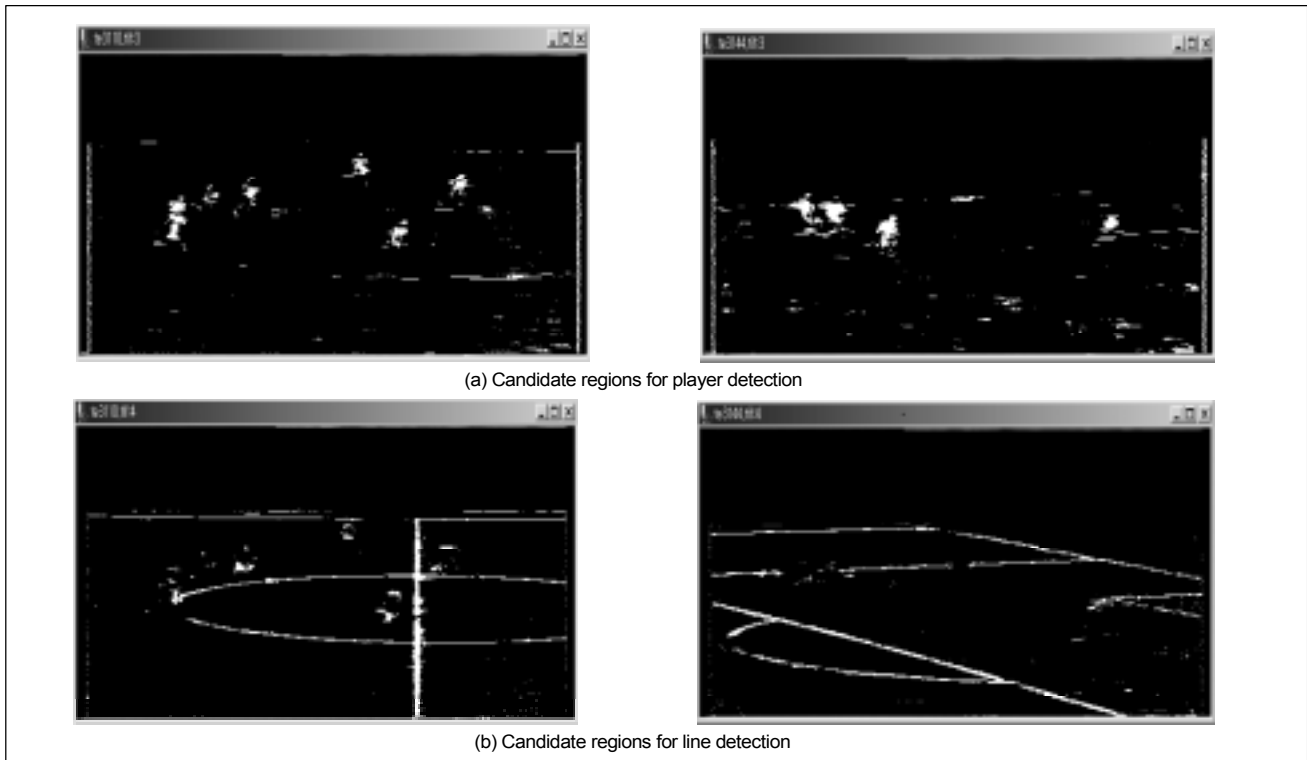


Fig. 2. The results of the player extraction algorithm.

- Line-Rule 1 (about the gray level): check the gray level
- Line-Rule 2 (about line segments): check the Hough value

In most cases, a line in the current image can be detected using the Hough transform [16], [17]. The Hough transform can be used to detect sparsely represented straight lines, arcs of circles, and curvatures. These two types of rules can categorize the candidate regions into line regions. Table 1 shows the threshold values.

Most of the threshold values in Table 1 are variable numbers that depend on camera and illumination conditions. We cannot find the optimal threshold values in all natural cases, but we can detect the optimal threshold values from several games. From these threshold values and experience, we think that the new game's threshold values are more easily detected and processed.

3. Line Classification

Line classification steps are very important for soccer game analysis. The next section's perspective transform algorithm is necessary to predefine three points. According to the line classification step, we can detect the corresponding points. Because we cannot know the current captured viewpoint in a soccer shirt, we will detect candidate lines, such as the centerline, center circle, sideline, and penalty line. If all the lines are detected by the Hough transform in the current image, we can determine the sideline from horizontal line segments. If

Table 1. Threshold values used for field, player, and line extraction.

Rule Number	Range of Threshold Values
Field Extraction	$R_t = 10, G_t = 15, B_t = 10$ and $G_{th} = 150$
Player-Rule 1	$50 < \text{area} < 1,500$ and $130 < \text{MBR} < 5,000$
2	Compactness > 0.3
3	$R > G, B > G$
4	Vertical vs. horizontal length ratio < 3.0
Line-Rule 1	Gray level > 150
2	abr space strength [17] by Hough Transform > 5.0

the captured image has only one horizontal line and one vertical line then these lines are a sideline and a centerline. If the captured image has many lines, then we cannot set all the line information automatically, so we set the line information manually.

In Fig. 3, automatic processing classifies the line information in the left image, and manual and automatic processing classify the line information in the right image.

4. Team Identification

Kawashima [18] analyzed the group behavior of soccer players using color histogram back-projection to isolate the

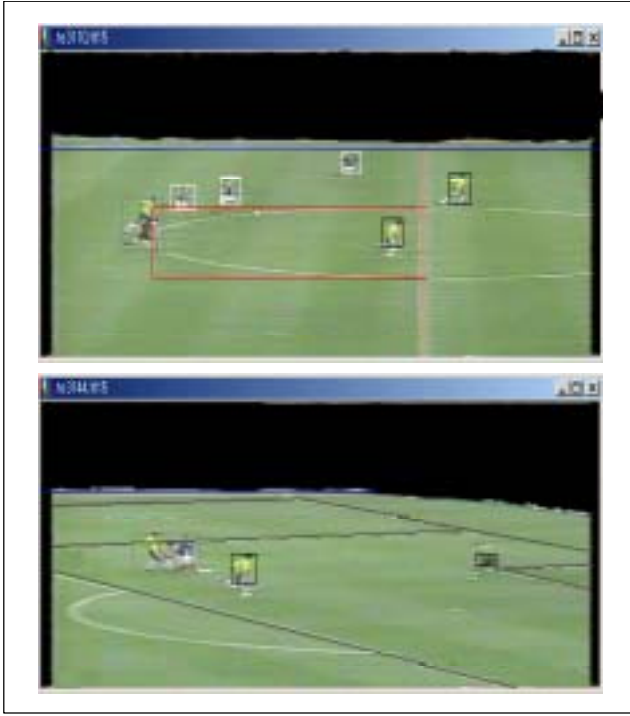


Fig. 3. The results of the player and line extraction algorithm.

players on each team. However, different teams can have similar histograms, so we used a vertical distribution of colors and grays.

Usually, soccer players' uniforms are designed so that the shirt and shorts are differentiated. Also, teams wear uniforms with different colors so spectators can distinguish the movements of each team's players. Thus, distinguishing between the players of two teams can be made by the following steps.

Player's Team Distinction Algorithm

Model Histogram Creation

Step 1: Divide the shirts and shorts of the player area using detection of the stepping point from the vertical gray level histogram.

Step 2: Store the color distribution of each shirt and shorts using the vertical color histogram to model histograms $H^{shirtA}(R,G,B)$ and $H^{shortsA}(R,G,B)$ and $H^{shirtB}(R,G,B)$ and $H^{shortsB}(R,G,B)$, where A and B are the different soccer teams.

Compare Similarity

Step 3: From the entered input image, we can compare the color histogram distribution of each shirt and shorts using the histogram intersection method proposed by Swain and Ballard [19] as follows:

$$S_{shirt,input_up} = \max \left(\frac{\sum \min(H^{shirtA}(R,G,B), H^{input_up}(R,G,B))}{\sum H^{shirtA}(R,G,B)}, \frac{\sum \min(H^{shirtB}(R,G,B), H^{input_up}(R,G,B))}{\sum H^{shirtB}(R,G,B)} \right)$$

$$S_{shorts,input_down} = \max \left(\frac{\sum \min(H^{shortsA}(R,G,B), H^{input_down}(R,G,B))}{\sum H^{shortsA}(R,G,B)}, \frac{\sum \min(H^{shortsB}(R,G,B), H^{input_down}(R,G,B))}{\sum H^{shortsB}(R,G,B)} \right) \quad (2)$$

The $H^{input_up}(R,G,B)$ denotes the upper area histogram distribution and the $H^{input_down}(R,G,B)$ is the lower area histogram distribution of input segments. $S_{shirt,input_up}$ indicates the similarity value between the model and input color histogram distributions as similar $S_{shorts,input_down}$.

Step 4: Determine each player's affiliation using a histogram similarity value. If two similarity values have a higher value from team A, then we will set to A team. Sometimes, if two similarity values have different results, such as $S_{shirt,input_up}$ from A, $S_{shorts,input_down}$ from B, then we select the team with higher similarity values.

Figure 1(a) shows the 1998 France World Cup final match. The French soccer team is wearing blue shirts and white shorts and the Brazilian soccer team is wearing yellow shirts and blue shorts. From the player's team distinction algorithm, we set the white box to the French team and the black box to the Brazilian team. The dotted black box indicates a non-terminative area that includes a merged player.

5. Player Tracking

We use template matching for player tracking. The templates of players are extracted from the first image or new images that have new players using results of player detection in section II. New player templates that do not significantly overlap with the bounding box are registered in the tracking list. Locations of players at the next frame are calculated by template matching based on the center points of the nearest neighborhood in the next frame and team information. Next, player templates are updated with new templates.

The main problem of player tracking is occlusion. In our approach, we only track a player when occlusion occurs between different teams. When two players are merged in one connected area, we make a new template including two players and trace this template (occlusion reasoning). In addition, we make two new player templates when this template is divided into two separate players according to team information. Figure 4 shows the whole procedure for tracking players.

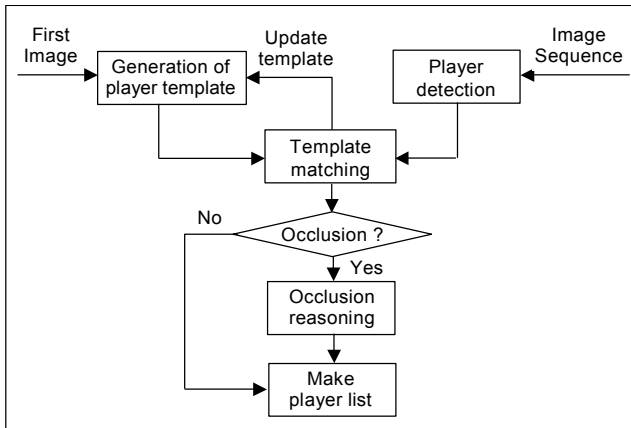


Fig. 4. The flowchart of the tracking procedure.

III. ADVERTISEMENT REGION MATCHING ALGORITHM FOR MOSAICKING

In section II, we described how to detect a line segment when some lines are visible. If we detect the centerline and sideline from two successive frames, then we can make a mosaic image that merges one image using centerline and sideline coordinates. But if there is less than a two-line component or the line component is not visible, we cannot make the mosaic image. To solve this problem, we use the advertisement board region matching algorithm. Generally, some soccer game images have advertisement board regions. Figure 5(a) shows the last frame, which can be applicable to mosaicking using centerline and sideline information and (b) shows the first frame, which can be applied to the region matching algorithm.

For the region matching algorithm, we have to detect the advertisement board region. The following steps describe the detection algorithm of the advertisement board region.

Advertisement Board Region Detection Algorithm

Step 1: Define the ground region using the results of the ground extraction algorithm.

Step 2: For the intermediate region between the ground and advertisement board, α pixels are skipped. The α is determined by variance checking of the color distribution. From the ground to the advertisement board, we examine the changing of the color distribution from the horizontal line. The boundary of the color distribution between the ground and the advertisement board is set to α .

Step 3: Search the upper area with β pixels heights from the skipped region. This algorithm calculates the color distance between the current and next line. If the color distance has a

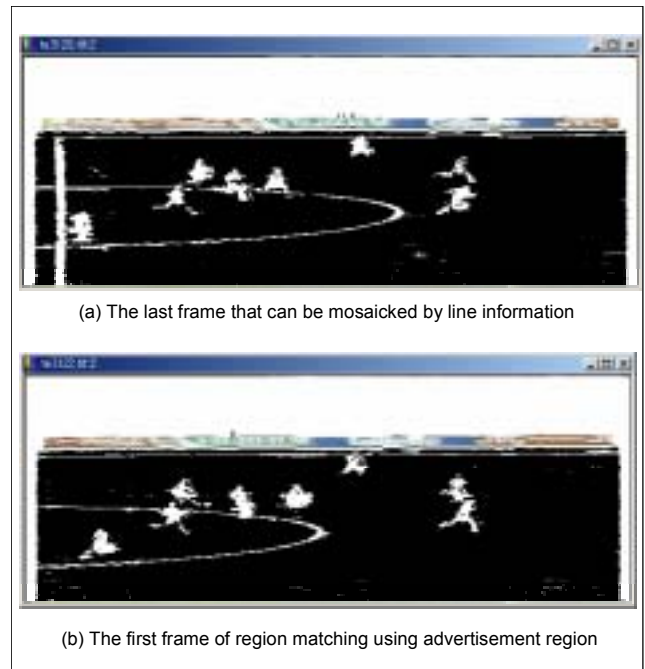


Fig. 5. The extracted advertisement region (color region only).

bigger value than the threshold value, then this position is the advertisement region with a β pixel height (Fig. 5).

Our advertisement region matching algorithm is motivated by the *a priori knowledge* that the brightness value of corresponding points is the same during t and $t+1$.

$$I(x', y', t+1) = I(x, y, t) \quad (3)$$

where (x', y') indicates points at time $t+1$ corresponding to (x, y) at time t and I is the gray level value in each coordinate. This problem can be defined as the searching method of error function $E^y(t)$ in (4).

$$E^y(t) = \sum_{y=1}^n \{I(x', y', t+1) - I(x, y, t)\}^2 \quad (4)$$

The error function $E^y(t)$ indicates the summation of the gray level difference between two intersected pixels between time t and $t+1$ during y -direction shifting. The final problem can be defined as a searching problem from $y = 1$ to $y = n$ that has a minimum value of $E^y(t)$, where n denotes the width of the advertisement board. In our experiment, most of the captured images have a left or right panning camera operation, so we can find the minimum $E^y(t)$ easily. However, this algorithm has some errors when the captured image is skewed or tilted by the camera operation. To solve this problem, we adapt the angle and size normalization algorithm

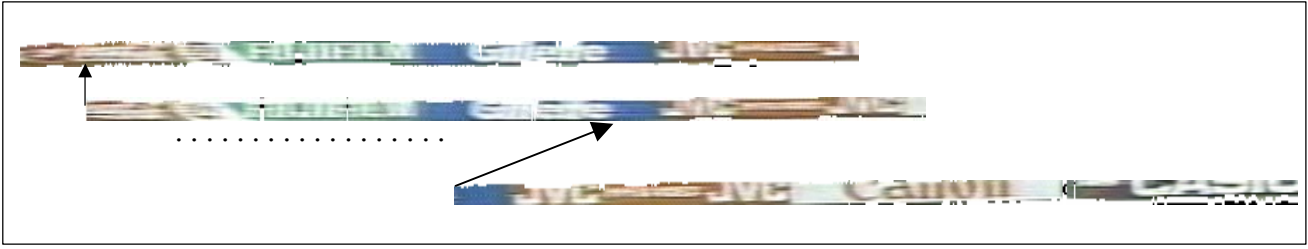


Fig. 6. Matched points between frames.



Fig. 7. Mosaic image from the 1st to 40th frames.

using previously detected end positions of the advertising board. These factors can be transferred to the perspective transformation in section IV.

Figure 6 explains the matched points between successive image frames using this algorithm.

Using the results of the advertisement board matching algorithm, a panoramic view image is generated by mosaicking the successive frames (Fig. 7). We can display the extracted player in a panorama image continuously from the starting frame to the final frame. This panoramic image provides an augmented reality, such as the view of the audience in the soccer stadium.

IV. PERSPECTIVE TRANSFORMATION FOR GAME ANALYSIS

1. Perspective Transformation

In this part, we describe the game analysis method using the perspective transform. We will create a field model and describe how to get a transformation parameter between the extracted shirt region and the ground model. Because camera operations, such as panning and zooming, are adopted for capturing images, the viewpoint will change from frame to frame. To find a player's position on the ground model, we use perspective transforms that are a subset of the perspective transformation algorithm [20].

Perspective transformation includes scale, rotations, translations, or any combination of these. Perspective mapping may be specified geometrically by three points. We can use a grid of triangles for our control points. With perspective mapping, we can map triangles into triangles. Those are the only distortions that can be mapped with perspective transformations. The forward mapping functions are

$$x = a_{11}u + a_{12}v + a_{31}, \quad (5)$$

$$y = a_{21}u + a_{22}v + a_{32}. \quad (6)$$

The coefficients are determined by solving the system of six linear equations:

$$\begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix} = \begin{bmatrix} u_0 & v_0 & 1 \\ u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}. \quad (7)$$

The $x_0, y_0, x_1, y_1, x_2,$ and y_2 describe the coordinates of the destination triangle. The $u_0, v_0, u_1, v_1, u_2,$ and v_2 are the coordinates of the source triangle (Fig. 8).

To determine the coefficients, we must isolate their matrix by multiplying both sides by the inverse of the matrix containing the source reference coordinates. The inverse of a matrix is defined as its adjoint divided by its determinant. This equation is

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix} \begin{bmatrix} v_1 - v_2 & v_2 - v_0 & v_0 - v_1 \\ u_2 - u_1 & u_0 - u_2 & u_1 - u_0 \\ u_1 v_2 - u_2 v_1 & u_2 v_0 - u_0 v_2 & u_2 v_1 - u_1 v_0 \end{bmatrix} \begin{bmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix} = \frac{\det UV}{\det UV} \quad (8)$$

$$\text{where } \det UV = u_0(v_0 - v_2) - v_0(u_1 - u_2) + (u_1 v_1 - u_2 v_1). \quad (9)$$

From (9), a perspective transformation algorithm calculates six variables from a_{11} to a_{32} using the corresponding three points as shown in Figs. 8(a) and (b). Using this equation, the ground model image generation algorithm proceeds to the next steps:

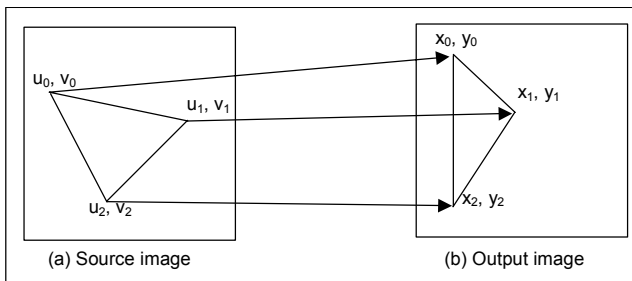


Fig. 8. Coordinates of source and destination triangles.

Ground Model Image Generation Algorithm

Step 1: Select the three points from the captured image coordinates.

Step 2: Set the three points from the ground model coordinates corresponding to the three points from the captured image.

Step 3: Calculate the six variables from a_{11} to a_{32} using (9).

Step 4: All captured image points are mapped to the ground model coordinates using (8).

Figure 9 shows all the candidate points for a perspective transform that can be selected by three pre-defined points.

The center circle, centerline, and sideline detection algorithm described in section II is used to select the position of the three points from step 1 above. Figure 10(a) shows the results of the detected lines and selected three points, and Fig. 10(b) describes the corresponding three points in the ground model. Figure 10(c) shows the results of the perspective transformation.

2. Reverse Perspective Transform

As Fig. 10(c) shows, the perspective transform results in many holes in the model coordinates. There are two problems with forward mapping: holes and overlaps. Holes are pixels

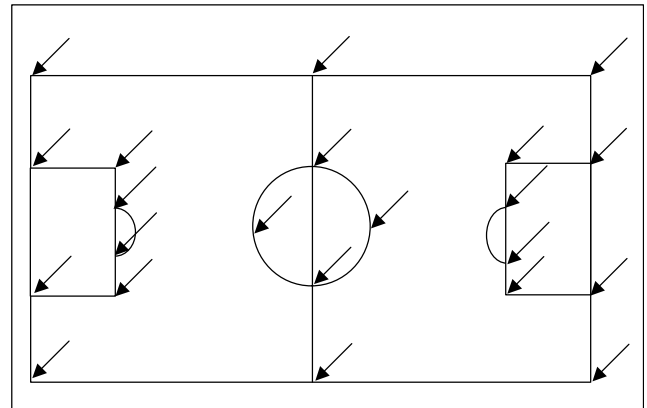


Fig. 9. All the candidate points for a perspective transform.

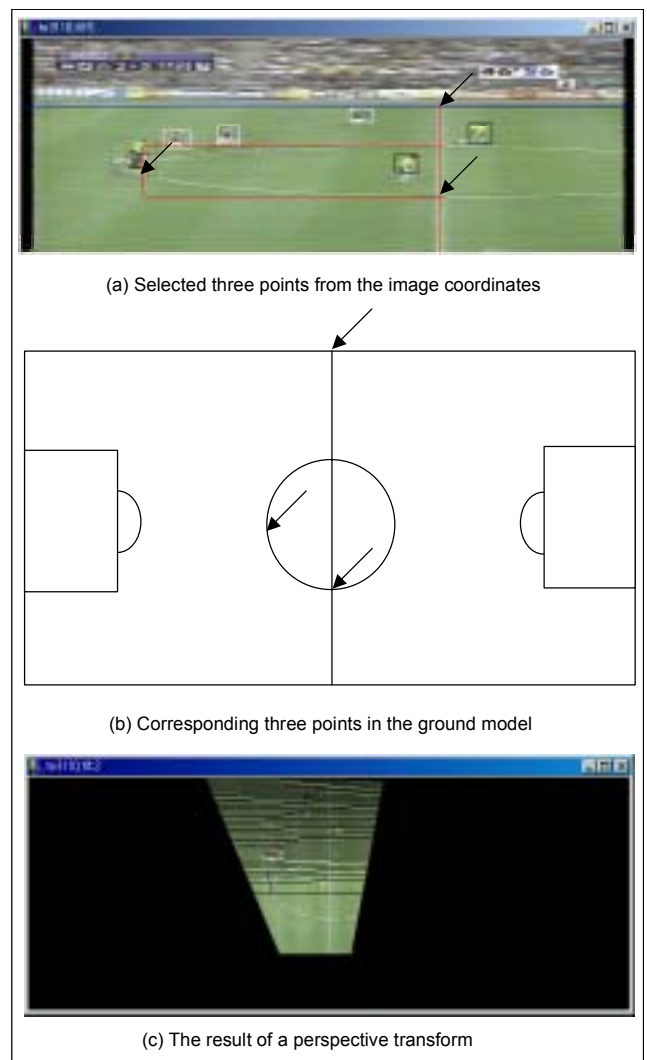


Fig. 10. The steps for ground model generation.

that are undefined, and the destination pixel has no corresponding source pixel. Overlaps occur when two input

pixels get mapped to the same output pixel. There are several interpolation algorithms that can solve this problem, such as nearest neighbor, bilinear, higher order, cubic convolution, and B-spline, but a more efficient method is to reverse the transform.

Reverse mapping traverses the destination image and calculates with some inverse transform in which a pixel in the source image will be used to produce the destination pixel. Computing the destination image in this manner eliminates the problems of holes and overlaps. Figure 11 shows the mapping direction for the coordinate transform and Fig. 12 shows the results of the reverse perspective transform.

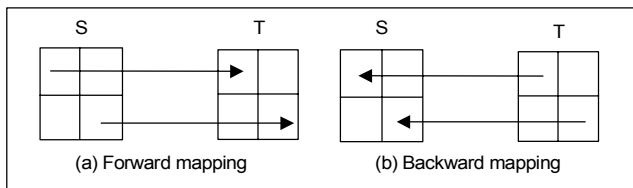


Fig. 11. The mapping direction.



Fig. 12. The result of the reverse perspective transform.

The reverse mapping is computed by

$$[u \ v \ 1] = \frac{[x \ y \ 1]}{a_{11}a_{12} - a_{21}a_{12}} \times \begin{bmatrix} a_{22} & -a_{12} & 0 \\ -a_{21} & a_{11} & 0 \\ a_{21}a_{32} - a_{31}a_{22} & a_{31}a_{12} - a_{11}a_{32} & a_{11}a_{22} - a_{21}a_{12} \end{bmatrix} \quad (10)$$

This equation will perform a perspective transformation on a complete image as shown in Fig. 11.

V. CONCLUSIONS

An overview of our approach is illustrated in Fig. 13. The scene selection module detects the soccer video sequence for ground mosaicking by using automatic detection or pointing by the user. This module, which is about scene change detection

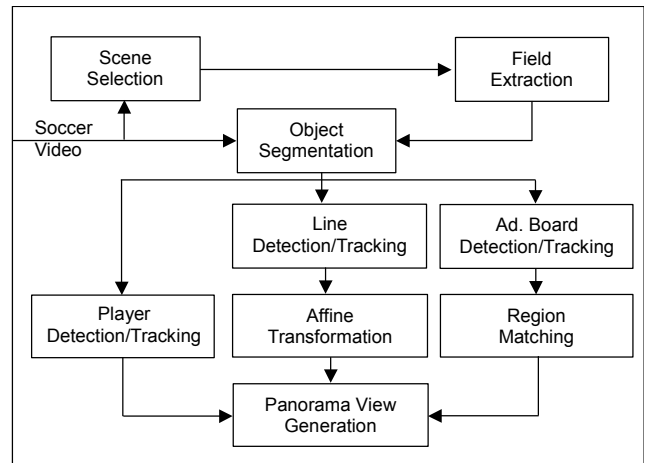


Fig. 13. The system overview.

and scene understanding, is not included in this topic but the field extraction module provides some information about scene understanding.

The first stage of this algorithm divides the field region and the object region. The field extraction algorithm detects the ground field using domain dependent color segmentation. The object segmentation algorithm consists of the three stages: player, line, and advertisement board detection/tracking processing. The results of the line processing steps adapt an affine model and transfer each line's coordinates from the ground to the mosaic. The results of the advertisement board processing steps using the region-matching algorithm also transfer object coordinates from the ground to the mosaic, while not using line information. Finally, the panorama view generation module makes the panoramic scene using segments of each player and coordinates from the affine model or region matching steps.

1. Experimental Results

The proposed system was implemented on a personal computer with an image capture board (Matrox Meteor II). Input image sequences were captured by a VHS tape from a VTR with a resolution of 640×240 and 15 frames per second. The computing power was 3-5 frames per second using a dual Pentium II 450 MHz processor. Since our computer could process over 5 frames per second, we could view the proposed system for real-time mosaic generation. The developed software was implemented in Visual C++ 6.0 on Windows 98.

Figure 14 shows the successive results of the player and line extraction algorithm. During the 9th frame, we could make the mosaic image using line information and the mosaic image using the advertisement region matching algorithm from the



Fig. 14. The successive results of the player and line extraction algorithm.



Fig. 15. The mosaic image with the top one player's motion.

10th frame to the final frame. Figure 6 shows the mosaic image from the 1st to the 40th frames and Fig. 15 shows the mosaic image with the top player from the initial frame. For soccer

game analysis, all players' trajectories can be displayed in the ground model coordinates using the perspective transform. Figure 16 shows the final results for the soccer game analysis.

Each line denotes each player's trajectory. The two teams are distinguished by dash lines and straight lines.

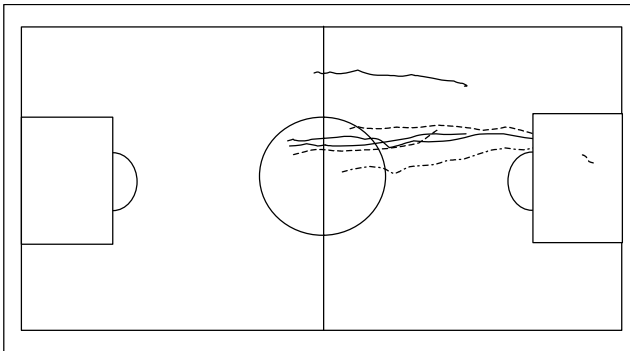


Fig. 16. The final results for the soccer game analysis.

2. Conclusion

The major portion of the essence of a soccer game is captured in relatively short periods of intense action. These highlights are summaries of the games. The sports highlights, such as movements of exciting action and shots, are subjective in nature; in a soccer game, these action shots are most likely found when a goal is scored or there is a near miss. After that highlight moment or after the first-half period, the TV broadcasters replay these highlight scenes. The sportscaster needs new scenes to explain the details of the situation in the highlight scenes. For this purpose, we developed a system for mosaicking sequences of soccer images in a panoramic view for soccer game analysis. Although our system cannot cover all situations of highlight scenes because of the variance of camera parameters and the existence of the advertisement board, we think that our results are very useful for a real-time replay system.

Our algorithm for mosaicking sequences of soccer images used a region matching algorithm for panoramic view generation and analysis. We detected the advertisement board from each frame and used this information for detecting the intersection region. This region matching algorithm generated a panoramic view image by mosaicking the successive frames. Finally, we can display the extracted player in the panoramic image continuously from the first frame to the last frame. This panoramic image provides an augmented reality, such as the view of the audience in the soccer stadium. We also applied the perspective transform to the panorama image; this information is useful to the coach or broadcaster for analyzing how goals occurred and how players moved. The general perspective transform has two problems: holes and overlaps. To solve these problems, we used a reverse perspective transform. The reverse perspective transform is a more time-consuming process

because the matrix calculation is twice that of general processing, but the results have better quality information than other interpolation algorithms. We tested our method with real image sequences from the 1st to 40th frames. Our results produced a panoramic image and provided information on the players' trajectories.

In future studies, we will develop an algorithm that can deal with more complex problems, such as occlusion reasoning of more complex cases, tracking the ball position, automatic scaling of player size, analysis of player motion (kick, dribble, run, walk, shoot), and virtual 3D replay.

REFERENCES

- [1] Y. Gong, T.S. Lim, H.C. Chuan, H.J. Zhang, and M. Sakauchi, "Automatic Parsing of TV Soccer Programs," *IEEE Trans. on Multimedia*, May 1995, pp. 167-174.
- [2] S. Intille and A. Bobick, "Visual Tracking Using Closed-Worlds," *Perceptual Computing Section Technical Reports*, MIT Media Lab, no. 294, Nov. 1994.
- [3] L. Teodosio and W. Bender, "Salient Video Stills: Content and Context Preserved," *ACM Multimedia*, June 1993, pp. 39-46.
- [4] D. Yow, B.L. Yeo, M. Yeung, and B. Liu, "Analysis and Presentation of Soccer Highlights from Digital Video," *The Second Asian Conf. in Computer Vision (ACCV'95)*, vol. 7, no. 1, 1991, pp. 11-32.
- [5] M. Irani, D. Anandan, and S. Hsu, "Mosaic Based Representation of Video Sequences and Their Application," *Int'l Conf. on Computer Vision*, 1995, pp. 605-611.
- [6] Y. Seo, S. Choi, H. Kim, and K.S. Hong, "Where are the Ball and Players? Soccer Game Analysis with Color-Based Tracking and Image Mosaic," *Int'l Conf. on Image Analysis and Processing*, Florence, Italy, Sept. 1997, pp. 196-203.
- [7] T. Taki, J.I. Hasegawa, and T. Fukumura, "Development of Motion Analysis System for Quantitative Evaluation of Teamwork in Soccer Games," *IEEE Int'l Conf. on Image Processing*, vol. III, 1996, pp. 815-818.
- [8] K. Matsui, M. Iwase, M. Agata, T. Tanaka, and N. Ohnishi, "Soccer Image sequence Computed by a Virtual Camera," *IEEE Conf. on Computer Vision and Pattern Recognition*, 1998, pp. 860-865.
- [9] T. Bebie and H. Bieri, "SoccerMan-Reconstructing Soccer Game from Video Sequence," *IEEE Int'l Conf. on Image Processing*, 1998, pp. 898-902.
- [10] H. Nicolas, "New Methods for Dynamic Mosaicking," *IEEE Trans. on Image Processing*, Aug. 2001, pp. 1239-1251.
- [11] Z. Sun and A. Tekulp, "Trifocal Motion Modeling for Object-Based Video Compression and Manipulation," *IEEE Trans. on Circuits and Systems for Video Technology*, Sept. 1998, pp. 667-685.
- [12] P. Xu, L. Xie, S-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and System for Segmentation and Structure Analysis in Soccer Video," *IEEE Int'l Conf. on Multimedia and Expo*, Aug.

2001.

- [13] A. Ekin and A.M. Takalp, "A Framework for Tracking and Analysis of Soccer Video," *Symp. Electronics Imaging: Science and Technology: Visual Communication and Image Processing*, Jan. 2001, pp. 763-774.
- [14] C.S. Won, D.K. Park, and S.-J. Park, "Efficient Use of MPEG-7 Edge Histogram Descriptor," *ETRI J.*, vol. 24, no. 1, Feb. 2002, pp. 23-30.
- [15] J.-G Kim, H.S. Chang, Y.T. Kim, K. Kang, M. Kim, J. Kim, and H.-M. Kim, "Multimodal Approach for Summarizing and Indexing News Video," *ETRI J.*, vol. 24, no. 1, Feb. 2002, pp. 1-11.
- [16] E. Gose, R. Johnsonbauch, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall PTR, pp. 352-357.
- [17] S.G Park, S.Y. Kim, J.Y. Kim, and S.J. Kim, "Finding the Information of the Ellipse from the Optical Hough Transform," *Proc. of SPIE- the Int'l Society for Optical Engineering*, vol. 4043, 2000, USA, pp. 352-363.
- [18] T. Kawashima, K. Yoshino, and Y. Aok, "Qualitative Image Analysis of Group Behavior," *Proc. of Computer Vision and Pattern Recognition*, June 1994.
- [19] M.J. Swain and D.H. Ballard, "Color indexing," *Int'l Journal of Computer Vision*, 1991, pp. 11-32.
- [20] R. Crane, *A simplified Approach to Image Processing*, Prentice Hall PTR, 1997, pp. 206-211.



Ho-Sub Yoon received the BS and MS degrees in computer science from SoongSil University, Seoul, Korea, in 1989 and 1991. Joining the Electronics and Telecommunications Research Institute (ETRI) in 1991, he is currently a Senior Research Scientist at the Computer & Software Research Laboratory (CSL). He is also a PhD student at KAIST in Daejeon. His research

interests are pattern recognition and computer vision.



Young-lae J. Bae He received the BSc degree in marine physics from Seoul National University, MSc degree in computer sciences from Hanyang University, Seoul, and PhD degree in electronic engineering from University of Kent, England, in 1976, 1986 and 1995. From 1976 to 1979 he was a naval officer. From 1980 to 2002 he was with Computer &

Software Research Laboratory / ETRI, where he was the Leader of the Visual Information Processing Research Team, a National Research Laboratory. He was also a part time Professor at Pai Chai University. As a Visiting Scientist, he was with the Informatik Institut, Technische Universität München, Germany, from 1988 to 1989. He is now a Professor at Chungbuk Provincial Univ. of Science & Technology. His major research interests include image processing, biometrics, electronic commerce, parallel processing, and pattern recognition. He received the President's Excellent Scientist Award (1985) and Excellent Paper Awards (Korea Information Processing Society, 2000 and 2001).

Dr. Bae has been a regular member of the IEEE PAMI, IEEE GRSS, IEE, KITE, KISC, KSRS.



Young-kyu Yang received his BS and MS degrees in Seoul National University in 1972, 1974. He received the PhD degree in satellite image processing from Texas A&M University in 1985. He joined KIST (Korea Institute of Science and Technology)/SERI (Systems Engineering Research Institute) in 1973 and transferred to ETRI in 1996. Most recently, he worked as the Director of the Spatial Information Technology Center of Computer and Software Research Laboratory and is currently working as a Visiting Researcher at University of California at Irvine (UCI), USA. He is also serving as the President of the Korea Society of Remote Sensing (April 2001 to March 2003) and Honorary President of the Korea Open GIS Society (April 2002 to March 2004). His research interests include spatial information processing (GIS, RS, GPS), image processing, and Location Based Service (LBS) systems.