Robust Camera Calibration and Player Tracking in Broadcast Basketball Video

Min-Chun Hu, *Student Member, IEEE*, Ming-Hsiu Chang, Ja-Ling Wu, *Fellow, IEEE*, and Lin Chi

***Abstract*—With the growth of fandom population, a considerable amount of broadcast sports videos have been recorded, and a lot of research has focused on automatically detecting semantic events in the recorded video to develop an efficient video browsing tool for a general viewer. However, a professional sportsman or coach wonders about high level semantics in a different perspective, such as the offensive or defensive strategy performed by the players. Analyzing tactics is much more challenging in a broadcast basketball video than in other kinds of sports videos due to its complicated scenes and varied camera movements. In this paper, by developing a quadrangle candidate generation algorithm and refining the model fitting score, we ameliorate the court-based camera calibration technique to be applicable to broadcast basketball videos. Player trajectories are extracted from the video by a CamShift-based tracking method and mapped to the real-world court coordinates according to the calibrated results. The player position/trajectory information in the court coordinates can be further analyzed for professional-oriented applications such as detecting wide open event, retrieving target video clips based on trajectories, and inferring implicit/explicit tactics. Experimental results show the robustness of the proposed calibration and tracking algorithms, and three practicable applications are introduced to address the applicability of our system.**

***Index Terms*—Broadcast basketball video, camera calibration, CamShift algorithm, highlight extraction, player tracking.**

I. Introduction

A. Motivation

COMMERCIAL applications of video analysis are getting valuable with the development of digital television. People can easily record all kinds of programs and enjoy the videos in their leisure time. Among these programs, broadcast sports videos are usually more tedious than others since they involve not only the main games, but also breaks or commercials. Even main games comprise periods which are not splendid enough for the audience. Therefore, a considerable amount of research focuses on automatically annotating semantic concepts in sports videos, and providing a spellbinding way to browse videos. From a sports-professional's point of view, they have different requirements from general viewers while watching sports videos. They usually thirst for possible tactics taken by the opponents since they hope to find the competitor's weaknesses and practice corresponding strategies before the game. Conventionally, videos of several matches are collected and reviewed to conclude tactical information, which is obviously a time-consuming and exhausting work for a professional. Flourishing researches on automatically analyzing tactics in sports video have been proposed, but there is still no good solution for broadcast basketball video since trajectory information is difficult to be extracted from videos captured in complex scenes with plenty camera motions. Other proposed applications like event detection and highlight extraction for basketball videos are very limited to be used due to the following two reasons: 1) A basketball match involves plenty event types and viewers have different preferences while watching the video. However, the current existing techniques can only extract a few appealing basketball events for the viewer. 2) Some works detect more event types combining web-casting text analysis, but only games of the NBA have enough web-casting information. These facts motivate us to try to develop a more applicable professional-oriented system for analyzing broadcast basketball videos.

B. Related Work

There has been a proliferation of research on sports video analysis in the past ten years, and most of them focused on highlight extraction, structure analysis, and semantic event annotation. For example, Gong *et al.* [1] utilized object color and texture features to generate highlights in broadcast soccer videos. Xu *et al.* [2] and Xie *et al.* [3] detected plays/breaks in soccer games by using frame view types and motion/color features, respectively. Li *et al.* [4] summarized football video by play/break and slow-motion replay detection using both cinematic and object descriptors. Rui *et al.* [5] detected highlights using audio features alone without relying on expensively computing video features. Besides visual/audio features extracted from the video, Babaguchi *et al.* [6] combined text information from closed captions (CC) to seek for time spans in which events are likely to take place. With the aid of web-casting text information, Xu *et al.* [7] tried to annotate sports videos with semantic labels which not only cover general events, e.g., scoring/fouls, but also the semantics of events, e.g., names of players. Moreover, some works analyzed the superimposed caption to more accurately annotate the videos [8], [9].

Manuscript received March 29, 2010; revised August 28, 2010 and October 19, 2010; accepted November 28, 2010. Date of publication December 17, 2010; date of current version March 18, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ajay Divakaran.

M.-C. Hu is with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 106, Taiwan (e-mail: trimy@cmlab.csie.ntu.edu.tw).

M.-H. Chang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: cmhsiu@cmlab.csie.ntu.edu.tw).

J.-L. Wu is with the Graduate Institute of Networking and Multimedia, and also with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: wjl@cmlab.csie.ntu.edu.tw).

L. Chi is with the Physical Education Center, Ta-Hwa Institute of Technology, Hsinchu County 307, Taiwan (e-mail: chilin1215@hotmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2010.2100373

Object trajectories also provide rich information for semantic understanding, especially for tactic inferring. Assfalg *et al.* [10] employed camera motion and locations of the players to detect events in soccer videos. Tovinkere *et al.* [11] utilized object trajectories to achieve semantic event detection in soccer video with a set of heuristic rules which are derived from a hierarchical entity-relationship model. Intille *et al.* [12] analyzed interactions in football videos based on object trajectories, which could be clues for play classification. Not surprisingly, the works presented in [10]–[12] assumed that trajectory information is obtained in advance. To obtain the object trajectories automatically, Pingali *et al.* [13] proposed a real-time tracking system for tennis videos captured by a stationary camera. In [13], player trajectories are obtained by dynamically clustering tracks of local features, and ball segmentation/tracking is realized based on shape and color features of the ball. Guéziec[14] exploited the kinematic properties of the baseball's flight to track the ball during pitches in real-time. However, extensive prior knowledge such as camera locations and coverage have to be known for tracking the ball. Zhu *et al.* [15] used object trajectories and web-casting text to extract tactical information from the goal events in broadcast soccer videos. Unfortunately, web-casting text is not always available for each game, and only relative coordinates rather than absolute court coordinates of the object trajectory were extracted.

In contrast to other popular sports videos (e.g., soccer, tennis, or baseball videos), only a few works have been done for basketball video analysis. Zhou *et al.* [16] exploited motion, color, and edge features to construct a supervised rule-based classifier, which can detect left/right offense, left/right fast break, left/right scores, left/right dunks, and close-up scenes in basketball videos. Saur *et al.* [17] developed a basketball annotation system which combines the low-level information extracted from MPEG stream with the prior knowledge of basketball video structure to annotate events such as wide-angle/close-up views, fast breaks, steals, potential shots, number of possessions, and possession times. Liu *et al.* [18] presented a multiple-modality method to extract semantic information from basketball videos. The visual, motion, and audio information are extracted from a video clip to first generate some low-level video segmentation and classification. Domain knowledge is then exploited for detecting events like "foul" and "shot at the basket" in basketball videos. Also using a multiple-modality framework, Zhang *et al.* [19] aligned the web-casting text and the video to extract more event types. These works mainly focused on scene classification and only detected a small part of basketball events which are less informative for professional viewers.

To extract more high-level semantics in basketball videos, Chen *et al.* [20] designed a physics-based algorithm to reconstruct 3-D ball trajectories, and then obtain shooting location statistics to helped the defense team infer which area has to be guarded with more attention. However, [20] can only extract ball trajectories for long distance shooting due to the limitation of the physics assumption, while many successful offensive tactics result in short distance shooting. Moreover, depending only on ball trajectory information is not sufficient for gathering tactics. Another problem of [20] is that the authors directly applied a court-based camera calibration method proposed for tennis video [21], [22] to calibrate the image and the real-world court coordinates. Unfortunately, the calibration results are not robust since court view frames in basketball videos are more complicated and have larger location variation than court view frames in tennis videos.

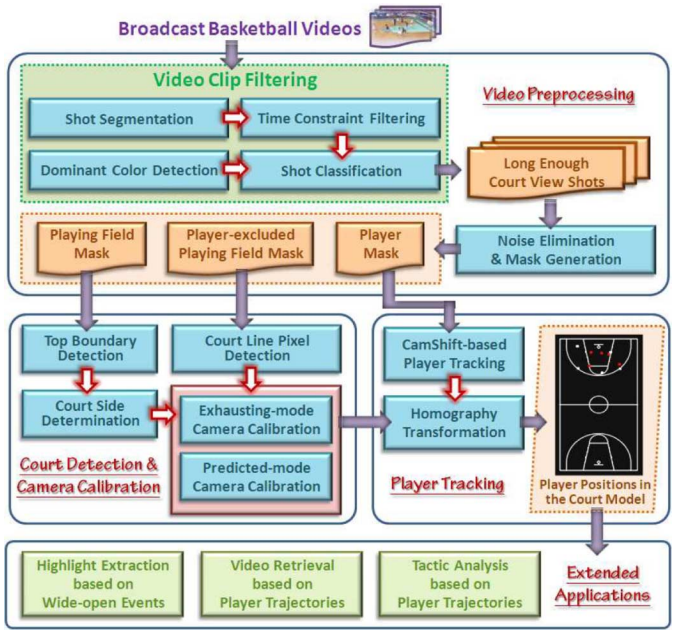


Fig. 1. Flowchart of the proposed video analysis system for broadcast basketball game.

C. Contribution

With the foregoing motivation and limitations of the existing work, we develop a robust camera calibration and player tracking system for broadcast basketball videos. As shown in Fig. 1, the system contains three main components including video-preprocessing module, court detection/camera calibration module, and player tracking module. Video-preprocessing module extracts video clips having higher probability of being a play segment and generates three useful masks for the following analysis. For all the play clip candidates, a court detection/camera calibration technique is applied to obtain the court location and the relationship between the image court and the real-world court model. Clips without court information are filtered out from play clip candidates, and players are then detected/tracked by the player tracking module for the remaining clips. Finally, we map player positions to the court-model coordinates, and the extracted trajectory information can be applied to three professional-oriented applications. Compared with existing works on sports video analysis, the main contributions of our work are summarized as follows.

- The proposed preprocessing steps including video clip filtering, noise elimination, and mask generation can effectively speed up the processing time and increase the accuracy of the following analyses.
- The court-based calibration method presented in [21] and [22] is modified to be more robust for analyzing broadcast basketball videos on the basis of the proposed quadrangle candidate generation algorithm and the model fitting score.
- A multi-players tracking algorithm is designed based on the continuously adaptive mean shift (CamShift) technique [23] for efficiently extracting player trajectories in the image coordinates.

(a) (b) (c)

Fig. 2. Examples of different view shots. (a) Close-up view frame. (b) Median view frame. (c) Court view frame.

The rest of this paper is organized as follows. Section II presents the video-preprocessing steps, in which we extract long enough court-view shots for further analysis. Section III describes the details of court detection and camera calibration for the analysis of broadcast basketball videos. Players can be detected and tracked using color information as expounded in Section IV. Experimental results are reported and discussed in Section V, and we show three applications of the proposed calibration/tracking algorithm in Section VI. Finally, we conclude the paper in Section VII.

II. Video Preprocessing

Broadcast video of a complete basketball game may last longer than one hour, while some parts of the video are less informative to professional viewers. In the video-preprocessing steps, irrelevant video clips are filtered out from the input video sequence according to time-length and dominant color ratio, and three useful masks are extracted from the rest of the clips.

A. Video Clip Filtering

The input video is first segmented into shots by a typical shot change detection method based on histogram difference [24]. Video shots shorter than Th_{time} seconds will be excluded from the following analysis since a meaningful basketball event/tactic usually lasts for some time. Two hundred plays containing successful offense are manually selected from ten basketball videos by a professional basketball coach. Among all selected plays, the mean (μ_t) and standard deviation (σ_t) of the video time length are utilized to determine the threshold Th_{time}, i.e., $Th_{time} = \mu_t - 2\sigma_t$.

Video clips with little court information such as a close-up view shot and a median view shot [as shown in Fig. 2(a) and (b), respectively] will increase the difficulty in tactic inferring. In contrast, a court view shot [as shown in Fig. 2(c)] is much richer in spatial relationship between players and the court, and therefore, we only keep court view shots for conducting further analysis. Court view shots are extracted according to the ratio of dominant (playing field) color pixels. Frames in a court view shot contain a large ratio of pixels possessing the dominant (playing field) color. However, the playing field color varies in different basketball games or under different lighting conditions. Thus, we adaptively determine the color characteristics of the playing field for each game on the basis of Gaussian mixture model (GMM) [25]. Given a basketball video of a game, we periodically adjust the dominant color ranges every T (say 5) minutes according to the histograms collected from the latest K (say 500) frames.

For each video shot longer than Th_{time}, we sample frames in it and classify each sampled frame as a court view frame or not on the basis of the dominant color ratio. A majority mechanism is then applied to determine whether a shot is a court view shot, and all long enough court view shots are retained in the filtering step.

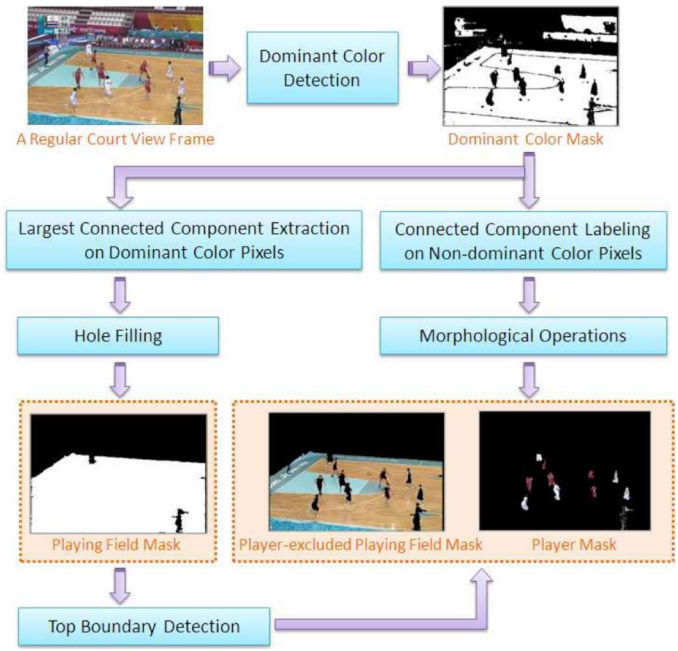


Fig. 3. Block diagram of noise elimination and mask generation.

B. Noise Elimination and Mask Generation

In a regular broadcast basketball video, a court view frame is composed of the area of the audience or stadium, and the playing field. We can further divide the playing field into the player regions and the playing field excluding players. Noises from the areas of the audience or players' uniforms will reduce the stability of the court detection/camera calibration module. Similarly, objects from the audience region will raise false alarms in the player tracking process. For the sake of improving accuracy and diminishing computational cost of the following processes, we generate three masks (i.e., playing field mask, player mask, and player-excluded playing field mask), and take different masks as input for the following modules. Fig. 3 illustrates how to eliminate noises and generate masks for each court view frame.

We first create a binary dominant mask by the dominant color detection technique mentioned in Section II-A. Based on the dominant color mask, we apply connected component labeling to both the dominant color pixels and non-dominant color pixels. The playing field mask is obtained by extracting the largest connected component of dominant color pixels and filling holes in it. For all connected components of non-dominant color pixels, morphological operations including region-growing and small object elimination are exploited to obtain better results of the player mask. However, players nearby the court boundary may not be correctly extracted when parts of their bodies are out of the playing field. To solve this problem, we utilize the top boundaries of the playing field (cf. Section III-A) to separate players from the audience/stadium area. Fig. 4 shows an example of the improved player extraction

(a) (b) (c)

Fig. 4. Example result of player extraction. (a) Original frame. (b) Player extraction result without using the top boundary information. (c) Player extraction result with the aid of top boundary detection.

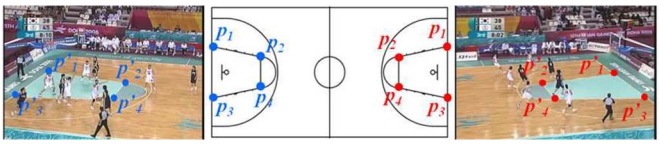


Fig. 5. Corresponding four pairs of intersections for the left-side court or the right-side court are taken as hints for locating the court in the image and calculating the homography matrix $\mathbf{H}$.

method. As shown in Fig. 4(b) and (c), the naive method fails to segment players near the top court line, while all player blobs inside the playing field can be successfully extracted with the aid of the top boundary information. The player-excluded playing field mask can be easily obtained by filtering out player pixels from the playing field mask.

III. Court Detection and Camera Calibration

The camera calibration module provides a geometric transformation which maps a point $\mathbf{p}' = (x', y', 1)^T$ in the image coordinates to a point $\mathbf{p} = (x, y, z, 1)^T$ in the real-world coordinates. According to the pinhole camera model, the paired points $(\mathbf{p}', \mathbf{p})$ have the following relationship:

$$\begin{aligned}\mathbf{p}' &= \mathbf{K}[\mathbf{R}|\mathbf{T}]\mathbf{p} \\ &= \begin{bmatrix} f_x & c & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_x \\ r_{10} & r_{11} & r_{12} & t_y \\ r_{20} & r_{21} & r_{22} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \end{aligned} \quad (1)$$

where $\mathbf{K}$ is a 3×3 matrix containing five camera intrinsic parameters. $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrix and the translation vector, respectively. Assuming the court model is on the plane of $z = 0$, (1) can be rewritten as

$$\mathbf{p}' = \begin{bmatrix} f_x & c & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} & t_x \\ r_{10} & r_{11} & t_y \\ r_{20} & r_{21} & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{H}\mathbf{p} \quad (2)$$

where $\mathbf{H}$ is a 3×3 homography transformation matrix containing eight independent parameters [22], and a point on the plane $z = 0$ can be denoted as $\mathbf{p} = (x, y, 1)^T$ for convenience. To estimate the homography matrix $\mathbf{H}$, we need four pairs of corresponding points to calculate the eight unknown parameters. Since a left/right-side court view frame usually contains the whole free-throw lane, we utilize four corners of the free-throw lane (i.e., four pairs of blue/red intersections for the left/right-side court, respectively, as shown in Fig. 5) to locate the court position in the image and to calculate the homography matrix $\mathbf{H}$.

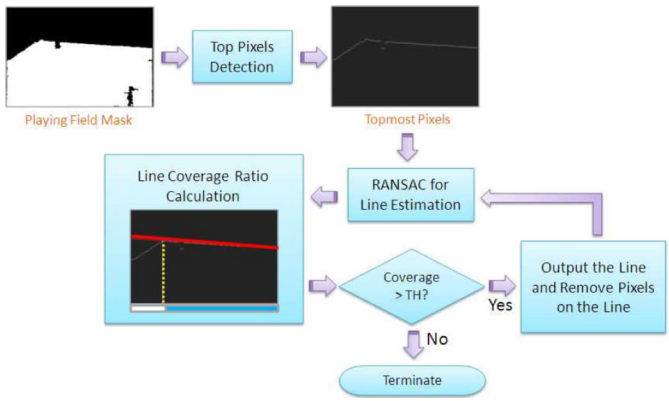


Fig. 6. Block diagram of the top boundary detection algorithm.

A. Top Boundary Detection

Top boundary information contributes to our system in three perspectives: 1) improve the result of player extraction when players' bodies are out of the playing field (cf. Section II-B), 2) determine the court side for current frame by the slopes of top boundaries (details will be described in Section III-B), and 3) use slopes of boundaries to restrict the sampling area in the RANSAC-based court-line estimation process (which will be described in Section III-DI).

Fig. 6 depicts the top boundary detection algorithm. We take the obtained playing field mask as input and keep the topmost pixels for each column of the frame. A RANSAC-based algorithm [26] is then used to estimate line parameters of the top boundary that best matches all the topmost pixels of the playing field. The RANSAC-based line estimation can achieve a reliable result within few iterations since the topmost pixels are clear and with little noise. For a left-side or right-side court view frame, there exist two top boundaries of the playing field. However, just one top boundary appears when the camera is panned to the middle-court. To address this problem, we project each pixel of the detected top boundary onto the horizontal axis (as illustrated by the figure inside the line coverage ratio calculation block). If the line coverage ratio (i.e., the blue partition of the bottom horizontal-line) is above a threshold Th_{cover}, we output the current line, remove topmost pixels on the line, and estimate another line for the rest of the topmost pixels; otherwise, the top boundary detection process terminates. We set the second top boundary the same as the first one if only one top line of the playing field can be found.

B. Court Side Determination

Court side information is important since we have to decide which four intersections are legitimate for calculating the homography matrix $\mathbf{H}$. The results of top boundary detection can be utilized to determine the court side of a court view frame. Let $S_1(i)$ and $S_2(i)$ be the slopes of the first and the second detected top boundaries in frame i, respectively. Fig. 7 illustrates the trends of slopes along the time, where the red circles and the green plus signs denote $S_1(i)$ and $S_2(i)$, respectively. We observe that when the camera pans from the right-side court to the middle court and then to the left-side court, $S_2(i)$ values vary from large negative values to nearly zero values and then to large positive values. Hence, $S_2(i)$ can be utilized to classify

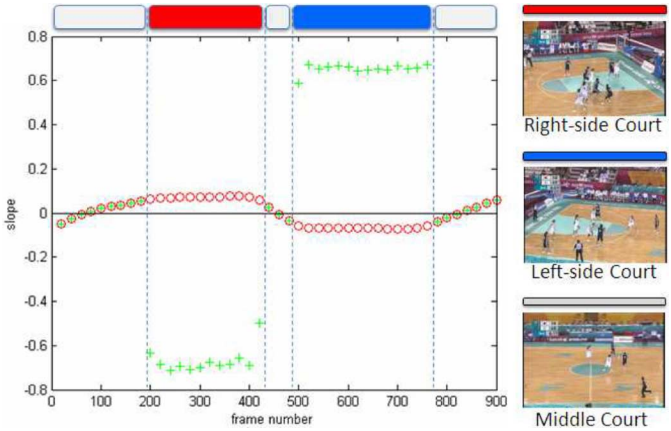


Fig. 7. Slope trends of the first and the second top boundaries along the time in a video clip panning from the right-side court to the left-side court. Slope of the first detected boundary (represented by a red circle) keeps a small value around zero, while the slope of the second detected boundary (represented by a green plus sign) varies when camera pans to different scenes.

a court view frame into a right-side court frame, a left-side court frame, or a middle court frame. Moreover, we can detect court side change to finely segment a court view video clip into several plays since a court view video clip may contain more than one offensive play if the video producer does not change shot in that period.

C. Court Line Pixel Detection

Deriving the four intersections for calculating homography matrix $\mathbf{H}$ from the image frame is challenging since the court involves many noises; for example, the intersections may be occluded by players. Additionally, it is hard to recognize each intersection individually because too many points on the court have similar color or texture characteristics. Instead of finding intersections directly, Farin *et al.* [21], [22] searched for court lines crossing at these points. However, Farin's work is mainly designed for tennis and soccer videos, which are much easier cases due to their clean courts and small ratio of player regions. In this work, we modify Farin's method and make it practical to be applied to broadcast basketball videos.

Following the predefined four intersections for a basketball video, we have to recognize court lines of the free-throw lane. The first step of searching for these lines is to extract possible court line pixels in the frame. Farin's method first identified white court-line pixels in the whole frame, and exploited an additional filter to remove false detections in the texture areas. According to the two eigenvalues of a structure tensor J, the pixels with complex or flat textures such as the noises from the audience or player clothing can be removed to some extent. The texture-based filtering process on the whole frame obviously took lots of computation time and still retained many noisy pixels.

In contrast, we directly filter out the audience/stadium and player regions by dominant color detection (a commonly necessary step in sports video analysis) and simple operations (cf. Section II-B). The obtained player-excluded playing field mask is then utilized as the input for court line pixel detection. The luminance and color thresholds [21] are applied to the input mask to extract white court-line pixels as shown in Fig. 8(b). Furthermore, we take advantage of some pre-knowledge about the basketball court to eliminate white pixels out of the free-throw lane

(a) (b) (c)

Fig. 8. Result of court line pixel detection. (a) Player-excluded playing field mask. (b) White court-line pixels in the input mask. (c) Court line pixels after eliminating pixels with court pre-knowledge.

[cf. Fig. 8(c)]. That is, the free-throw lane is located near the middle part (in terms of the vertical axis) of the playing field for a regular court view frame.

D. Exhausting-Mode Camera Calibration

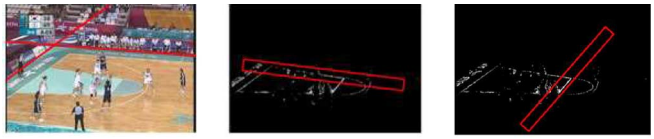
A given court view video shot may contain frames of the left/right-side court and the mid-court, and we only calibrate left/right-side court frames in this work. The homography matrix $\mathbf{H}$ of the first detected left/right-side court frame is estimated in the exhausting-mode and a frame is well calibrated when $\mathbf{H}$ is acquired with a high enough court fitting score. Once several successive frames are well calibrated, the system switches to the predicted-mode for the following frames to speed up the calibration procedure. When the fitting score gets lower than a threshold, the system switches back to the exhausting-mode. The exhausting-mode camera calibration is achieved as follows.

1) Dominant Court Line Detection: Taking the middle-part court line pixels as input, we apply a RANSAC-based line detector to derive the parameters of all dominant lines. Farin *et al.* [22] applied a RANSAC-based line detector which hypothesizes a line using two randomly selected points. They computed the support of a line hypothesis g as

$$s(g) = \sum_{(x',y')\in P} \max(\tau - d(g, x', y'), 0) \qquad (3)$$

where P is the set of court-line pixels and $d(g, x', y')$ denotes the distance between (x', y') and the line hypothesis g. After several iterations, the hypothesis with the highest score is chosen and all court line pixels along the line are removed. The algorithm described above is repeated to estimate the remaining court lines. However, the broadcast basketball video contains large amount of noisy white pixels, and we must increase the number of RANSAC iterations for each line to get an acceptable result. To substantially reduce the number of iterations but still keep good line estimation result, we hypothesize each line by sampling the two points in the direction approximating to the orientation of the top boundaries (as shown in Fig. 9) since all dominant court lines of the free-throw lane are near parallel to top boundaries of the playing field.

2) Quadrangle Candidate Generation: After extracting dominant court lines from the image, we calculate intersections in the frame and generate quadrangle candidates for the free-throw lane. Court model fitting is then applied to find a best fitting homography matrix $\mathbf{H}$ among all quadrangle candidates. However, the number of quadrangle candidates increases rapidly with the amount of intersections. To diminish the processing time spent on court model fitting, we generate reasonable quadrangle candidates by Algorithm 1, which is based on the following two observations.



(a) (b) (c)

Fig. 9. RANSAC sampling in the exhausting-mode. (a) Top boundaries of the playing field. (b) and (c) indicate that the two points of a line hypothesis can be sampled in the directions near parallel to the first and the second top boundaries, respectively.

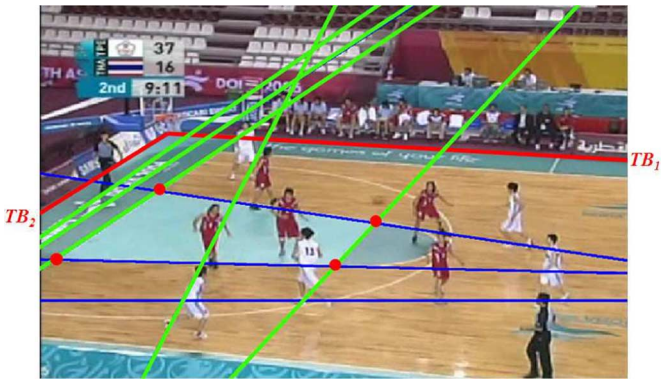


Fig. 10. Red circles indicate the four intersections of the expected quadrangle and red lines denote the detected top boundaries (i.e., TB_1 and TB_2). Green and blue lines are the detected dominant court lines near parallel to TB_1 and TB_2, respectively.

- **Observation 1:** The expected quadrangle is on the four lines of the free-throw lane; hence, intersections outside the playing field can be filtered out before generating all possible quadrangles with the aid of the predetermined playing field mask.
- **Observation 2:** As shown in Fig. 10, each corner of the free-throw lane is crossed by one blue dominant line near parallel to the first top boundary (TB_1), and one green dominant line near parallel to the second top boundary (TB_2). Moreover, each line of the free-throw lane passes through exactly 2 corners.

Algorithm 1: (Quadrangle Candidate Generation)

Given the $\{L_1, L_2, \ldots, L_N\}$ dominant court lines of a frame. Find a set of quadrangle candidates $\boldsymbol{Q}$ for court fitting.

1) Set the horizontal line set $\boldsymbol{HL}$ (lines with similar slopes as $TB_1's$), the vertical line set $\boldsymbol{VL}$ (lines with similar slopes as $TB_2's$) and the quadrangle candidate set $\boldsymbol{Q}$ as empty sets.

/* Classify each line into horizontal or vertical line set according to slope information. */

2) **for** $i = 1$ to N **do**

3) Compute $\theta_1 = |\tan^{-1} m_{TB_1} - \tan^{-1} m_{L_i}|$, $\theta_2 = |\tan^{-1} m_{TB_2} - \tan^{-1} m_{L_i}|$.

4) **if** $\min(\theta_1, \pi - \theta_1) < min(\theta_2, \pi - \theta_2)$ **then**

5) Add L_i to $\boldsymbol{HL}$.

6) **else**

7) Add L_i to $\boldsymbol{VL}$.

8) **end if**

9) **end for**

/* Compute all intersections between horizontal lines and vertical lines and record all intersections in array $\boldsymbol{A}$ with size $|\boldsymbol{VL}| \times |\boldsymbol{HL}|$. Intersections outside the playing field will be ignored according to ***Observation 1***. */

10) **for** $i = 1$ to $|\boldsymbol{VL}|$ **do**

11) **for** $j = 1$ to $|\boldsymbol{HL}|$ **do**

12) **if** the intersection p of $\boldsymbol{VL}[i]$ and $\boldsymbol{HL}[j]$ is inside the playing field **then**

13) $\boldsymbol{A}[i][j] = p$

14) **else**

15) $\boldsymbol{A}[i][j] = null$

16) **end if**

17) **end for**

18) **end for**

/* Sample four intersections to construct a quadrangle candidate according to ***Observation 2***. */

19) **for** $i = 1$ to $|\boldsymbol{VL}| - 1$ **do**

20) **for** $j = 1$ to $|\boldsymbol{HL}| - 1$ **do**

21) **for** $k = i + 1$ to $|\boldsymbol{VL}|$ **do**

22) **for** $l = j + 1$ to $|\boldsymbol{HL}|$ **do**

23) **if** $\boldsymbol{A}[i][j] \neq null$ and $\boldsymbol{A}[k][j] \neq null$ and $\boldsymbol{A}[i][l] \neq null$ and $\boldsymbol{A}[k][l] \neq null$ **then**

24) Add a quadrangle composed of $\boldsymbol{A}[i][j]$, $\boldsymbol{A}[k][j]$, $\boldsymbol{A}[i][l]$, and $\boldsymbol{A}[k][l]$ to $\boldsymbol{Q}$.

25) **end if**

26) **end for**

27) **end for**

28) **end for**

29) **end for**

30) **return** $\boldsymbol{Q}$

3) Court Model Fitting: The court model fitting step calculates the homography matrix $\mathbf{H}$ for each quadrangle candidate and the correct homography matrix $\mathbf{H}$ for current frame is obtained by selecting the one with highest fitting score. In [22], the fitting score was determined by the matching error of all line intersections in the court model and the corresponding intersections detected in the frame. In Farin's previous work [21], all points on the court line rather than intersections only were transformed from the court model coordinates to the image coordinates to evaluate the matching error. However, both methods worked under two assumptions: 1) almost all court lines of the

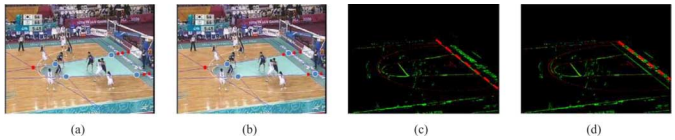


Fig. 11. Court transformation results of two similar quadrangle configurations (indicated by blue circles). (a) Correct configuration. (b) False configuration. The green pixels in (c) and (d) represent court-line pixels detected in the frame, and red pixels represent the transformation result of the court lines from the real-world court coordinates to the image coordinates using the homography matrix computed by configurations in (a) and (b), respectively.

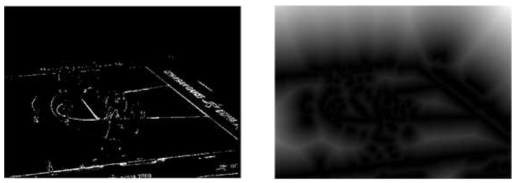


Fig. 12. Distance transform of white court line pixels. (a) Binary image of white court pixels. (b) Result of applying distance transform on the binary image in (a).

court model appear in the image, and 2) there are few noisy dominant lines in the image such that all the detected lines/intersections are quite near to the true lines/intersections of the court.

Farin's methods which only consider the line/intersection correspondence for evaluating the correctness of the homography matrix are not applicable in broadcast basketball videos since the above-mentioned assumptions fail. Fig. 11 shows the results of using two similar (but actually different) quadrangle configurations to calculate the homography matrix and transforming court lines in the court model to the image coordinates, respectively. Fig. 11(a) chooses the correct quadrangle configuration, while Fig. 11(b) chooses the wrong one due to the noisy dominant lines out of the court. Fig. 11(c) and (d) shows the reconstructed court lines from the court model using the correspondingly obtained homography matrices (as indicated by red pixels). The two configurations are very close, and the transformation results are almost the same, except for the red dashed lines in Fig. 11(c) and (d). If we select the best transformation matrix by only considering the line/intersection correspondence, both Fig. 11(c) and (d) get high scores but the system cannot successfully find the true homography matrix for the frame.

In this work, we propose a more robust court model fitting algorithm by modifying the definition of model fitting score. In addition to line correspondence, the correspondence of the playing-field area between the real-world court model and the image is another valuable hint for determining the correctness of model fitting. The geometric characteristic of the court is also taken into consideration. Therefore, in our work, the model fitting score is composed of the following three criteria.

Criterion 1: Correspondence of court lines between the image and the court model. Given a binary image indicating all white court line pixels for the current frame, we compute its Euclidean distance transform as shown in Fig. 12. A pixel gets a small value on the distance transform map if it is close to a white court line pixel. All court line points in the court model (i.e., the set P_{line}) are then transformed to the image coordinates (i.e., the set P'_{line}) using the obtained homography matrix $\mathbf{H}$, and a distance score g_1, defined by (4) is utilized as a factor to determine the court line correspondence. That is

$$g_1 = 1 - \frac{\sum_{(x,y)\in P'_{line}\cap I} dist(x,y)}{\sum_{(x,y)\in I} dist(x,y)} \tag{4}$$

where $dist(x,y)$ is the distance transform value of the pixel (x,y), and I is the set of all points in the image frame. As defined in (4), a configuration making few court line points appear in the frame after coordinates transformation will get a high score in g_1. However, a court view frame should contain most of the court line pixels; therefore, the ratio of court line points transformed into the frame [defined by (5)] is another factor for court line correspondence determination, that is

$$g_2 = \frac{|P'_{line}\cap I|}{|P_{line}|}. \tag{5}$$

We use weighted sum of the two factors to measure the correspondence of court lines between the image coordinates and the court model coordinates, that is

$$c_1 = \alpha_1 * g_1 + \alpha_2 * g_2 \tag{6}$$

where α_i is the weighting of g_i.

Criterion 2: Correspondence of playing-field area between the image and the court model. We transform points inside the playing-field area of the court model to the image coordinates. If the corresponding pixel in the image is a dominant color pixel, we add it to the set M; otherwise, we add it to the set N. The correspondence score of the playing-field area is then defined by

$$c_2 = \frac{|M| - |N|}{S_{CM}} \tag{7}$$

where S_{CM} is the total amount of points in the playing-field area of the court model.

Criterion 3: Geometric characteristic of the court. Since the court model is a rectangle, the left/right side lines of the court will have similar slopes even after being transformed to the image coordinates. Similarly, the top/bottom side lines will have similar slopes after transformation. Each pair of side lines is transformed to the image (i.e., $L'_{TopSide}$, $L'_{BottomSide}$, $L'_{LeftSide}$, *and* $L'_{RightSide}$), and we check the difference in slopes. The score for the court geometric characteristic is defined by

$$c_3 = e^{-\lambda_1 * angle\left(L'_{TopSide}, L'_{BottomSide}\right)} * e^{-\lambda_2 * angle\left(L'_{LeftSide}, L'_{RightSide}\right)} \tag{8}$$

where $angle(\cdot,\cdot)$ is the minimum included angle of two lines. λ_1 and λ_2 are normalization terms. Paired sidelines with quite different slopes in the image coordinates will result in low score in c_3.

Finally, the three scores defined by different criteria mentioned above are combined using weighted sum to measure the court model fitting score, that is

$$score = \sum_{i=1}^{3} \beta_i c_i \tag{9}$$

Fig. 13. RANSAC sampling area for dominant court line detection in the predicted-mode. The four red rectangles are search ranges centered at the four apexes detected in the previous frame.

where $\sum_{i=1}^{3} \beta_i = 1$ and β_i is the weighting of c_i. For all quadrangle candidates, we calculate the corresponding transformation matrixes $\mathbf{H}$'s and select the one having the maximum court fitting score to be the best transformation.

E. Predicted-Mode Camera Calibration

Adjacent frames in the same video clip usually have small difference in court positions since fierce camera movements will make the viewer dizzy. Once we switch to the predicted-mode camera calibration, the correct quadrangular configuration in the previous frame is used as an initial guess for the RANSAC-based dominant court line detection step in the next frame. As illustrated in Fig. 13, the system only considers pixels around the four apexes of the previous configuration to narrow down the RANSAC sampling area, and dominant lines can be extracted more accurately with the aid of a good initial guess. Since the sampling area is slashed, we once again lower the number of RANSAC iterations and speed up the calibration process. The rest of the procedures of the predicted-mode camera calibration are the same as that of the exhausting-mode.

IV. Player Tracking

Tracking players in broadcast basketball video is also a challenging task since many players wear uniforms of the same color, and occlusion of players wearing similar uniform will cause poor tracking results. In this work, we first extract player candidates based on the player mask (cf. Section II-B) of each frame in the court view shot, and track players in the image coordinates with the CamShift-based tracking algorithm. The homography matrix $\mathbf{H}$ obtained from the camera calibration step is then utilized to obtain player positions in the court model.

A. CamShift-Based Player Tracking

The CamShift tracking algorithm [23], [27] is an adaptation of the Mean Shift tracking algorithm, a non-parametric technique that climbs the gradient of a color probability distribution to find the nearest dominant mode (peak) of the distribution. However, color distributions derived from video image sequences change over time due to object movement. CamShift uses an adaptive search window with varying sizes so that it adapts to dynamically changing distributions. The conventional CamShift algorithm can be outlined in Algorithm 2 [27].

Algorithm 2: (CamShift Algorithm)

Step1. Set the region of interest (ROI) in the first frame.
Step2. Choose an initial location of the Mean Shift search window. (The selected location is the target distribution to be tracked.)
Step3. Compute a color probability distribution of the region centred at the Mean Shift search window.
Step4. Iterate Mean Shift algorithm to find the centroid of the probability image. Store the zeroth moment (distribution area) and the centroid location.

CamShift algorithm handles the following problems: 1) *Irregular object motion:* CamShift is able to handle dynamic distributions by readjusting the search window size for the next frame based on the zeroth moment of the current frame's distribution. 2) *Distractors:* CamShift is robust to distractors. Once CamShift is locked onto the mode of a color distribution, it will tend to ignore other nearby but non-connected color distributions. 3) *Transient occlusion:* CamShift tends to be robust against transient occlusion because the search window will tend to first absorb the occlusion and then stick with the dominant distribution mode when the occlusion passes. 4) *Computational efficiency:* CamShift is a simple and computational efficiency algorithm and can achieve real-time target tracking. Therefore, we modify the CamShift algorithm to be capable of tracking multiple players in broadcast basketball videos.

The detailed tracking algorithm is described in Algorithm 3. Each player region in the first frame is a possible region of interest (ROI), while some are noisy player regions (for example, referees and part of small pieces of the playing field wrongly recognized as player regions by the mask generation step due to lighting conditions). We first compute the Kullback-Leibler divergence (KL distance) [28] between the team probability distributions and the region probability distributions to filter out noisy regions (please refer to step2 of Algorithm 3). The CamShift algorithm is then applied to the remaining player regions for predicting each player's location in the next frame. To specify the initial ROI more clearly and improve the tracking results, we use the corresponding team probability distribution as initial color probability distribution for the Mean Shift step. In the following frames, if a player region is not tracked by existing trackers, we create a new tracker for it. A tracker is terminated when the player runs out of the image.

Algorithm 3: (Multiple Players Tracking Based on CamShift)

Step1. Compute team probability distributions. All pixels in the player regions of the current frame are separated into two clusters by K-means clustering based on the colors information, and each cluster represents the uniform color information of a team. We generate the color probability distribution for each cluster by using the technique of color histogram and denote the distributions for the two teams as PDF_A and PDF_B.

Step2. Create a new tracker. If a player region has not been tracked, the color probability distribution of

this region is calculated and denoted as PDF_R. The initial probability distribution $PDF_{initial}$ of the target region is set according to (10)–(12). If the $PDF_{initial}$ is non-null, a new CamShift tracker is created for this region with the corresponding $PDF_{initial}$. We set the initial location of the Mean Shift search window as the mean position of all pixels in that region, and set the initial tracking box the same as the untracked region:

$$KL(PDF_i||PDF_j)=-\sum_x PDF_i(x)\log PDF_j(x) + \sum_x PDF_i(x)\log PDF_i(x) \quad (10)$$

$$KL_S(PDF_i,PDF_j)=\frac{[KL(PDF_i||PDF_j)+KL(PDF_j||PDF_i)]}{2} \quad (11)$$

$$PDF_{initial}=\begin{cases} null, & \text{if } KL_s(PDF_R,PDF_A)\geq Th_{KL} \\ & \text{and } KL_s(PDF_R,PDF_B)\geq Th_{KL}, \\ PDF_A, & \text{if } KL_s(PDF_R,PDF_A)\leq KL_s(PDF_R,PDF_B) \\ & \text{and } KL_s(PDF_R,PDF_A)<Th_{KL}, \\ PDF_B, & \\ & \text{if } KL_s(PDF_R,PDF_A)>KL_s(PDF_R,PDF_B) \\ & \text{and } KL_s(PDF_R,PDF_B)<Th_{KL}. \end{cases} \quad (12)$$

Step3. Predict player position. For each new tracker, we apply CamShift tracking algorithm (i.e., Algorithm 2) to anticipate player's positions in the successive frames. The CamShift algorithm finds the local maximum location (p_M) of the probability distribution in the direction of gradient, and the player region in the next frame which contains the current p_M is set as tracked. For the next frame, the search window is centered at p_M and the search window size is readjusted based on the zeroth moment. The new color probability distribution is computed using the region centered at the Mean Shift search window.

Step4. Terminate a tracker. When a player runs out of the image, the probability values inside the tracking window turns to be very small. If the zeroth moment of the tracker is less than a threshold, we terminate this tracker.

B. Homography Transformation

In the CamShift-based player tracking procedure, the tracker finds the local maximum location of the probability distribution and sets it as the position of the player. This position is usually close to the mean position of the player's uniform. However, for most of the applications, we would like to know the foot position of each player rather than the uniform position. Moreover, the homography matrix $\mathbf{H}$ is derived on the basis of the ground plane. Therefore, we adjust the player position by vertically shifting it down to an approximate foot position and transform the foot position to the court model coordinates by the corresponding homography matrix. The shift amount is determined by the height of the current search window size for each player.

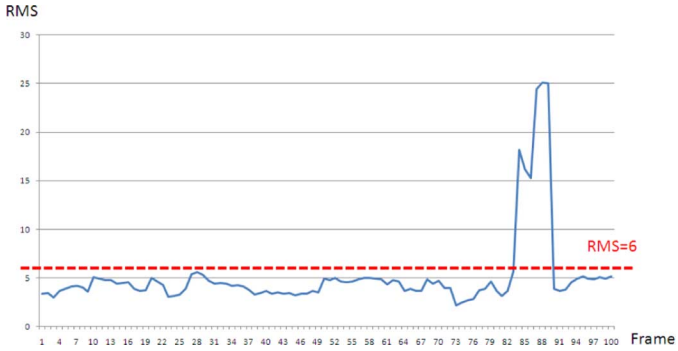


Fig. 14. Root mean square (RMS) error of the four calibrated corners for each frame in a test video using the proposed fitting score.

V. Evaluation

To demonstrate the effectiveness of the proposed court detection/camera calibration and player tracking approaches, we conducted the experiments on the video data of the 2006 Asian Games. The test videos including three women's basketball matches and three men's basketball matches recorded from live broadcast television program and compressed in MPEG-2 video standard with frame resolution of 720×480 (29.97 fps). Since labeling court/player information for all videos is time-consuming and exhausting, we manually select two offensive clips from each period of a match (i.e., $2\times4\times6=48$ video clips and 11 705 frames in total) for the evaluation of the proposed methods.

A. Performance of Court Detection and Camera Calibration

In Section III-D, three criteria including court line correspondence (c_1), playing-field area correspondence (c_2), and geometric characteristic (c_3) are considered for court fitting. To evaluate the avail of each criterion, we apply exhausting-mode camera calibration for each frame using different kinds of fitting scores. We map the four corners of the free-throw lane on the court model to the image using the obtained homography matrix $\mathbf{H}$ and calculate the root mean square (RMS) error by the ground truth positions (obtained by a user-friendly labeling tool) and the calibrated positions. Fig. 14 shows the RMS error of the four calibrated corners for each frame in a test video using the proposed model fitting score, i.e., combining all criteria and applying both exhausting-mode and predicted-mode. A new test frame is regarded as well-calibrated when each of these four calibrated points falls within d pixels away from the ground truth positions in the image, where d is set according to the mean and variance of the RMS error of the N frames having the smallest RMS error ($d=6$ pixels in this work). The accuracy of court detection/camera calibration is then defined by

$$\text{Accuracy}=\frac{\text{Number of well calibrated frames}}{\text{Number of total frames}}. \quad (13)$$

Fig. 15 shows example results of the well-calibrated and wrongly-calibrated frames using our proposed method, and Fig. 16 compares the accuracy of the court detection/camera calibration between different fitting scores. As illustrated in Fig. 16, considering each criterion individually results in bad accuracy, while our proposed fitting score (combining all the three criteria) improves the accuracy to 72.62%. We empirically adjust the weights of each criterion and observe that as long

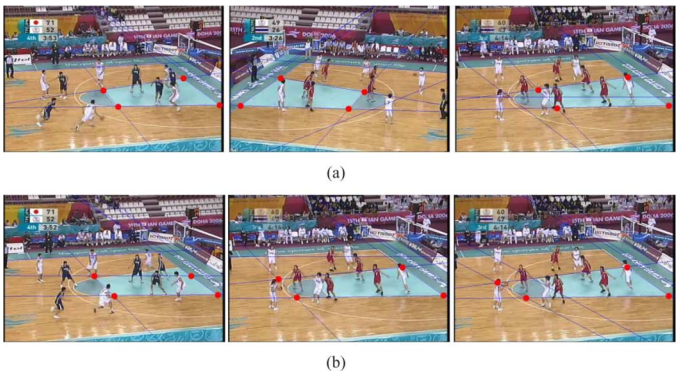


Fig. 15. Calibrated results. (a) Well-calibrated examples. (b) Wrongly-calibrated examples.

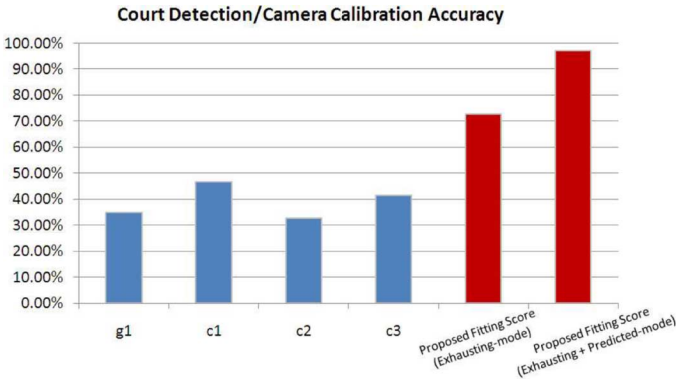


Fig. 16. Performance of the court detection/camera calibration method using different kinds of model fitting scores.

Fig. 17. Player tracking results of one test video clip. Each player is tracked and labeled with an ellipse and an ID under his/her feet.

as all the three criteria are considered (i.e., $\beta_1 \neq 0$, $\beta_2 \neq 0$, *and* $\beta_3 \neq 0$) and the correspondence of court lines (c_1) is high enough (i.e., $\beta_1 = 0.7 \sim 0.9$), we can obtain an accuracy above 60%. Finally, we set the weights as $\beta_1 = 0.8$, $\beta_2 = 0.1$, *and* $\beta_3 = 0.1$. We also evaluate the fitting score (g_1) proposed by Farin which only considers the correspondence between the detected lines in the image and the lines in the court model. Using g_1 as the fitting score apparently fails in broadcast basketball video analysis since there are plenty of noisy-lines causing a large number of false configurations.

Note that using the proposed fitting score, most of the wrongly-calibrated courts are caused by missing line candidates after applying the dominant court line detection step (such that no correct configuration is generated for the further model fitting step). However, in real applications, when several successive frames get high fitting scores, we apply predicted-mode camera calibration to the following frames and constrain the RANSAC sampling area for improving dominant court line detection. Fig. 16 shows that using the proposed fitting score, combining with the exhausting-mode and the predicted-mode achieves robust camera calibration with an average accuracy of 97.21%.

B. Performance of Player Tracking

Using the proposed CamShift-based tracking algorithm, players are automatically tracked and labeled with an ellipse and an ID under his/her feet. The color of the label indicates which team the player belongs to, and the player ID is assigned according to the created order of a track. Fig. 17 shows the player tracking results of one test video clip. Players are well-tracked from frame 239 to frame 252 even though the 9th white player is partially occluded in frame 252. The 9th white player is missed in frame 254 due to nearly fully-occluded by the 7th blue player for several frames, and then be tracked with a new ID when the player appears again in frame 261. In frame 280, the 7th blue player is missed due to the well-known "error merge" problem, which means the trackers of the same team lose their associated objects and falsely coalesce with other objects when two or more players of the same team occlude with each other.

We manually check the player tracking results (on the basis of hit, miss, and false alarm) of each frame in the 48 test video clips. A hit means the detected foot position really belongs to one player and the tracked ID in the current frame is consistent with the corresponding tracked ID in the previous frame. A miss means a player is not detected/tracked in the current frame. A detected foot position is regarded as a false alarm when it does not belong to any player in the frame or the tracked ID in the current frame is inconsistent with that of the previous frame. Fig. 18 shows the precision and recall of the proposed tracking algorithm for each video clip. The precision and recall are defined by

$$\text{Precision} = \frac{\text{Number of hit}}{\text{Number of hit} + \text{Number of false alarm}} \tag{14}$$

and

$$\text{Recall} = \frac{\text{Number of hit}}{\text{Number of hit} + \text{Number of miss}}. \tag{15}$$

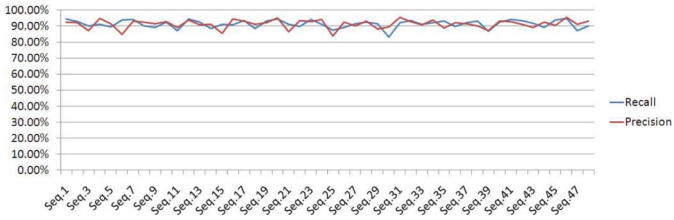


Fig. 18. Precision and recall of the proposed tracking algorithm for each video clip.

The average precision and recall values are 91.38% and 91.34%, respectively. Most of the false alarms are caused by recognizing referees as players when the color of the referee's uniform is similar to that of the player's, or by noisy regions having the same color distribution as that of the player's uniform. Misses mainly result from the previously mentioned "occlusion" and "error merge" problem. Some misses are caused by the failure of player detection when the player stands near the boundary of the court and most of his/her uniform are out of the playing field. However, players seldom occlude with each other or stand too near the court boundary for a long time in a basketball game, and the system will create a new tracker on the untracked player when he/she is detected or split from the merged objects again. Hence, we can achieve an acceptable performance above 91%.

An association-based method [29] can successfully help us to mine some players who are originally missed in our system. However, the association method loses the correct trajectory identity in basketball videos when the moving direction of a player changes greatly after being occluded by another player wearing the uniform of the same color (this situation occurs when the offensive team performs tactics such as "screen" or "pick and roll"). We will focus on this issue in our future work.

VI. Applications

Player trajectories in the court model coordinates provide more practical semantics than low-level visual/aural/camera motion features, especially for professional players/coaches. Based on the mapped player trajectories, we introduce three novel applications which meet professionals' demands, including highlight extraction, video retrieval, and tactic analysis.

A. Highlight Extraction Based on Wide-Open Detection

Previous works automatically generated highlights with dunk, fast break, or score events using state-of-the-art video analysis techniques. However, these kinds of audience-oriented events are less informative for professionals since a dunk is an individual playing skill and most of the fast break or score events occur without specific tactics. Viewing the highlight extraction issue from the professional perspective, video clips containing possible tactics should have higher priority to be included in the highlights. In basketball games, wide-open means some offensive player is not well defended by his/her opponents, and implies the occurrence of a successful offense tactic. Therefore, we extract wide-open events as highlights for professionals to observe possible tactics performed by the offensive team. Based on the absolute/relative player positions in the court model coordinates, we design a highlight extraction mechanism as depicted in Algorithm 4. All thresholds and weighting parameters can be determined by professional users according to their requirements. Users can also set a time constraint of the overall highlight and the system will automatically adjust $Th_{importance}$ to extract suitable video clips. Moreover, for each frame of the extracted highlight, we indicate the player who is wide-open by a colored warning-circle under his/her feet (as shown in Fig. 19), so that the professional can focus on this player to discover the intention and the routes of the involved tactics.

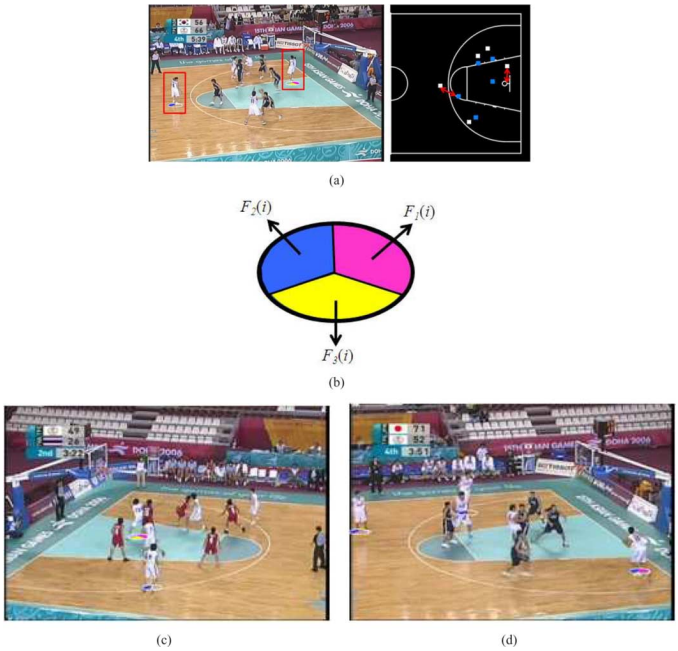


Fig. 19. Wide-open detection. (a) Player trajectories are mapped to the real-world court model and a player with high danger score is indicated with a warning circle. (b) Warning circle contains three hints: the pink part means the player is close to the basket, the blue part means the player has no close defender, and the yellow part means there is no defender between the player and the basket. (c) and (d) are two example frames showing players who are in wide-open situations.

Algorithm 4: (Professional-Oriented Highlight Extraction)

For each video clip with trajectory information in the court model coordinates:

1) **Determine the offensive/defensive team.** Find the first calibrated frame, and assign each player to one of the two teams by the difference of the color distributions (cf. Step 1 and Step 2 in Algorithm 3). The offensive team (OT) and the defensive team (DT) are then determined by

$$OT = \underset{team}{arc}\max \sum_{i \in team} Dis(post, i) \tag{16}$$

and

$$DT = \underset{team}{arc}\min \sum_{i \in team} Dis(post, i) \tag{17}$$

where $Dis(post, i)$ denotes the distance between the basket post and a player i.

2) **Find player who is wide-open (not well-defended) in each frame.** According to the defensive principle of basketball, an offensive player i has higher

possibility to score when the player is close to the basket, the nearest defender is at a distance from him/her, or there is no defender between the player and the basket (as formulated in (18)–(20)):

$$F_1(i) = \exp\{-\rho_1 * Dis(post, i)\} \tag{18}$$

$$F_2(i) = \frac{\min_{j \in DT} Dis(i,j)}{CL} \tag{19}$$

$$F_3(i) = \begin{cases} 1 & \textit{if there is no defensive player on the line (with width } \tau\textit{) between the offensive player and the basket post} \\ 0, & \textit{otherwise} \end{cases} \tag{20}$$

where ρ_1 and CL (the length of the half court) are normalization constants. Therefore, we examine whether an offensive player i is wide-open by a danger score, that is

$$Danger(i) = \sum_{k=1}^{3} \eta_k F_k(i) \tag{21}$$

where $\sum_{k=1}^{3} \eta_k = 1$ and η_k is the weighting of $F_k(i)$.

3) **Determine the importance of the current clip.** A successful offensive tactic results in more than one player being possible to shoot. We sum up danger scores of all offensive players in all frames of the current clip, and determine the importance of the clip as

$$Importance = \frac{1}{N} \sum_{i \in OT} Danger(i) \tag{22}$$

where N is a normalization term representing the number of calibrated frames in the current clip.

Sort all video clips according to the importance scores and generate highlights by concatenating all video clips having scores higher than $Th_{importance}$.

Applying the first step of Algorithm 4, we successfully assign the offensive/defensive team for all the 48 test video clips. The precision-recall [as respectively defined in (14) and (15)] is used to evaluate the accuracy of the automatic wide-open detection algorithm. One professional basketball coach is asked to watch 15 test video clips (totally 3672 frames) and indicate the feet positions of wide-open players in each frame through a simple user interface. The data labeled by the coach is taken as the ground truth (GT1), and the according precision-recall of the proposed wide-open detection algorithm is shown in the first row of Table I. We observe that even if the coach is well-trained to recognize a wide-open event, she might ignore some wide-open players because her attention is attracted by other noticeable wide-open players appearing in the same time. To acquire more reliable ground truth, we ask the coach to label on the 15 video clips with wide-open alarms automatically added by our system. We take the relabeled data as ground truth (GT2) to evaluate Algorithm 4, and the corresponding precision-recall is shown in the second row of Table I. Both the precision and the recall are improved by using GT2 as the ground truth, which means the automatic wide-open detection system can help the coach find wide-open players more effectively. The professional basketball coach also approved that the extracted highlights with wide-open alarms can help her find possible tactics in the video.

TABLE I
PRECISION AND RECALL OF THE WIDE-OPEN PLAYER DETECTION

	Precision	Recall
Wide-open detection evaluated with GT1	0.772 (1035/1341)	0.756 (1035/1369)
Wide-open detection evaluated with GT2	0.842 (1208/1434)	0.831 (1208/1453)

B. Video Retrieval Based on Player Trajectories

Mining explicit tactics in a basketball video is difficult since the extracted player trajectories are noisy due to plenty of player occlusions. Instead of developing an automatic tactic-mining methodology, using an electronic tactics board to interactively retrieve videos based on player trajectories would be more applicable. During a basketball game, the coach usually asks for a timeout to arrange tactics according to the current situation, and the players have to quickly grasp the tactics. Unfortunately, most of the basketball players cannot exactly memorize and perform a new tactic in such a short period. It would be helpful to develop an electronic tactics board to facilitate tactic-sketching for the coach, and help the players clearly understand the tactics by showing video examples with similar player trajectories.

We have implemented a preliminary prototype of the electronic tactics board and the demo can be watched at http://www.youtube.com/watch?v=7JTzpkVt3tY. A two-step hierarchical method is adopted to find similar tactics. The first step roughly filters out dissimilar tactics by computing the distance map between the query tactic (Q) and each of the tactics (T) in the database. The sum of all distance values in the map is denoted as $D_{map}(Q,T)$ and only these tactics with small $D_{map}(Q,T)$ are considered as possible similar tactics of the query tactic. The second step computes the tactics similarity more precisely by checking trajectory similarities between the query tactic and each of the possible similar tactics. Given two tactics $Q = \{Q_1, Q_2, \ldots, Q_m\}$ and $T = \{T_1, T_2, \ldots, T_n\}$, each element in Q or T indicates a trajectory in the tactics. We pairwisely compute trajectory distance $EDR(Q_i, T_j)$ based on the *Edit Distance on Real sequence* [30], which has been proved to be a robust distance function against data imperfections such as noise, shifts, and scaling in trajectories. The tactic distance between Q and T is then defined as

$$D_{tactic}(Q,T) = \frac{1}{|T|} \sum_{T_j \in T} \min_{Q_i \in Q} EDR(Q_i, T_j) \tag{23}$$

and we retrieve k tactics with the smallest $D_{tactic}(Q,T)$ values from the tactics database. Currently, the system can automatically retrieve similar tactics in the tactics database (with complete trajectory information) according to the input trajectories. We are still working on this topic and will include the video database (with incomplete and noisy trajectory information) into our system in the near future.

C. Defensive Tactic Analysis Based on Player Trajectories

The previous two applications mainly focus on offensive tactics, while defensive tactics are also important in a basketball

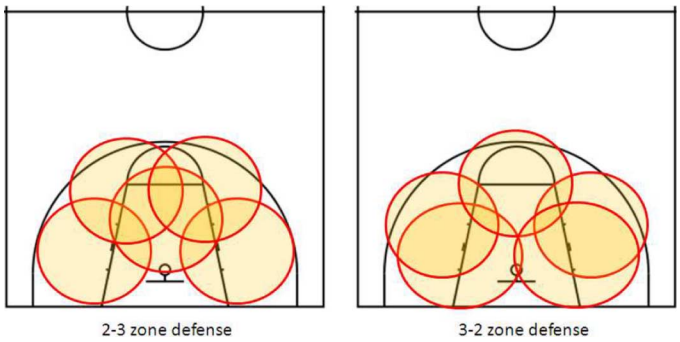


Fig. 20. Guard areas of each defensive player in 2–3 zone defense and 3–2 zone defense.

game since knowing the main defensive plan of the opponent helps the coach figure out better counterplots. Defensive tactics can be mainly categorized into two types: one-on-one defense (in which an individual defensive player guards an individual offensive player) and zone defense (in which each defensive player guards a specific area rather than a specific offensive player). As shown in Fig. 20, each defensive player of a zone defense strategy has an area to guard and they seldom step to other areas. Therefore, we evaluate the stability of defensive players according to their trajectories and recognize the possible defensive tactic for each video clip.

Given the five predefined defense zones $\{Z_1, \ldots Z_5\}$, each player trajectory PT is first assigned to Z_* by

$$Z_* = \arg\min_{Z_i} Dis(Centroid(Z_i), Centroid(PT)) \quad (24)$$

where $Centroid(Z_i)$ is the centroid of the zone Z_i, $Centroid(PT)$ is the average coordinates of all positions in the trajectory PT, and $Dis(A, B)$ denotes the distance between the two points A and B. The stable score of the player trajectory PT is then calculated by

$$StableScore(PT) = \frac{1}{|PT|} \sum_t Z_*(t) \quad (25)$$

where $Z_*(t) = 1$ when the position at time t is in the assigned zone Z_*; otherwise, $Z_*(t) = 0$. $|PT|$ is the number of frames in that trajectory. A video clip is recognized as a zone defense if the average stable score of all contained trajectories is larger than Th_s (Th_s is empirically set as 0.65). We ask a professional basketball coach to classify each of the 48 video clips into one-on-one defense or zone-defense. However, not every video clip can be clearly classified because the movements of defensive players are usually influenced by the offensive players and sometimes the strategy of the offensive players makes the one-on-one defense look like a zone-defense (and vice versa). Therefore, only six one-on-one defense clips and five 2–3 zone defense clips are used to evaluate the performance of the proposed defensive tactic analysis application. Among the 11 video clips, two one-on-one defense clips are wrongly classified as zone-defense because the trajectory of the same player is divided into several short trajectories due to occlusions, which results in the increase of the average stable score.

VII. Conclusions and Discussion

Semantic analysis of broadcast basketball video is challenging and previous works only focus on highlight extraction on the basis of some audience-based events. In this paper, we extract the player positions relative to the court by a robust camera calibration method and a CamShift-based player tracking algorithm. To more accurately and efficiently acquire player positions, a preprocessing step is designed to sieve out informative video clips and to generate useful masks. Player positions can be further utilized as high-level information for several professional-oriented applications which are more practical and usable as compared with conventional works based on low-level features. Through observing informative video clips with wide-open event warnings, retrieving video clips with tactic information, and acquiring statistics of player positions, a professional coach can efficiently summarize the key strategy of their opponents and train a basketball team to conquer all difficult situations in the game.

Ball trajectory is also crucial information for tactic analysis, while tracking the ball in broadcast basketball video is still an unsolved problem. In the future, we will focus on ball tracking and combine the player/ball trajectories to automatically analyze more detailed offensive/defensive strategies in broadcast basketball videos. In addition to assisting the professional by acquiring tactic information from game videos, we will also monitor the training process of a basketball team, and apply video analysis techniques to evaluate if all players correctly perform a given tactic.

References

[1] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," in *Proc. IEEE Int. Conf. Multimedia Computing and System*, 1995, pp. 167–174.
[2] 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," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2001, pp. 721–724.
[3] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden Markov models," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2002, pp. 4096–4099.
[4] B. Li and I. Sezan, "Event detection and summarization in American football broadcast video," *Storage and Retrieval for Media Databases (SPIE)*, vol. 4676, pp. 202–213, 2002.
[5] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Proc. ACM Int. Conf. Multimedia*, 2000, pp. 105–115.
[6] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 68–75, 2002.
[7] C. Xu, J. Wang, H. Lu, and Y. Zhang, "A novel framework for semantic annotation and personalized retrieval of sports video," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 325–329, 2008.
[8] W.-T. Chu and J.-L. Wu, "Explicit semantic events detection and development of realistic applications for broadcasting baseball videos," *Multimedia Tools Appl.*, vol. 38, no. 1, pp. 27–50, 2007.
[9] D. Zhang and S.-F. Chang, "Event detection in baseball video using superimposed caption recognition," in *Proc. ACM Int. Conf. Multimedia*, 2002, pp. 315–318.
[10] J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. pala, "Soccer highlights detection and recognition using HMMs," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2002, pp. 825–828.
[11] V. Tovinkere and R. J. Qian, "Detecting semantic events in soccer games: Towards a complete solution," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2001, pp. 1040–1043.
[12] S. S. Intille and A. F. Bobick, "Recognizing planned, multi-person action," *Comput. Vis. Image Understand.*, vol. 81, pp. 414–445, 2001.
[13] G. S. Pingali, Y. Jean, and I. Carlbom, "Real time tracking for enhanced tennis broadcasts," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 1998, pp. 260–265.
[14] A. Guéziec, "Tracking pitches for broadcast television," *IEEE Comput.*, vol. 35, no. 3, pp. 38–43, 2002.

[15] G. Zhu, C. Xu, Q. Huang, Y. Rui, S. Jiang, W. Gao, and H. Yao, "Event tactic analysis based on broadcast sports video," *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 49–67, Jan. 2009.
[16] W. Zhou, A. Vellaikal, and C.-C. J. Kuo, "Rule-based video classification system for basketball video indexing," in *Proc. ACM Int. Conf. Multimedia*, 2000, pp. 213–216.
[17] D. D. Saur, Y.-P. Tan, S. R. Kulkarni, and P. J. Ramadge, "Automated analysis and annotation of basketball video," *Storage and Retrieval for Image and Video Databases (SPIE)*, vol. 3022, pp. 176–187, 1997.
[18] S. Liu, M. Xu, H. Yi, L.-T. Chia, and D. Rajan, "Multimodal semantic analysis and annotation for basketball video," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1–13, 2006.
[19] Y. Zhang, C. Xu, Y. Rui, J. Wang, and H. Lu, "Semantic event extraction from basketball games using multi-modal analysis," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2007, pp. 2190–2193.
[20] H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, and S.-Y. Lee, "Physics-based ball tracking and 3D trajectory reconstruction with applications to shooting location estimation in basketball video," *J. Vis. Commun. Image Represent.*, vol. 20, no. 3, pp. 204–216, 2009.
[21] D. Farin, S. Krabbe, P. H. N. de With, and W. Effelsberg, "Robust camera calibration for sport videos using court models," *Storage and Retrieval Methods and Applications for Multimedia (SPIE)*, vol. 5307, pp. 80–91, 2004.
[22] D. Farin, J. Han, and P. H. N. de With, "Fast camera calibration for the analysis of sport sequences," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2005, pp. 80–91.
[23] G. R. Bradski, Intel Corp., *Open Source Computer Vision Library Reference Manual,* 2001, pp. 123456-001–123456-001.
[24] A. Hanjalic, "Shot-boundary detection: Unraveled and resolved?," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 2, pp. 90–105, 2002.
[25] Y. Liu, S. Jiang, Q. Ye, W. Gao, and Q. Huang, "Playfield detection using adaptive GMM and its application," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2005, pp. 421–424.
[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.
[27] J. G. Allen, R. Y. D. Xu, and J. S. Jin, "Object tracking using CamShift algorithm and multiple quantized feature spaces," in *Proc. Pan-Sydney Area Workshop Visual Information Processing*, 2003, pp. 3–7.
[28] T. M. Cover and J. A. Thomas*, Elements of Information Theory*. New York: Wiley-Interscience, 1991.
[29] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
[30] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2005, pp. 491–502.

Min-Chun Hu (S'08) received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively. She is currently pursuing the Ph.D. degree in the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan.

Her research interests include digital signal processing, digital content analysis, pattern recognition, computer vision, and multimedia information retrieval.

Ming-Hsiu Chang received the B.S. degree in computer science and information engineering from National Chung Cheng University, Minhsiung, Taiwan, in 2007 and the M.S. degree from National Taiwan University, Taipei, Taiwan, in 2009.

Since 2009, he has been an Engineer with Cyberlink Corp., Taipei. His research interests include image processing and digital content analysis.

Ja-Ling Wu (F'08) received the Ph.D. degree in electrical engineering from Tatung Institute of Technology, Taipei, Taiwan, in 1986.

From 1986 to 1987, he was an Associate Professor of the Electrical Engineering Department, Tatung Institute of Technology. In 1987, he transferred to the Department of Computer Science and Information Engineering (CSIE), National Taiwan University (NTU), Taipei, where he is presently a Professor. From 1996 to 1998, he was assigned to be the first Head of the CSIE Department, National Chi Nan University, Puli, Taiwan. During his sabbatical leave (from 1998 to 1999), he was invited to be the Chief Technology Officer of the Cyberlink Corp. In this one-year term, he was involved with the developments of some well-known audio-video software, such as the PowerDVD. Since August 2004, he has been appointed to head the Graduate Institute of Networking and Multimedia, NTU. He has published more than 200 technique and conference papers. His research interests include digital signal processing, image and video compression, digital content analysis, multimedia systems, digital watermarking, and digital right management systems.

Dr. Wu was the recipient of the Outstanding Young Medal of the R.O.C. in 1987 and the Outstanding Research Award of the National Science Council, R.O.C., in 1998, 2000, and 2004, respectively. In 2001, his paper "Hidden digital watermark in images" (coauthored with C.-T. Hsu), published in the IEEE TRANSACTIONS ON IMAGE PROCESSING, was selected to be one of the winners of the Honoring Excellence in Taiwanese Research Award, offered by ISI Thomson Scientific. Moreover, his paper "Tiling slideshow" (coauthored with his students) won the Best Full Technical Paper Award in ACM Multimedia 2006. He was selected to be one of the lifetime Distinguished Professors of NTU in November 2006. He was elected as an IEEE Fellow in 2008 for his contributions to image and video analysis, coding, digital watermarking, and rights management.

Lin Chi received the B.S. degrees from Ming Chuan University School of Communication, Taipei, Taiwan, and the Department of Physical Education, Fu Jen Catholic University, Taipei, in 1995, the M.S. degree in physical education from the University of New Orleans, New Orleans, LA, in 1998, and the Ph.D. degree in university sport teaching and training from Shanghai University, Shanghai, China.

Since 1998, she has been a teacher in the Physical Education Center of Ta Hwa Institute of Technology, Hsinchu County, Taiwan.