Video2Cartoon: A System for Converting Broadcast Soccer Video into 3D Cartoon Animation

Dawei Liang, Qingming Huang, *Member*, IEEE, Yang Liu, Guangyu Zhu,
and Wen Gao, *Senior Member*, IEEE

Abstract — *A system for converting broadcast soccer video into 3D cartoon animation is proposed. The system can provide viewers with a new experience that can not be acquired from the original soccer video, by taking advantage of computer vision and computer graphics techniques. Firstly, computer vision techniques are employed to estimate the 3D positions of players and the ball. Then, players, the ball and the playfield are modeled by computer graphics techniques. Finally, 3D cartoon animation is generated based on the extracted 3D information and the pre-constructed player motion database. The system allows users to watch the game at arbitrary viewpoint using a virtual camera. On the other hand, content service providers can distribute the generated cartoon animation on their web portals or to mobile devices, for consumers to better enjoy the game*[1].

Index Terms — Broadcast Soccer Video, 3D Cartoon Animation, Object Detection and Tracking, Camera Calibration, 3D Position Estimation.

I. Introduction

Soccer has been one of the most popular sports games around the world, and has attracted hundreds of millions of people. Every year thousands of soccer games are held worldwide, and a huge amount of broadcast soccer video is recorded to facilitate further transmission, browsing, editing, etc. In most cases, the recorded soccer video only shows a single viewpoint of the game, though many cameras are deployed around the playfield during the game. Some viewers may wish to see how the ball streaks towards the goalmouth from the goalkeeper's viewpoint. Others may want to have a bird's eye view of a goal. Moreover, many web portals provide net surfers with cartoon animation of goal events after important soccer games. The cartoon animation is generated mainly manually, which is labor-intensive and tedious. In addition, it only shows a single viewpoint of the game. Therefore, developing a system that can convert broadcast soccer video into 3D cartoon animation is of great value, which can relieve human burdens to a large extent, and can provide consumers with free viewpoint of the game.

There are only a few related works in the literature, e.g., [1]-[3]. Matsui et al. [1] proposed an image synthesis system with broadcast soccer video as input. Their system can generate computer graphics animation from the viewpoint of any player; however, soccer ball is not considered in their system. Bebie and Bieri [2] presented the SoccerMan system, which can generate an animated 3D scene from a soccer video sequence. In their system, player is modeled as the so-called animated texture object, i.e., quadrilateral holding the player's texture. Such a representation limits the available viewpoint, since a texture object is seen distorted when moving the viewpoint away from the original one. Moreover, soccer ball is not considered either. Recently, Yu et al. [3] presented a 3D reconstruction and enrichment system for broadcast soccer video, in which not only the goalmouth but also the midfield scene can be reconstructed. However, their system can just provide the main camera's viewpoint. It is also worth noting that 3D reconstruction of dynamic events from multiple cameras has attracted much attention [4]. 3D reconstruction from multiple cameras can recover much more 3D information, and make the reconstructed 3D scene more realistic. However, multiple cameras system is very costly, and is not easy to deploy. Compared with multiple cameras video, broadcast soccer video has the advantage of being easily acquired.

In this paper, we present a system Video2Cartoon, which can generate 3D cartoon animation from broadcast soccer video, and allows users to watch the game from arbitrary viewpoint using a virtual camera. The main advantages of our system are as follows. First, players and the ball are detected and tracked effectively. Second, by employing global motion estimation, the camera can be calibrated even when feature points on the playfield are insufficient. Third, the ball's 3D positions can be estimated by camera self-calibration, under the assumption that the ball follows a parabolic trajectory in the air. Fourth, players are modeled according to H-anim1.1 [5], which is the specification for a standard humanoid. Last but not least, to enhance viewing experience, the playfield is not only texture-mapped by image patches extracted from the playfield region, but decorated with billboards, auditoria, etc.

[1] This work was supported in part by NEC Laboratories China, and in part by National Hi-Tech R&D Program (863 Program) of China under grant No. 2006AA01Z117. The work described in this paper was done in the Graduate School of Chinese Academy of Sciences, Beijing, China.

D. Liang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China (e-mail: dwliang@jdl.ac.cn).

Q. Huang is with the Graduate School of Chinese Academy of Sciences, Beijing, 100080, China. And he is also with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China (e-mail: qmhuang@jdl.ac.cn).

Y. Liu, and G. Zhu are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China (e-mail: yliu@jdl.ac.cn, gyzhu@jdl.ac.cn).

W. Gao is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China. And he is also with the Institute of Digital Media, Peking University, Beijing, 100871, China, and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China (e-mail: wgao@jdl.ac.cn).

Contributed Paper
Manuscript received July 12, 2007

0098 3063/07/$20.00 © 2007 IEEE

The rest of the paper is organized as follows. Section II provides an overview of the proposed system. In section III, 3D information extraction is introduced. Section IV presents the method for generating 3D cartoon animation. Section V provides some experimental results. In the last section, we draw some conclusions with discussions on future work.

II. System Overview

Fig. 1 provides an overview of the proposed system, which consists of two main modules: 3D information extraction and 3D cartoon animation generation. By utilizing computer vision techniques, the first module estimates the 3D positions of players and the ball. First, players and the ball are detected and tracked. Then, the camera is calibrated for each frame. Finally, the 3D positions of players and the ball are estimated. In the second module, computer graphics techniques are employed to generate 3D cartoon animation. First, playfield is modeled according to the Laws of the Game of FIFA [6]. Second, player is modeled based on H-anim1.1 [5], which is the specification for a standard humanoid. Third, 3D cartoon animation is generated based on the extracted 3D information and the pre-constructed player motion database.

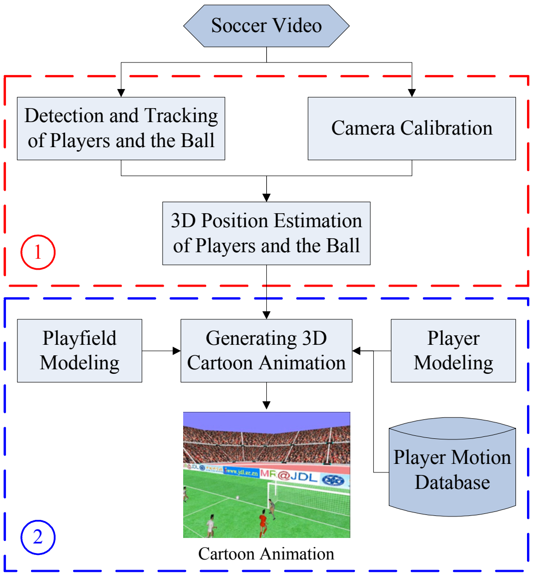
Soccer Video
Detection and Tracking of Players and the Ball
Camera Calibration
3D Position Estimation of Players and the Ball
1
Playfield Modeling
Generating 3D Cartoon Animation
Player Modeling
Player Motion Database
2
Cartoon Animation

Fig. 1. An overview of the proposed system.

III. 3D Information Extraction

3D information extraction from monocular video is generally a difficult task, since information is lost when projecting 3D scene onto 2D image plane. To make the task tractable, some prior knowledge is needed. In soccer domain, the lengths of mark lines within penalty area are known; therefore, the camera can be calibrated, and the corresponding position in world coordinate system of the pixel within playfield can be computed. In what follows, we introduce the key steps in the module of 3D information extraction in detail.

A. Players Detection and Tracking

In this subsection, a framework for multiple players detection and tracking is proposed as shown in Fig. 2. For each video frame, the playfield region is segmented firstly. Then, non-playfield regions are extracted and further classified as players and non-players by supervised learning. After that it is judged whether each identified player region is a new player. If new player appears, a tracker is assigned and initialized. Otherwise, player tracking is conducted. Finally, each tracked player region is verified by a classifier, which is the same as the one used in player detection step. If a tracked player region is classified as non-player region in several consecutive frames, the tracker is released. In the following, the methods for player detection and tracking are introduced.

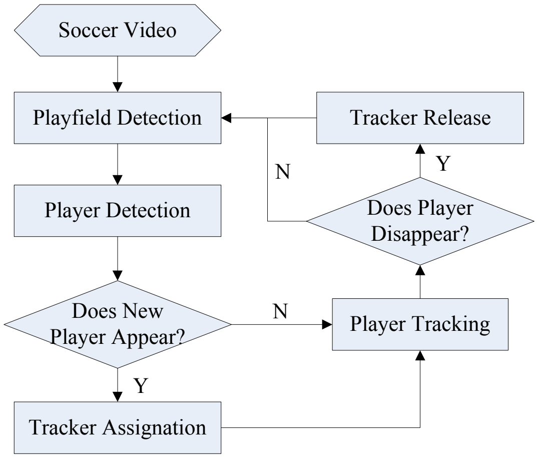

Fig. 2. The framework of multiple players detection and tracking.

1) Player Detection

In soccer video, players usually locate in the playfield region, and the playfield color is the dominant color in most cases. These constraints simplify and accelerate player detection. Firstly, the playfield region is segmented by an efficient method [7], in which Gaussian mixture model (GMM) is used to model the playfield color. The parameters of GMM are estimated by Expectation-Maximization (EM) algorithm from video frames which are sampled online. Then, region growing [8] is employed to extract connected components within the playfield region. After filtering out some noisy regions, the left ones are fed into a classifier to identify whether they are player regions or not.

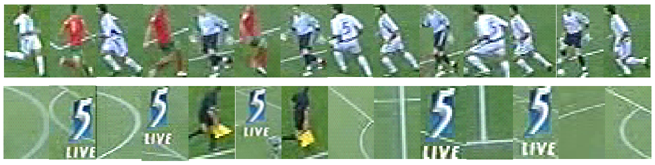

Fig. 3. Training samples of SVM for player detection.

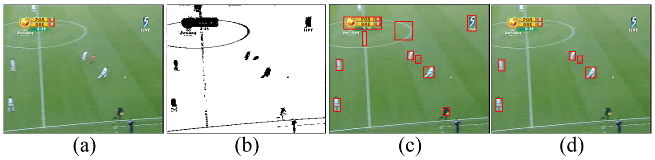

Fig. 4. Player detection results. (a) the original frame. (b) after playfield detection. (c) after post-processing. (d) after support vector classification.

Support vector machine (SVM) [9] has been extensively used in pattern recognition community, and has demonstrated good performance in small sample size. Here, SVM is adopted as the classifier. The feature used for SVM is the region's color histogram in Hue-Saturation-Value (HSV) color space [10]. Since color information is only reliable when both the saturation and the value are not too small, only those pixels whose saturation and value are above certain thresholds contribute to HS histogram, and the left ones contribute to V histogram. Then, the two histograms are concatenated to form the final feature vector. Some training samples are shown in Fig. 3. Fig. 4 provides the procedure of player detection.

2) *Player Tracking*

Among a large body of tracking algorithms, particle filter [11] has demonstrated good performance due to its ability to deal with nonlinear and non-Gaussian cases. Moreover, it is robust to partial occlusion and background clutter by virtue of its ability to maintain multiple hypotheses. Color-based particle filter [10] is used to track each player. The design of particle filter contains two main parts: dynamic model and observation model. For dynamic model, first order autoregressive model is adopted. For observation model, color histogram which is the same as the one used in player detection is employed. Moreover, we improve particle filter by support vector regression (SVR) [9], and term the improved particle filter as SVR particle filter [12]. The basic idea is to apply SVR after particle weighting step to obtain a regression function, which is then used to re-weight particles. Since neighborhood particles are considered in regression, the influence of some noisy particles can be relieved.

B. *Ball Detection and Tracking*

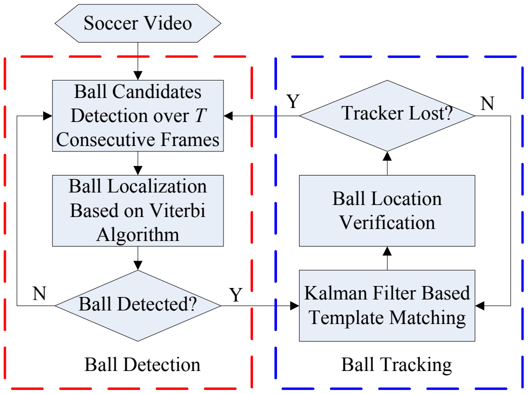
Soccer Video
Ball Candidates Detection over *T* Consecutive Frames
Ball Localization Based on Viterbi Algorithm
Ball Detected?
N
Y
Ball Detection
Tracker Lost?
Y
N
Ball Location Verification
Kalman Filter Based Template Matching
Ball Tracking

Fig. 5. The flowchart of ball detection and tracking.

Ball detection in broadcast soccer video is a challenging task mainly due to the following factors. First, the ball's color, shape, and size change over frames. Second, the ball is sometimes occluded by players, or merged with mark lines. Third, many other objects are similar in appearance to the ball, such as some regions of players. To enhance the robustness of ball detection, temporal information should be considered. In this subsection, we propose a scheme for ball detection and tracking as shown in Fig. 5. In ball detection procedure, color, shape and size are used to extract ball candidates in several consecutive frames. Then a weighted graph is constructed, with each node representing a candidate and each edge linking two candidates in adjacent frames. Finally, Viterbi algorithm is applied to extract the optimal path which is most likely to be the ball's path. Once the ball is detected, tracking procedure based on Kalman filter and template matching is started. Kalman filter and the template are initialized using ball detection results. In each tracking step, the ball's location is verified to update the template and to guide possible ball re-detection.

1) *Ball Detection*

The basic idea of ball detection is to hold multiple hypotheses of the ball's locations by graph. We extract the optimal path of the graph as the ball's locations rather than identify whether a single object in a single frame is the ball or not, since many objects are similar in appearance to the ball.

Based on the observation that the ball's color is nearly white in far-view video, white pixels are first segmented by (1) in normalized RGB color space, i.e., rgb color space.

$$B(x,y)=\begin{cases}1 \quad (r(x,y)-\frac{1}{3})^2+(b(x,y)-\frac{1}{3})^2\le a^2 \wedge I(x,y)\ge c \\ 0 \quad otherwise\end{cases}, \quad (1)$$

where $B(x,y)$ is a binary image, and $I(x,y)$ is the intensity value of the pixel (x,y). The thresholds are set to be $a=0.05$ and $c=160$ empirically. After filtering out noises by morphological close operation, region growing [8] is used to extract connected components. Then, several features are used to obtain ball candidates, including the object's size, the ratio of the length and the width of the object's minimal bounding rectangle (MBR), the area ratio of the object and its MBR. In order to adapt to various ball appearances, the thresholds are set as loosely as possible. In our experiment, the threshold of the first feature is set as differently as the object appears at different image position, the threshold of the second one is set to be 1.5, and the threshold of the third one is set to be 0.5.

After candidates detection over T consecutive frames, a weighted graph is constructed. Each graph node represents a ball candidate. Each graph edge links two candidates i and j in adjacent frames t and $t+1$, respectively, whose Euclidean distance $d_{i,j}^t$ is smaller than the threshold $d_{\max}$. By (2) each node is assigned a weight representing how it resembles a ball. And each edge is assigned a weight by (3) denoting how likely the two nodes correspond to the same object.

$$v_i^t=\begin{cases}1-\sqrt{c_i^t} & c_i^t\le 1 \\ 0 & c_i^t>1\end{cases}, \text{ with } c_i^t=\frac{1}{M\mu_r^2}\sum_k(\|p_k-\mu\|-\mu_r)^2, \quad (2)$$

$$e_{i,j}^t=(\varpi_s s_{i,j}^t+\varpi_g g_{i,j}^t)/\sqrt{1+(d_{i,j}^t/d_{\max})^2}, \quad (3)$$

where c_i^t is called Circular Variance (CV) [13] (The less CV is, the more the contour resembles a circle.), p_k is a contour point,

M is the number of the contour points, μ is the centroid of the contour, μ_r is the average distance from the contour points to the centroid, and $s_{i,j}^t$ and $g_{i,j}^t$ are the size and the gray-level similarity of the two candidates respectively, with ϖ_s and ϖ_g as the corresponding weights (For simplicity we set $\varpi_s = \varpi_g = 0.5$). We assume that $(\Delta w, \Delta h)$ obeys Gaussian distribution, where Δw and Δh are the width difference and the height difference between MBRs of the two candidates, respectively. Then, $s_{i,j}^t$ can be defined by (4), where Σ can be estimated from ball samples. $g_{i,j}^t$ as defined in (5) is the gray-level normalized cross correlation of the two candidate regions, where vectors $\vec{I}_1$ and $\vec{I}_2$ are obtained through raster scanning of the candidate regions. If the candidate regions are not equal in size, they are adjusted to equal size before scanning.

$$S_{i,j}^t = N(0, \Sigma) \tag{4}$$

$$g_{i,j}^t = \frac{\sum_k \vec{I}_1(k) \cdot \vec{I}_2(k)}{\sqrt{\sum_k \vec{I}_1(k) \cdot \vec{I}_1(k)}\sqrt{\sum_k \vec{I}_2(k) \cdot \vec{I}_2(k)}} \tag{5}$$

```
1. Initialization: $\delta_i^1 = v_i^1$, $\psi_1(i) = 0$, $1 \le i \le N_1$;
2. Recursion: $\delta_j^t = \max_{1 \le i \le N_{t-1}} (\delta_i^{t-1} + e_{i,j}^{t-1} + v_j^t)$,
   $\psi_t(j) = \arg\max_{1 \le i \le N_{t-1}} (\delta_i^{t-1} + e_{i,j}^{t-1} + v_j^t)$, $1 \le j \le N_t$, $2 \le t \le T$;
3. Termination: $q_T = \arg\max_{1 \le i \le N_T} (\delta_i^T)$;
4. Path backtracking: $q_t = \psi_{t+1}(q_{t+1})$, $t = T-1, \cdots, 1$
```

Fig. 6. Ball localization based on Viterbi algorithm

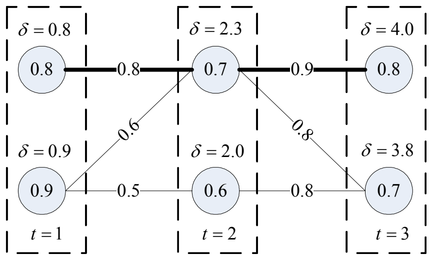

Fig. 7. An illustration of the weighted graph. The optimal path is marked in bold black line.

Finding the optimal path of the graph is a typical dynamic programming problem, where Viterbi algorithm is employed and is depicted in a similar way to [14] as shown in Fig. 6. Note that the graph can be constructed incrementally. Let P_j^t be the optimal path ending at jth candidate in frame t , and then the notations in Fig. 6 are explained as follows. N_t is the number of candidates in frame t , δ_j^t is the sum of node and edge weights along P_j^t , $\Psi_t(j)$ is the index linking to the candidate in frame $t-1$ on P_j^t , and $\{q_t\}_{t=1,\cdots,T}$ is the optimal path. If the length of the optimal path is less than T , then the observation window is moved forward by one frame and the ball detection procedure is run again. An illustration of the graph and its optimal path is shown in Fig. 7.

2) Ball Tracking

In ball tracking procedure, Kalman filter based template matching (in terms of gray-level normalized cross correlation in (5)) is employed. Kalman filter predicts the ball's location in the next frame and filters the tracking result in the current frame. Template matching is used to obtain the observation. Kalman filter and the template are initialized using the ball detection results. A constant velocity dynamics model is adopted for Kalman filter. Due to space limit, we refer the interested readers to [15] for more details about Kalman filter.

A simple but effective method is adopted to make the tracker adaptable to the ball's scale change over frames. A slightly larger block $(x1-\Delta, y1-\Delta, x2+\Delta, y2+\Delta)$ is generated for the matched ball region $(x1, y1, x2, y2)$, where $(x1, y1)$ and $(x2, y2)$ are the top-left coordinates and the bottom-right coordinates of the matching region, respectively. The same method in ball detection procedure is used to extract the object and equation (2) is used to evaluate whether it is the ball. If the ball is detected, the template is updated. Otherwise, the number of consecutive missing detections is counted, and if it is larger than a predefined threshold (say 5); the ball detection procedure is run again.

C. Camera Calibration

Camera calibration is a necessary step in order to extract 3D metric information from 2D images. With the increase of the number of parameters to be estimated, techniques for camera calibration vary from simple to complex forms. In soccer video, some prior knowledge can be used to simplify the process of camera calibration. First, the playfield can be regarded as a plane, which reduces 3×4 projective matrix (camera matrix) to 3×3 one, i.e., homography [16]. Second, the lengths of mark lines within penalty area are known in advance according to the Laws of the Game [6]. This can be used to recover 3D metric information. Third, the position of the main camera is nearly fixed, which makes the 3D position estimation of the camera possible. Here, we are only interested in the homography between the playfield model and its image, which is termed as model-to-image homography, and the 3D position of the main camera. In the following, we introduce the techniques to estimate these parameters.

1) Model-to-Image Homography

Given four correspondence points in general positions between the playfield model and its image, homography $\mathbf{H}_t$ can be estimated directly [16]. Since the length and the width of the playfield are not known, only the intersection points of mark lines within penalty area can be used to estimate the homography as shown in Fig. 8. Let $\mathbf{M}_p = [X, Y]^T$ be a point on the playfield and $\mathbf{m}_t = [u, v]^T$ be its image with $\tilde{\mathbf{M}}_p = [X, Y, 1]^T$ and $\tilde{\mathbf{m}}_t = [u, v, 1]^T$ as their homogenous coordinates, respectively. Then $\tilde{\mathbf{M}}_p$ and $\tilde{\mathbf{m}}_t$ have the relation

$$\tilde{\mathbf{m}}_t = \mathbf{H}_t \tilde{\mathbf{M}}_p. \tag{6}$$

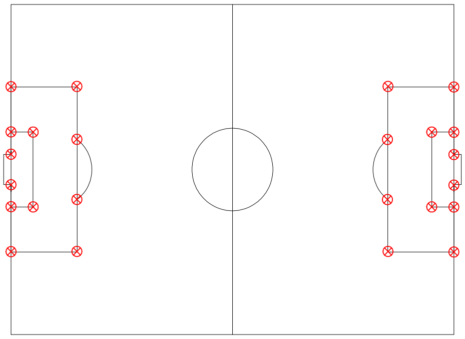

Fig. 8. The playfield model of soccer game. The points marked by red dot can be used to estimate model-to-image homography.

If correspondence points are insufficient, i.e., less than four, model-to-image homography can be estimated indirectly. Since the camera is fixed and only perform rotation and zooming, two images of the playfield can also be related by a homography [16]. To avoid confusion, we call this homography as image-to-image homography, which is estimated by global motion estimation [17]. To enhance the robustness of global motion estimation, player regions are eliminated, and Kanade-Lucas-Tomasi (KLT) good features are used to initialize the translation parameters. More details can be found in our previous work [18]. Let $\mathbf{P}_{t\leftarrow t-1}$ be image-to-image homography between video frames $t-1$ and t. Then $\tilde{\mathbf{m}}_{t-1}$ and $\tilde{\mathbf{m}}_t$ can be related by (7)

$$\tilde{\mathbf{m}}_t = \mathbf{P}_{t\leftarrow t-1}\tilde{\mathbf{m}}_{t-1} \tag{7}$$

From (6) and (7), we can derive (8)

$$\mathbf{H}_t = \mathbf{P}_{t\leftarrow t-1}\mathbf{H}_{t-1}. \tag{8}$$

From (8) we can further derive (9). If $\mathbf{H}_{t-k}$ in the previous frame has been estimated, $\mathbf{H}_t$ can be obtained by (9). Note this technique has extensively been used in image mosaic.

$$\mathbf{H}_t = \mathbf{P}_{t\leftarrow t-1}\mathbf{P}_{t-1\leftarrow t-2}\cdots\mathbf{P}_{t-k+1\leftarrow t-k}\mathbf{H}_{t-k} \tag{9}$$

2) Calibration of the Camera Position

Let $\mathbf{M}=[X,Y,Z]^T$ be a scene point with its homogenous coordinate as $\tilde{\mathbf{M}}=[X,Y,Z,1]^T$, then imaging process can be described as follows

$$s\tilde{\mathbf{m}} = \mathbf{K}[\mathbf{R}\,|\,\mathbf{t}]\tilde{\mathbf{M}},\ \text{ with } \mathbf{K}=\begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix}, \tag{10}$$

where s is an arbitrary non-zero scale factor, $\mathbf{R}$ and $\mathbf{t}$, called the extrinsic parameters, are rotation and translation respectively, which relate the world coordinate system to the camera coordinate system, and $\mathbf{K}$ is called the camera intrinsic parameters matrix, with α and β the scale factors in image u and v axes, γ the skewness of the two image axes, and $[u_0,v_0]$ the coordinates of the principal point. Note the subscript t is omitted for brevity in this subsection.

Denote $\mathbf{H}=[\mathbf{h}_1\ \ \mathbf{h}_2\ \ \mathbf{h}_3]$ and $\mathbf{R}=[\mathbf{r}_1\ \ \mathbf{r}_2\ \ \mathbf{r}_3]$. Since $Z=0$ on the playfield plane, then from (6) and (10) we can derive (11)

$$\mathbf{K}[\mathbf{r}_1\ \ \mathbf{r}_2\ \ \mathbf{t}] = \lambda[\mathbf{h}_1\ \ \mathbf{h}_2\ \ \mathbf{h}_3], \tag{11}$$

where λ is a non-zero scale factor. Once $\mathbf{K}$ is known, the extrinsic parameters are readily computed. From(11), we have

$$\mathbf{r}_1=\lambda\mathbf{K}^{-1}\mathbf{h}_1,\ \mathbf{r}_2=\lambda\mathbf{K}^{-1}\mathbf{h}_2,\ \mathbf{r}_3=\mathbf{r}_1\times\mathbf{r}_2,\ \mathbf{t}=\lambda\mathbf{K}^{-1}\mathbf{h}_3 \tag{12}$$

with $\lambda = 1/\|\mathbf{K}^{-1}\mathbf{h}_1\| = 1/\|\mathbf{K}^{-1}\mathbf{h}_2\|$ [19]. Finally the camera's 3D position $\mathbf{t}_{cw}$ in world coordinate system can be obtained by

$$\mathbf{t}_{cw} = -\mathbf{R}^{-1}\mathbf{t}. \tag{13}$$

Since the main camera can be regarded as a rotating and zooming camera, we adopt [20] to calibrate the intrinsic parameters $\mathbf{K}$. γ is assumed to be zero, and $[u_0,v_0]$ is assumed to be at the image center. This setting hardly affects the result as indicated by [21]. Then only α and β need to be estimated. They can be acquired by solving the equation system (14)

$$\begin{bmatrix}p_{11}p_{21} & p_{12}p_{22}\\ p_{11}p_{31} & p_{12}p_{32}\\ p_{11}p_{31} & p_{22}p_{32}\end{bmatrix}\begin{bmatrix}\alpha^2\\ \beta^2\end{bmatrix}=\begin{bmatrix}-p_{13}p_{23}\\ -p_{13}p_{33}\\ -p_{23}p_{33}\end{bmatrix}, \tag{14}$$

where $p_{ij}, i,j=1,2,3$ is the element of homography $\mathbf{P}$.

D. 3D Position Estimation of Players and the Ball

Once players and the ball are tracked from frame to frame and the camera is calibrated, 3D positions of players and the ball can be estimated. In what follows, we introduce the techniques for 3D position estimation of players and the ball.

1) 3D Position Estimation of Players

3D position estimation of players is straightforward, since players locate on the playfield plane in most cases. When players jump away from the playfield, it is difficult to estimate the 3D position only from monocular video. Knowing the player's image position $\tilde{\mathbf{m}}_t$, the player's position on the playfield plane $\tilde{\mathbf{M}}_p$ can be recovered by (15)

$$\tilde{\mathbf{M}}_p = \mathbf{H}_t^{-1}\tilde{\mathbf{m}}_t \tag{15}$$

2) 3D Position Estimation of the Ball

Estimating 3D position of the ball from monocular video is generally a difficult task. However, if we assume that the ball travels a parabolic trajectory, i.e., the ball flies in a plane which is perpendicular to the playfield plane, 3D position of the ball can be estimated. As shown in Fig. 9, 3D position of the ball is determined by the intersection point of the line l and the plane π . l is determined by the camera position C and the ball's virtual shadow S on the playfield plane (regarding the camera as a light source). π is determined by A and B, which are the intersection points between the ball's trajectory and the playfield plane. A, B, S can be determined by (15), given these points' image positions. Note virtual shadow's image position is actually the ball's image position. When A or B can not be obtained in some conditions, e.g., the ball is kicked by a player before it reaches the playfield, π can also be determined by searching a plane, in which the ball's trajectory is most likely to be a parabola. More details can be found in our previous work [22].

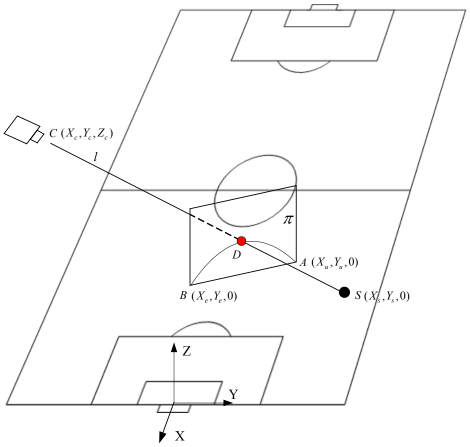

Fig. 9. The geometrical relationship among the camera position, the ball's 3D position, the ball's shadow position, and the ball's flying plane.

IV. 3D Cartoon Animation Generation

Once 3D information is extracted, 3D cartoon animation can be generated by employing computer graphics techniques. In the following, we introduce the key steps in this module.

A. Playfield Modeling

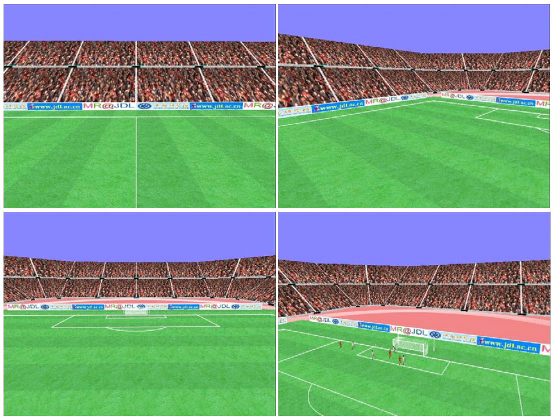

Fig. 10. Several different views of the playfield

Playfield modeling is straightforward, since many parameters of the playfield are known according to the Laws of the Game [6], except the length and the width of the playfield, which can be estimated online by model-to-image homography. To enhance viewing experience, the playfield is not only texture-mapped by image patches which are extracted from the playfield region, but also decorated with billboards, auditoria and so on. Fig. 10 provides several different views of the playfield.

B. Player Modeling

Player model is built according to H-anim1.1 [5], which is the specification for a standard humanoid; hence, it can simulate almost all kinds of actions that a player can perform. Furthermore, we can change the color of the player's jersey, shorts, stockings, and shoes. Several different views of a player are shown in Fig. 11.

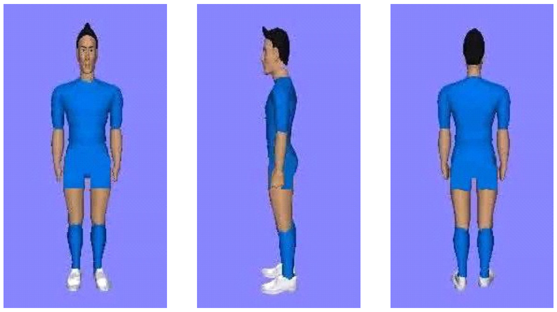

Fig. 11. Several different views of a player.

C. Generating 3D Cartoon Animation

The generation of 3D cartoon animation consists of two main parts. One is the ball's animation. And the other is the player's animation. Before generating cartoon animation, the trajectories of players and the ball are smoothed by Gaussian filtering in order to eliminate noises. Compared with the player's animation, the ball's animation is much easier. The ball is modeled as a white sphere, whose center is placed on the 3D trajectory of the ball. The player's animation is not that easy, since it is a difficult task to recover the player's 3D articulated motion in monocular video. And the fact that the player's region is in low resolution in far-view video also adds much challenge. In our current implementation, we try to recover the player's motion type including run, walk and stop from analyzing the player's trajectory. Firstly, using motion capture device, we built the player's motion database which includes run, walk, etc. Then, the player's trajectory is divided into several non-overlapping segments according to the same temporal interval (e.g. 1s). For each segment, the average magnitude of velocity is computed. If it is greater than a predefined threshold, the motion type of the segment is classified as run; or else if it is less than another predefined threshold, the motion type of the segment is classified as stop; otherwise, the segment is classified as walk segment. Finally, for each segment the corresponding motion data is retrieved from the database to drive the player model, and the player's motion direction is determined by the direction of velocity.

V. Experimental Results

We developed the Video2Cartoon system based on Visual C++ 6.0 and OpenGL. The system allows users to control a virtual camera to perform pan, tilt, and zoom, and to change the viewpoint. Moreover, by using the direction keys and the mouse, users can roam around the playfield. In this section, we perform experiments on several clips of broadcast soccer video to evaluate the performance of the system. The video data was recorded from TV broadcast of Euro Cup 2004, and was compressed by MPEG-II with the frame rate of 25 fps and the resolution of 352x288.

A. Players Detection and Tracking

Experiments are performed on five far-view video clips. Table I shows the results of the five clips with 1405 frames in total. Due to good performance of SVM detector and particle filter tracker, the precision rate is very high; therefore, only recall rate is given. From Table I we can see that the overall recall rate is 90.4%. In some situations players can not be detected and tracked. This arises from two main reasons. One is that the player region is so small that it is treated as noise. The other is that the player region is so close to the caption, logo, and mark line that it is filtered out in the post-processing step after playfield detection.

TABLE I
PLAYERS DETECTION AND TRACKING RESULTS

video clip	#frame	#players	#detected &tracked	recall (%)
pclip1	399	1232	1146	93.0
pclip2	271	795	713	89.7
pclip3	315	1241	1053	84.9
pclip4	230	763	694	91.0
pclip5	190	644	619	96.1
total	1405	4675	4225	90.4

B. Ball Detection and Tracking

To evaluate the performance of ball detection and tracking, we perform experiments on two representative video clips with 1369 frames in total. Some statistics of the results are shown in Table II. The playfield appearance of the first clip is really poor. The playfield color is far away from green. Part of the playfield is under sunshine, while the rest of the playfield is under the shadow of the stadium. The playfield appearance of the second clip is very good. The playfield color is green. And there is no shadow on the playfield. We manually marked the ground truth. We say that a frame contains a ball, if we can find it not depending on the previous and the following frames.

TABLE II
BALL DETECTION AND TRACKING RESULTS

video clip	# frame	# ball	#detected &tracked	#false positive	precision (%)	recall (%)
bclip1	650	600	460	53	89.7	76.7
bclip2	719	631	533	65	89.1	84.5
total	1369	1231	993	118	89.4	80.7

From Table II we can see that the precision rates of the two clips are very close, while the recall rate of the first clip is lower than that of the second one. This is mainly because poor lighting condition makes detection of the ball difficult. Another reason of low recall rate is that the ball is occluded by players, or merged with mark lines. The false positives are mainly due to the fact that the ball is totally occluded by players or merged with mark lines, while at the same time the socks of players or some line segments are most likely to be the ball.

C. 3D Position Estimation

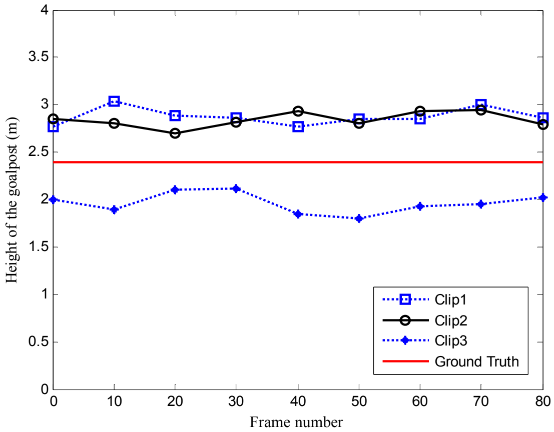

Fig. 12. The estimated height of the goalpost on three video clips. The red straight line indicates the true height of the goalpost which is 2.44m.

In this subsection we evaluate the performance of 3D position estimation. However, the ground truth of the ball is very hard to be obtained. Fortunately, the height of the goalpost is known according to the Laws of the Game [6]. And we use it to evaluate our approach. The plane is determined by the two goalposts. The line is determined by the camera's position and the virtual shadow of the top of the goalpost. The estimated height of the goalpost on three video clips is shown in Fig. 12, where the red straight line indicates the ground truth (2.44m). Only the first frame's model-to-image homography is computed from the intersection points of mark lines. The subsequent frames' homography is computed by global motion estimation. The deviation from the ground truth is mainly due to the error accumulation of global motion estimation, the digitization error of video frame, the inaccuracy of pin-hole camera model, etc. However, for generating cartoon animation, the proposed approach for 3D position estimation is acceptable.

D. Generating 3D Cartoon Animation

Since goal events are most attractive to viewers in soccer games, we take them as case study and perform experiments on five video clips of goal events. Due to space limit, we only present the result from a video clip of a goal, which is recorded from TV broadcast of a match between Portugal and Russia in Euro Cup 2004. Sample frames are shown in Fig. 13. Fig. 13(a) is the original soccer video. Fig. 13(b)-(d) present the generated cartoon animation from different viewpoints. The video data can be accessed on our lab's website http://www.jdl.ac.cn/en/project/mrhomepage/En_index.htm.

VI. SUMMARY AND FUTURE WORK

A system for converting broadcast soccer video into 3D cartoon animation is proposed. By taking advantage of computer vision and computer graphics techniques, the system allows users to change the viewpoint, and also allows users to roam around the playfield using the direction keys and the mouse. Our system has some potential applications. Content service providers can use the system to generate cartoon

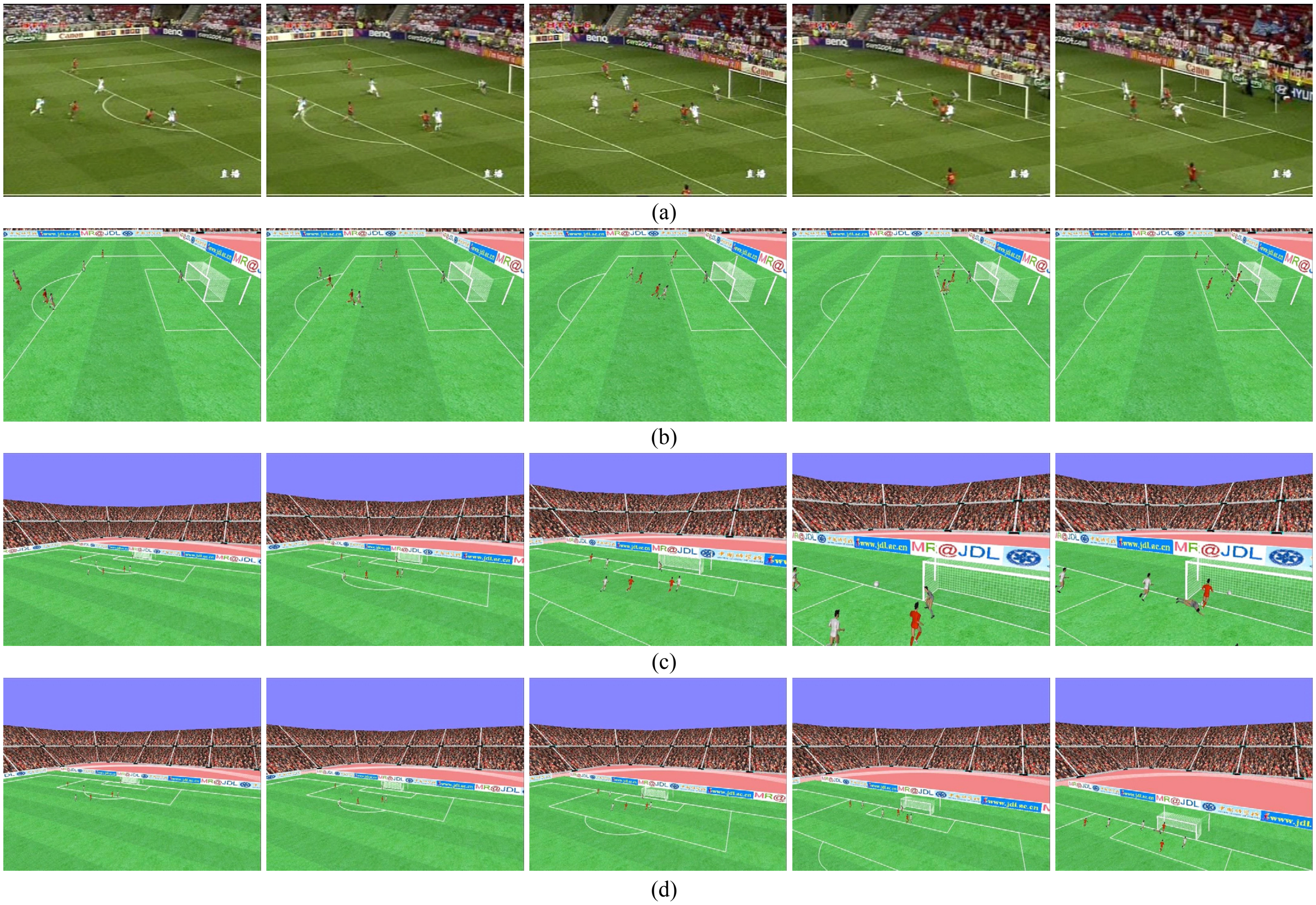

Fig. 13. Cartoon animation of a video clip of a goal (sample frames 1, 25, 50, 75, and 100 are shown). (a) the original soccer video. (b) a view from the right side of the playfield. (c) a zoom-in view. (d) a flying-through view.

animation of a goal from arbitrary viewpoint, and distribute it on their web portals, or to mobile device users (e.g. cell phone users). On the other hand, customers themselves can use the system to generate their personalized cartoon animation, by switching the viewpoint, changing the billboards from one to another, etc.

Currently the system is not fully automatic. This is mainly due to the following factors. First, automatically determining the ball's starting and ending points on the playfield plane is not easy. Second, it is not trivial to robustly determine the correspondence points between the playfield model and its image. Third, long-time occlusion among players makes the tracker fail to work. Currently, we deal with these problems manually, and leave the solutions to them for future work.

Key frame based tracking algorithms [23], [24] provide a possible solution to player occlusion problem. By identifying the player's region in key frames before and after occlusion, optimization algorithm can be employed to extract the player's trajectory during occlusion. In future work, recovering the player's 3D articulated motion from low resolution image is another problem worthy of research. This will make the generated cartoon animation more realistic and vivid. Efros et al.'s work [25] lies in this direction.

Acknowledgment

The authors would like to thank Mr. Changshui Yang for his help in player modeling, Mr. Zhebin Zhang for his work in playfield modeling, and Dr. Shuqiang Jiang for his helpful suggestions.

References

[1] K. Matsui, M. Iwase, M. Agata, T. Tanaka, and N. Ohnishi, "Soccer image sequence computed by a virtual camera," In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 860-865, 1998.

[2] T. Bebie and H. Bieri, "SoccerMan-reconstructing soccer games from video sequences," In *Proceedings of IEEE International Conference on Image Processing*, vol. 1, pp. 898-902, 1998.

[3] X. Yu, X. Yan, T. S. Hay, and H. W. Leong, "3D reconstruction and enrichment of broadcast soccer video," In *Proceedings of the 12th Annual ACM international Conference on Multimedia*, pp. 260-263, 2004.

[4] T. Kanade and P. J. Narayanan, "Virtualized reality: perspectives on 4D digitization of dynamic events," *IEEE Computer Graphics and Applications*, vol. 27, no. 3, pp. 32-40, 2007.

[5] Humanoid Animation Working Group, "Specification for a Standard Humanoid Version 1.1." http://h-anim.org/Specifications/H-Anim1.1/.

[6] FIFA, "Laws of the Game." http://www.fifa.com/mm/document/affederation/federation/lotg2006_e_1581.pdf.

[7] S. Jiang, Q. Ye, W. Gao, and T. Huang, "A new method to segment playfield and its applications in match analysis in sports video," In *Proceedings of the 12th Annual ACM international Conference on Multimedia*, pp. 292-295, 2004.

[8] Q. Ye, W. Gao, and W. Zeng, "Color image segmentation using density-based clustering," In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing,* pp. 345-348, 2003.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[10] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," In *Proceedings of European Conference on Computer Vision*, pp. 661-675, 2002.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb 2002.

[12] G. Zhu, D. Liang, Y. Liu, Q. Huang, and W. Gao, "Improving particle filter with support vector regression for efficient visual tracking," In *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 422-425, 2005.

[13] M. Peura and J. Iivarinen, "Efficiency of simple shape descriptors," In C. Arcelli et al. (Eds.) Advances in Visual Form Analysis, World Scientific, Singapore, pp. 443-451, 1997.

[14] L. R. Rabiner, "A tutorial on hidden Markov model and selected applications in speech recognition," In *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[15] G. Welch and G. Bishop, "An introduction to the Kalman filter," Technical Report TR95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 1995. http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html.

[16] R. Hartly and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, 2003.

[17] F. Dufaux and J. Konrad, "Efficient, robust, and fast global motion estimation for video coding," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 497-501, Mar. 2000.

[18] Y. Liu, Q. Huang, Q. Ye, and W. Gao, "A new method to calculate the camera focusing area and player position on playfield in soccer video," In *Proceedings of SPIE Visual Communications and Image Processing*, Beijing, China, pp. 1524-1533, Jul.12-15, 2005.

[19] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov 2000.

[20] Y. Seo and K.S. Hong, "Auto-calibration of a rotating and zooming camera," In *Proceedings of the IAPR Workshop on Machine Vision Applications*, pp. 274-277, 1998.

[21] L. Agapito, E. Hayman, and I. Reid, "Self-calibration of rotating and zooming cameras," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 107-127, 2001.

[22] Y. Liu, D. Liang, Q. Huang, and W. Gao, "Extracting 3D information from broadcast soccer video," *Image and Vision Computing*, vol. 24, no. 10, pp. 1146-1162, Oct.2006.

[23] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz, "Keyframe-based tracking for rotoscoping and animation," In *Proceedings of ACM SIGGRAPH,* pp. 584-591, 2004.

[24] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum, "Bidirectional tracking using trajectory segment analysis," In *Proceedings of Tenth IEEE International Conference on Computer Vision*, vol. 1, pp. 717-724, 2005.

[25] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," In *Proceedings of Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 726-733, 13-16 Oct. 2003.

Dawei Liang received the B.S. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2003. Currently, he is pursuing the Ph.D. degree in computer science at the same institution.

Since June 2004, he has been a research assistant in Joint R&D Laboratory for Advanced Computer and Communication Technologies, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, pattern recognition, and video analysis.

Qingming Huang (M'04) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China in 1994.

Dr. Huang was a Postdoctoral Fellow in National University of Singapore from 1995 to 1996, and worked in Institute for Infocomm Research, Singapore as Member Research Staff from 1996 to 2002. Currently, he is a professor in Graduate School of Chinese Academy of Sciences. He has published over 100 scientific papers. His current research areas are image processing, video analysis, video coding, and pattern recognition.

Yang Liu received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2006, respectively.

From 2001-2005, he was a Research Assistant with the Joint R&D Lab, Chinese Academy of Sciences, Beijing, China. Since 2007, he has been a lecturer with the department of computer science, Harbin Institute of technology, Harbin. He has published more than 20 scientific papers. His research interests include computer vision, pattern recognition and machine learning.

Guangyu Zhu received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, where he is currently pursuing the Ph.D. degree.

His research interests include image/video processing, multimedia content analysis, computer vision and pattern recognition, and machine learning.

Wen Gao (M'92–SM'05) received the M.S. degree and the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1985 and in 1988, respectively, and the Ph.D. degree in electronics engineering from University of Tokyo, Japan, in 1991.

Prof. Gao was a research fellow at Institute of Medical Electronics Engineering, the University of Tokyo, in 1992, and a visiting professor at Robotics Institute, Carnegie Mellon University, in 1993. From 1994 to 1995, he was a visiting professor at the AI Lab of MIT. Currently, he is a professor in School of Electronic Engineering and Computer Science, Peking University, China, and a professor in computer science at Harbin Institute of Technology. He is also the honor professor in computer science at City University of Hong Kong, and the External Fellow of International Computer Science Institute, UC Berkeley. He has published 4 books and over 400 scientific papers. His research interests are in the areas of signal processing, image and video communication, computer vision and artificial intelligence.

Dr. Gao serves as Associate Editor of *IEEE Transactions on Circuits and Systems for Video Technology*, Associate Editor of *IEEE Transactions on Multimedia*, Editor-in-Chief of the *Journal of Computer* (in Chinese), and Editor of the *Journal of Visual Communication and Image Representation*.