Court Reconstruction for Camera Calibration in Broadcast Basketball Videos

Pei-Chih Wen, Wei-Chih Cheng, Yu-Shuen Wang, Hung-Kuo Chu,
Nick C. Tang, and Hong-Yuan Mark Liao, Fellow, IEEE

Abstract—We introduce a technique of calibrating camera motions in basketball videos. Our method particularly transforms player positions to standard basketball court coordinates and enables applications such as tactical analysis and semantic basketball video retrieval. To achieve a robust calibration, we reconstruct the panoramic basketball court from a video, followed by warping the panoramic court to a standard one. As opposed to previous approaches, which individually detect the court lines and corners of each video frame, our technique considers all video frames simultaneously to achieve calibration; hence, it is robust to illumination changes and player occlusions. To demonstrate the feasibility of our technique, we present a stroke-based system that allows users to retrieve basketball videos. Our system tracks player trajectories from broadcast basketball videos. It then rectifies the trajectories to a standard basketball court by using our camera calibration method. Consequently, users can apply stroke queries to indicate how the players move in gameplay during retrieval. The main advantage of this interface is an explicit query of basketball videos so that unwanted outcomes can be prevented. We show the results in Figures 1, 7, 9, 10 and our accompanying video to exhibit the feasibility of our technique.

Index Terms—Camera calibration, basketball, stroke, player trajectory, video retrieval

1 Introduction

Camera calibration for broadcast sport videos has been thoroughly studied in the past few years. Calibrated scenes are particularly beneficial to applications such as tactical analysis, summarization, and virtual realities. A straightforward idea to achieve camera calibration is reconstructing a 3D basketball court by using structure from motion. But this approach will fail when the camera of a court view shot video does not contain translation [1]. Hence, most previous methods apply a 2D model to achieve the aim. The first approach is to detect intersecting points of court lines, and subsequently mapping the points to the predefined court with a homography [2], [3], [4], [5], [6], [7], [8]. Given that the recognition of court lines is challenging due to noise, player occlusions, and illumination conditions, the derived homographies are not reliable, and the calibration usually fails. Another approach is to manually align a court model on the first frame of a video clip and then applying the iterated closest point (ICP) method to estimate a homography between the model points and court line pixels in consecutive frames [9]. The approach achieves robust and reliable calibration. However, manual registration of the first frame in each video clip is tedious because a broadcast video usually contains many shot changes. In general, the mentioned techniques consider only spatial features to handle camera calibration and inevitably suffer from reliability problem or demand tedious manual registration.

We generate the panoramic image from a court view shot video that covers the whole basketball court to achieve camera calibration. Our system estimates a homography transformation based on the tracked Kanade-Lucas-Tomasi (KLT) features [10] between consecutive frames and transforms each frame to an identical coordinate system. Colors of the pixels transformed to the same position are linearly blended. After that, we detect the court region in the panoramic image based on the dominant color [6]. Considering that panoramic court may be distorted due to accumulated transformation errors, we further warp the court to a quadrangle. Finally, by employing the obtained corner correspondence, we rectify the quadrangular basketball court to a standard one using a homography to remove the perspective effect. Since our court generation can project pixels from a frame to a basketball court, this procedure is also able to transform player positions between the two coordinate systems. The success of our method is due to the court reconstruction that considers all video frames in a clip. Missing court features in one frame can

- *P. C. Wen, W. C. Cheng and Y. S. Wang are with the Department of Computer Science, National Chiao Tung University, Taiwan*
  E-mail: pjwen0329@gmail.com
  E-mail: ws6501@gmail.com
  E-mail: yushuen@cs.nctu.edu.tw
- *H. K. Chu is with the Department of Computer Science, National Tsing Hua University, Taiwan*
  E-mail: hkchu@cs.nthu.edu.tw
- *N. C. Tang and H. Y. Mark Liao are with the Institute of Information Science, Academia Sinica, Taiwan*
  E-mail: nickctang@gmail.com
  E-mail: liao@iis.sinica.edu.tw

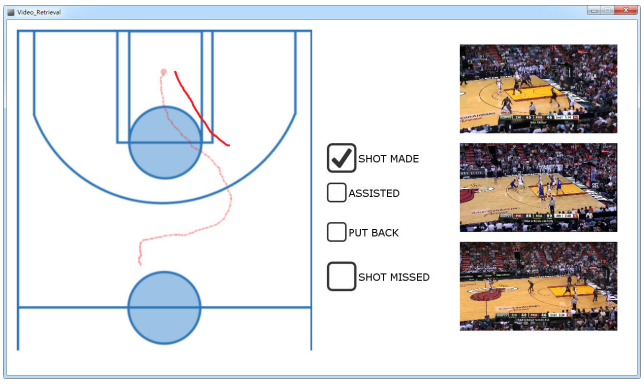


Fig. 1. (Left) The complete and half saturated red curves denote the user-specified stroke and extracted player trajectory, respectively. (Middle) Event filter. (Right) Retrieved basketball videos.

be obtained from other frames. Hence, our system achieves not only full automation but also robustness.

Our automatic camera calibration works when the video contains a whole basketball court. For the remaining videos, the calibration requires an additional mapping to a reference video that covers a complete court. Therefore, we determine the court completeness by identifying whether the video covers the left and the right courts simultaneously. We then select a video clip with a complete court as a reference and map other videos to this reference based on the tracked KLT features. The strategy enables our system to map player trajectories from all court view shot videos to an identical coordinate, regardless of court completeness of a video.

To demonstrate the feasibility of our technique, we present a stroke-based system that allows users to retrieve basketball videos by specifying player trajectories. Unlike existing retrieval systems, which generally rely on local features such as color, texture, shape, and spatial relations, the presented approach supports semantic queries during retrieval. These stroke queries are very useful in basketball video retrieval because player trajectories can be used to compose high level tactics that cannot be described by low level features. In addition, considering that an event is normally achieved in different ways due to various offensive and defensive strategies, a text-based retrieval system is insufficient. Users may intend to specify where a player cuts in and makes a shot to prevent unwanted results. Therefore, we track player trajectories from broadcast basketball videos and rectify the trajectories based on the presented camera calibration technique. When a stroke query is given, the videos in which player trajectories best fit the query are returned as the searching results. This intuitive interface effectively reduces the semantic gap of basketball video retrieval between users and machines (Figure 1).

We present a robust and fully automatic approach of calibrating camera motions in broadcast basketball videos. In particular, it rectifies player trajectories to the same coordinate system and enables stroke-based basketball video retrieval. Although our system is introduced to handle basketball videos, the same methodology can be applied to many other sport videos. We show the user-specified strokes and the retrieved videos in Figures 1, 7, 9, 10 and our accompanying video to demonstrate the feasibility of this novel interface. We also presented our system to college basketball players to evaluate its usability. They show high preference after using our system.

2 Related Work

Basketball video processing. Basketball video processing has received increasing attention in recent years because of the urgent need from coaches and spectators. The techniques were presented to track ball positions [11], [12], [13], identify players [9], [14], shots [15], enhance visual experience [4], and summarize games [16], [17]. The information extracted from this procedure is useful in tactical analysis and development.

Camera calibration. Mapping player positions from each video frame to a basketball court coordinate is essential because the rectified information can be further analyzed for many applications. The goal is often achieved with a homography because a basketball court can be regarded as a plane. Homography provides quick mapping and is particularly useful in intensive camera motions. Specifically, two classes of algorithms are presented to map features from video frames to a basketball court coordinate. The first class [2], [3], [4], [5], [6], [7], [8] detects semantic features such as court lines and corners, and maps the features to the predefined court. Given that these features are usually occluded by players in broadcast basketball videos, Hu et al. [6] extended the algorithm introduced by Farin et al. [2], [3] and further detected free throw line from video frames to enhance calibration quality. However, due to noise, illumination changes, and player occlusions, this method has high failure rate of detecting court features and is considered less reliable. The second class of the algorithms [9], [18], [19], [20] is based on a predefined court model manually registered to the first frame of a video. The model points are matched with the detected edge pixels in the next video frame and so on by conducting the ICP method to determine the homographies. A model-based homography estimation enjoys robustness but suffers from tedious manual registration. In contrast, our algorithm reconstructs a basketball court from the video, followed by mapping the four corners of this court to calibrate camera motions. The consideration of all video frames in a clip enables our method to achieve robustness and full automation.

Video retrieval. Most video retrieval methods use either visual or non-visual features to compare content similarity. A brief descriptor used to represent

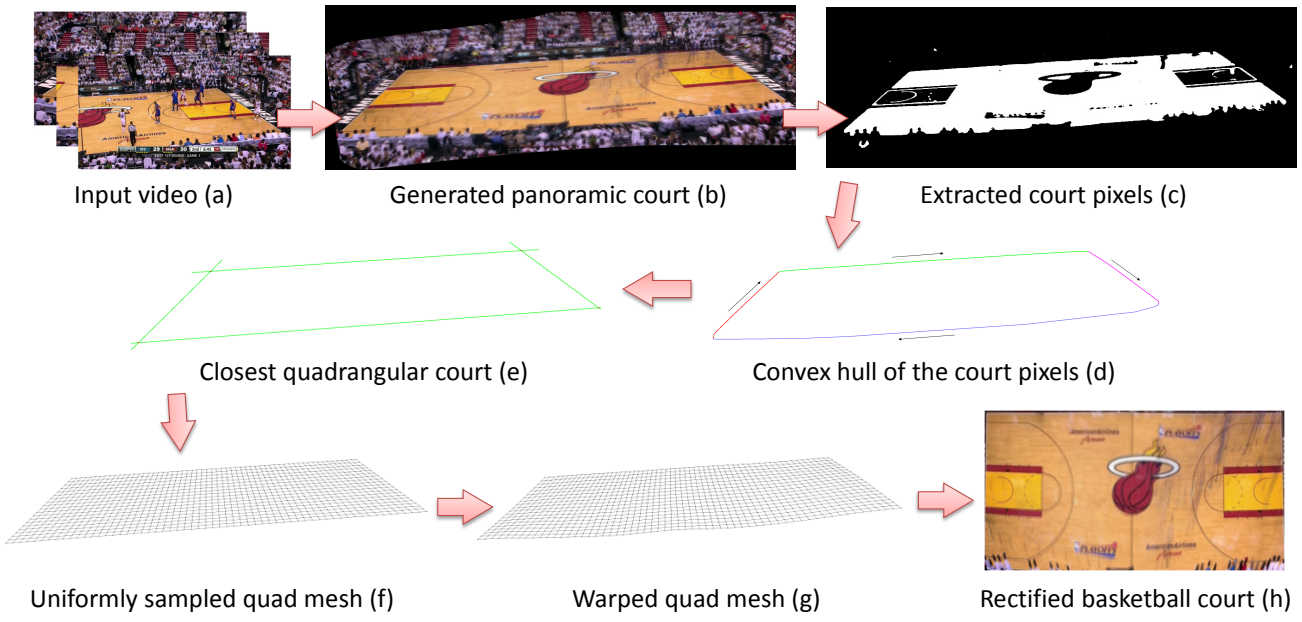


Fig. 2. Flowchart of our algorithm. Given a basketball video (a), a panoramic court is generated (b) based on the tracked KLT features, and court pixels are extracted (c) according to the dominant color. The court may contain logos, lines, and spectators; hence, our system computes a convex hull (d) of the component that connects the most court pixels to represent the court region. Hull boundary edges are then partitioned into four groups based on their orientations. Because each group of edges can be approximated using a straight line, a quadrangle (e) that best fit the court shape is obtained. Considering that the panoramic court is not exactly quadrangular, we represent the quadrangle (e) using a quad mesh (f), warp the mesh to fit the panoramic court (g), and apply the inverse warping to rectify the court. Finally, the perspective effect is removed using a homography (h).

a video is especially important for executing video browsing, querying, and navigation in a large-scale database [21]. Previous methods retrieve videos by considering motion flows [22], bag-of-features [23], or textual annotation [24], to determine whether a retrieved video matches the input query. Since this work focuses on camera calibration, we refer readers to the survey paper [25] for more details.

All of the above methods rely on low level features. Using these methods to retrieve basketball videos, however, often incurs unwanted results because low level features are not informative enough to describe strategies in a game. In contrast, our system allows users to specify player trajectories and events during retrieval, which effectively reduces the semantic gap between humans and machines.

3 Camera Calibration

Camera calibration enables the transformation of player positions from a video coordinate to a basketball court coordinate. The objective is generally achieved by multiplying a homography because a basketball court can be considered a plane. However, estimating the homography is challenging because the correspondence between low level features of a video frame and high level definitions of a basketball court is unknown. To achieve robustness and full automation, we calibrate camera motions with the consideration of the whole video clip rather than a single frame to prevent player occlusions and sudden illumination changes. This work primarily aims to generate a panoramic basketball court from a video. According to the fact that this court is linearly projected from the real world, the four corners on the reconstructed and the standard basketball courts can be easily corresponded. We apply the mapped corners to compute homographies and calibrate camera motions. Figure 2 shows the flowchart of our calibration method.

3.1 Calibration in complete court videos

We generate the panorama from a video that contains the whole basketball court. To determine the court completeness of a video, we verify whether the lower corners of the left court and the right court appear in the video simultaneously. That is, the court region is extracted from a video based on the dominant color [6] because the court generally occupies most central area in court view shot videos. To enhance robustness against noise introduced by illumination, each pixel color is first converted to YCrCb space. Our system then computes a 16×16 histogram embedded with Cr and Cb channels to represent the video. The bin with the most pixels stands for the dominant color and the

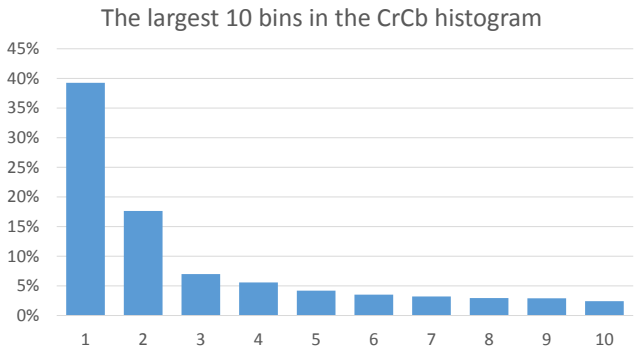


Fig. 3. The bar charts show the largest 10 bins in the CrCb histogram, where the the horizontal and vertical dimensions indicate the bin index and the percentage of pixels in a bin.

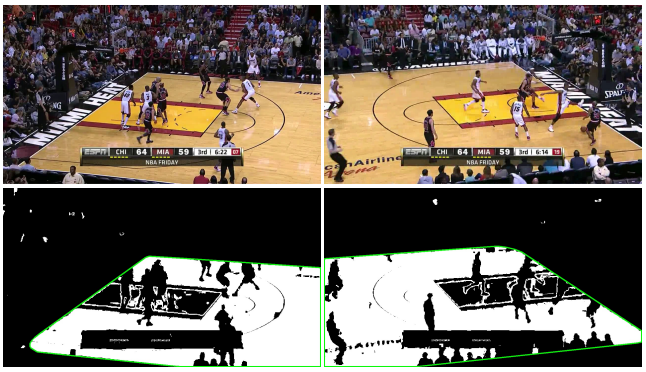


Fig. 4. (Top) Left and right images show the left and right court views, respectively. (Bottom) The components that connect most of court pixels (white) and corresponding convex hulls (green) are shown. Clearly, convex hulls in these two viewpoints contain only one vertical boundary. We apply this criteria to detect if both left and right court views appear in a video clip.

pixels in that bin are considered *court pixels*. It is worth noting that, the focus of a basketball video is the court region. Hence, we multiply a central weight to each pixel when it is added to the histogram. Formally, the weight is given by $w = d_i/d_{max}$, where d_i is the distance from pixel i to the closest top or bottom boundary and d_{max} is half of the image height. We also point out that this approach can stably extract the court region under various illumination because flashlights appear only in few frames. Figure 3 shows the color histogram of a video clip. The largest bin is composed of court pixels, which is 2.2x larger than the second largest bin.

Considering that the detected court region is not perfect due to the disturbances of noise, players, and score board, the component that connects the most court pixels is selected, and its convex hull is computed to represent the *court region*. Observing that the corner appears as a point (Figure 4), a frame contains the left (right) lower corner if a long vertical hull edge appears at right (left). In other words, we determine the court completeness based on the existence of these two video frames.

3.1.1 Panoramic court generation

We generate a panoramic basketball court based on the OpenCV[1] tracked KLT features [10] in consecutive frames. Let $\mathbf{P}^f = \{\mathbf{p}_1^f, \mathbf{p}_2^f, ..., \mathbf{p}_n^f\}$ be a set of KLT feature positions in frame f, where $\mathbf{p} = \{\mathbf{p}_x, \mathbf{p}_y\} \in R^2$ and n is the total number of features. Considering basketball courts are planar, our system computes a homography that can transform features from $\mathbf{P}^f$ to the corresponding positions $\mathbf{P}^{f-1}$.

Our system rejects the KLT features on a score board and on players to enhance the robustness of court reconstruction. It also rejects pixels within those regions when blending the panoramic court. Observing that the score board is fixed at a certain position in most broadcast basketball videos, we manually specify the region. For the player regions, they are automatically detected by using the deformable part model [26] and the tracking algorithm [27].

3.1.2 Quadrangular court detection

Similar to the mechanism of court extraction in each video frame, the basketball court is obtained from the panoramic image based on the dominant color. The largest component connecting court pixels is selected (Figure 2(c)) and the convex hull of this component is determined to resist noise. Given that the court region is linearly projected from a rectangular court, this court must be quadrangular. Hence, we compute a quadrangle that approximates the court region. We first trace the hull boundary in the clockwise direction and get the angle of each edge (Figure 2(d)). We then apply K-means algorithm to partition edges on the hull boundary into four groups according to the traced edge angles. After that, a straight line is estimated to approximate edge centroids in each group using the principal component analysis [28] because these centroids should be collinear. Once the four straight lines are determined, the quadrangle representing the basketball court is obtained by computing the intersecting points (Figure 2 (e)).

3.1.3 Quadrangular court warping

The basketball court in a panorama may deviate from a quadrangle due to the accumulated transformation errors. We warp this panoramic court to a quadrangle in a content preserving manner. Because uniform sampling in a quadrangle is easier than that in the panoramic court, we adopt an inverse strategy, which warps the quadrangle to fit the panoramic basketball court while minimizing the shape distortion of each local region. Once the warped mesh is obtained, the

1. http://opencv.org/

Fig. 5. Left to right: video frames captured in the left, middle, and right viewpoints. The slope of the top court boundary changes when the camera pans. We apply this characteristic to determine the viewpoint similarity between two video frames before tracking KLT features and computing the homography transformation.

inverse warping of each local region can be used to warp the panoramic court to a quadrangular one. Figure 2 (f) and (g) show the original and warped meshes used to calibrate the basketball court.

We represent the quadrangular court using a quad mesh, in which each quad roughly contains 20×20 pixels. Let $\mathbf{M} = \{\mathbf{V}, \mathbf{F}\}$ be the quad mesh, where $\mathbf{V} = \{\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_n\}$, n is the total number of vertices, $\mathbf{v} \in \mathcal{R}^2$ is the vertex position, and $\mathbf{F}$ is the set of quad faces. We iteratively move the boundary vertices $\partial\mathbf{V}$ toward their closest court pixels while preventing the shape of each quad from distortion. Specifically, two energy terms D_b and D_s are formulated according to the constraints and the objective function is minimized to obtain the warped vertex positions. To approximate the panoramic court, we present the energy term

$$D_b = \sum_{i \in \partial V} \left| \hat{\mathbf{v}}_i - \mathbf{u}_i \right|^2, \tag{1}$$

where $\hat{\mathbf{v}} \in \hat{\mathbf{V}}$ denotes the warped vertex position and $\mathbf{u}$ is the court pixel closest to $\hat{\mathbf{v}}$. To retain the quad shape, each quad is enforced to undergo a similarity transformation. That is,

$$\begin{bmatrix} s & r \\ -r & s \end{bmatrix} \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix} + \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}_x \\ \hat{\mathbf{v}}_y \end{bmatrix}. \tag{2}$$

Let $f_0 - f_3$ be the four vertices of quad f and $[s_f, r_f, u_f, v_f]^T$ be the unknown similarity transformation. The constraint is given by

$$\begin{bmatrix} \mathbf{v}_{f_0,x} & \mathbf{v}_{f_0,y} & 1 & 0 \\ \mathbf{v}_{f_0,y} & -\mathbf{v}_{f_0,x} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_{f_3,x} & \mathbf{v}_{f_3,y} & 1 & 0 \\ \mathbf{v}_{f_3,y} & -\mathbf{v}_{f_3,x} & 0 & 1 \end{bmatrix} \begin{bmatrix} s_f \\ r_f \\ u_f \\ v_f \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{v}}_{f_0,x} \\ \hat{\mathbf{v}}_{f_0,y} \\ \vdots \\ \hat{\mathbf{v}}_{f_3,x} \\ \hat{\mathbf{v}}_{f_3,y} \end{bmatrix} \tag{3}$$

and briefly represented by $\mathbf{A}_f\mathbf{S}_f = \mathbf{V}_f$. Based on the conformal mapping theory, the unknown similarity transformation $\mathbf{S}_f$ can be eliminated and the expression $(\mathbf{A}_f(\mathbf{A}_f^T\mathbf{A}_f)^{-1}\mathbf{A}_f^T - \mathbf{I})\mathbf{V}_f = 0$ is obtained. Therefore, we introduce the term

$$D_s = \sum_{f \in \mathbf{F}} \left| (\mathbf{A}_f(\mathbf{A}_f^T\mathbf{A}_f)^{-1}\mathbf{A}_f^T - \mathbf{I})\mathbf{V}_f \right|^2. \tag{4}$$

Note that the only unknown variable in this expression is $\mathbf{V}_f$. The term D_s is quadratic.

We integrate the weighted energy terms to form the objective function $\omega D_b + D_s$. Given that spectators usually appear around the boundaries of a court, in which some court pixels are not detected, drastically enforcing each boundary vertex to locate at the position of the closest court pixel would introduce serious artifacts. Therefore, we set $\omega = 0.2$ in our experiments. Apparently, the objective function $\omega D_b + D_s$ is nonlinear because the unknown variables $\mathbf{v}$ and $\mathbf{u}$ are correlated. Therefore, we begin the minimization by setting $\mathbf{u}_i$ to the court pixel position closest to $\mathbf{v}_i$. In this scenario, the objective function becomes linear and the warped vertex positions $\hat{\mathbf{V}}$ can be obtained by solving a linear system. Once the new $\hat{\mathbf{V}}$ is computed, we refine each closest court pixel $\mathbf{u}$. Our system alternatively computes the two sets of unknown variables until the solution converges or the energy $\omega D_b + D_s$ increases. We refer the readers to [29] for more details on our optimization.

3.1.4 *Homography calibration*

We warp the quadrangular basketball court to a rectangle using a homography to remove the perspective effect. The dimension of this rectangular court is 28.64 by 15.24 defined by International Basketball Federation (FIBA). The correspondence of four corners between the two courts can be obtained according to their positions. Accordingly, the homography $\mathbf{H}$ is computed from $\mathbf{q} = \mathbf{H}\mathbf{p}$, where $\mathbf{q}$ and $\mathbf{p}$ denote the corner positions on the rectangular and the quadrangular courts, respectively. Once the court is rectified, we successfully calibrate camera motions.

3.2 Calibration in incomplete court videos

Our camera calibration method works only when the video contains a complete basketball court. For the remaining videos, the calibration requires an additional homography to a video that covers the whole court. Let $\mathbf{W} = \{w_0, w_1, ...\}$ and $\mathbf{Z} = \{z_0, z_1, ...\}$ denote the videos covering the whole and half basketball courts, respectively. To calibrate frame z_i, our goal is to compute a homography that maps z_i to w_j based

on the tracked KLT features, (Section 3.1.1). However, given that frames z_i and w_j may be captured from different viewpoints and have few objects in common, the computed homography is not reliable or even invalid. This problem can be solved by stochastically trying all frames in $\mathbf{W}$. The system then uses the transformation computed according to the largest amount of corresponding KLT features. Given that this approach suffers from heavy computational cost, we alternatively determine frame w_j based on the court shape to speed up the matching process.

We observe that the camera is fixed at a certain position and performs only pan and zoom in court view shot videos. Under these circumstances, the slope of the top court boundary implies the viewpoint of a frame, as indicated in Figure 5. Therefore, we apply the convex hull that represents the court region in a frame (Section 3) and extract the top court boundary that is connected to the vertical edge. Frame w_j, in which the slope of the top court boundary is closest to that of frame h_i, is used to estimate the homography for calibrating frame h_i.

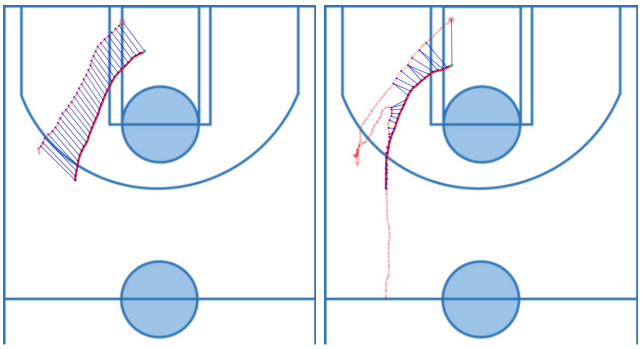


Fig. 6. User specified and retrieved player trajectories are rendered in complete and half saturated red, respectively. (Left) Blue points are uniformly sampled from the *end point* along each trajectory and the deviations (blue lines) of corresponding points are averaged to measure similarity. (Right) Blue points are uniformly sampled along the user specified stroke but correspond to the closest points on the extracted player trajectory. Dissimilar trajectory is obtained due to the wrong point mapping.

4 Application: Video Retrieval

We present a stroke-based interface of retrieving basketball videos to demonstrate the feasibility of the presented camera calibration method (Figure 7). This objective is achieved by first extracting player trajectories from basketball videos. The trajectories are then rectified to the standard basketball court. When stroke queries are provided by users, our system compares the similarity of the strokes and the rectified player trajectories in the database. It then returns the results (i.e., up to three in our default setting) that best fit the query and renders the player trajectories on the basketball court for illustration.

4.1 Preprocessing

The first step of this stroke-based retrieval is to extract player trajectories from broadcast basketball videos. Broadcast videos usually contain commercials, breaks, and close-up views for highlights, in which obtaining spatial information is difficult. Therefore, we extract player trajectories from court view shot videos. Our system partitions a broadcast video into short video clips, in which each clip contains only one shot, using the scene change detection algorithm presented by Hanjalic et al. [30]. We then apply the support vector machine to train a classifier, based on the color histogram of each video frame, to detect court view and non-court view shot videos. Specifically, our system transforms each pixel color to *YCrCb* color space and computes a *CrCb* color histogram with 32×32 bins to represent the feature of a video frame. Channel *Y* is excluded to resist illumination changes. In our experiment, we use 235 court view and 162 non-court view shot frames to train the classifier. This classifier is then adopted to test another 216 videos and obtain the correct rate of 96.76%. This simple approach performs well because the backgrounds of court view and non-court view shots are very different in broadcast basketball videos.

Given by a court view shot video, we apply the deformable part model [26] to detect player positions in each frame and track their motions to extract player trajectories [9]. To enable a precise query for basketball video retrieval, players who assist and score should be identified. This objective could be achieved by either player identification [9] or ball tracking [11]. But we build the information by manual annotation in our current system.

4.2 Stroke-player trajectory comparison

We estimate the trajectory similarity using Andrienko et al.'s method [31] to determine whether the video fits user's requirement. Denote by *start point* and *end point* the two ends of a trajectory, in which the player runs from the *start point* toward the *end point*. Our system uniformly samples corresponding points from the two *end points* along the stroke and rectified player trajectory, respectively. The sampling stops whenever it reaches the *start point* of the shorter curve. Figure 6 (left) illustrates our sampling approach. Once the corresponding points are sampled, our system averages the L2-norm deviations of the corresponding points to determine trajectory similarity. The failed example in Figure 6 (right) explains why the closest points on the extracted trajectory are not used. Besides the robustness of similarity measure, this partial comparison also allows users to control the degree of strictness during retrieval. Namely, drawing a short

11:19		0-3	M. Chalmers makes 3-pt from 25 ft (assist by M. Miller)
11:16	T. Young misses 2-pt shot from 1 ft	0-3	
11:05		0-3	Defensive rebound by U. Haslem
10:54		0-3	L. James makes 2-pt shot from 16 ft

Fig. 8. Example of *play-by-play* text; The columns from left to right indicate game time, events of team one, scores, and the events of team two, respectively.

stroke indicates where the player gets scores whereas drawing a long stroke specifies how the player moves in a game.

4.3 Play-by-play text

In addition to player trajectories, we employ the play-by-play text to enhance the semantics of a query. This play-by-play text is available on-line and records important events in a game such as shot made, shot miss, assists, fast break, ..., etc. (see Figure 8). Given that the event time is provided in a play-by-play text and the game time is depicted in each video frame, we obtain the event of a video clip by time matching. Given that the game time appears in bitmap format, we apply optical character recognition [32] to extract the information. The recognition has nearly perfect accuracy because the game time is fixed at a certain region and its background is homogeneous. As a result, in addition to the strokes used to specify player trajectories, our system allows users to indicate the events to enhance retrieval correctness.

4.4 Video retrieval interface

To retrieve basketball videos, users can draw one or two strokes to specify the moving trajectories of the players who shoot and assist. Red and green strokes indicate the scorer and assister, respectively. Users can also set the events such as shot made, shot miss, assist, and put back to further stipulate the videos they are looking for. The query combined with strokes and events clearly define each video clip to prevent ambiguity. The strokes are drawn on a half court because most tactical events activate at the positions close to a basket and teams have to switch sides after half of the game.

We describe how each kind of videos is retrieved based on the stroke queries and the events in details.

Cut-in. A player moves to the basket and then lays up or dunks to scores. This kind of video is retrieved by drawing the stroke toward and ends close to the basket. Users can also check the assist event during retrieval, in which the ball is passed from a player before he/she scores.

Cut-out. A player moves out to locate a wide open area and receives the ball from a teammate to shoot. This kind of video is retrieved by drawing the stroke away from the basket. The assist event is checked in this category because the scorer does not generally handle the ball in the beginning. Remarkably few players dribble the ball, move out, and shoot because passing the ball to one's teammate is always the better choice.

Fade-away shot. A player initially moves toward or around the basket but jumps back to acquire some space for shooting. This kind of video is retrieved by drawing the stroke with a small portion away from the basket at the end. Given that the scorer locates an open space for himself/herself to make a shoot, the assist event in this category is not checked.

Fast break. A player moves toward the basket and less than two or even no opposing player is in front of him/her to defend the attack. This kind of video is retrieved by drawing a long straight stroke that begins from the other side of the court. The straightness of the stroke signifies that tactics are not necessary and event periods are short.

Put back. A player grabs the offensive rebound and immediately scores. This kind of video is retrieved by generally drawing the stroke toward the basket with the put back event checked. This event is detected when the score and rebound are simultaneously denoted in the play-by-play text.

Pick-and-roll tactic. A player first sets a screen (pick) for his/her teammate who handles the ball, slips behind the defender (roll) to receive the pass, and then cuts in to make a score. In this tactic, the goal of the assister is to draw attention from the opposing players and let the scorer locate a space to shoot. Therefore, two strokes are required to retrieve this kind of video; one for the assister and one for the scorer. In other words, only the videos in which both player trajectories are matched will be retrieved. This kind of video is retrieved by drawing two strokes, in which a portion of them are parallel, close to each other, and head toward the opposite directions, because the open space immediately emerges when the defenders move in a phase similar to that of the assister.

5 Results and Discussions

We have implemented the presented system and run the code on a desktop PC with Core i7 3.0 GHz CPU. Broadcast basketball videos with 1280×720 and 720×480 resolutions are used in our experiments. Generally, calibrating a video frame by using our unoptimized code required more or less 0.2 seconds, where most of the time were taken in KLT feature tracking. Given that the calibration of each frame requires tracked KLT feature, the cost prevents our system achieving interactive performance. This problem could be greatly reduced by leveraging the GPU,

Fig. 7. Top to bottom: Retrieved basketball videos of fast break, fade away shot, and pick-and-roll tactics. The leftmost column shows the specified strokes and events. Red and green rectangles specify the scorer and assister; complete and half saturations denote the user specified strokes and retrieved player trajectories, respectively. The remaining columns on the right display the key frames of the retrieved video. Similarly, the red and green rectangles indicate the scorer and assister in the frames. Note that the length of a stroke query involves with query strictness. The top example shows the strict query, whereas the middle example shows the loose one.

yet we did not implement the part in our system. The homography transformation from a video coordinate to a panoramic court coordinate is efficient because of few unknown variables in each frame. The calibration from a panoramic court to a standard court is also fast because linear interpolation and matrix multiplication are used. As for the video retrieval application, given that player trajectories are pre-calibrated and stored in the database, our system achieves interactive performance when measuring the similarity of trajectories.

5.1 Experimental results

We tested a number of broadcast basketball videos, which contain zoom-in, zoom-out, fast camera panning, and flashes, to demonstrate the feasibility of our technique. Our system does not require camera intrinsic parameters because only homographies, court warping, and court rectification are applied to calibrate video frames. Figure 9 and our accompanying video show the results. To visualize the calibration quality, we place the standard basketball court at a fixed position and calibrate each video frame to overlay the court. The calibrated frames are rendered with half transparency to exhibit whether court features are properly aligned. As can be seen, the court features (i.e., boundary line, free throw line, and three-point lines) of each video frame are well aligned with the standard basketball court, which indicates that the rectified player positions will be accurate enough for basketball video retrieval. Moreover, the calibration between consecutive frames is temporally coherent and thus, the rectified player trajectories do not suffer from jittering artifacts.

We show the basketball videos retrieved by using our system in Figure 7. Users draw strokes and specify events to achieve the objective. Our system subsequently returns the results in which the trajectories of the scorer or assister best fit the query. Because the retrieval interaction cannot be appreciated from still images, we refer readers to our accompanying video.

5.2 Evaluations

Comparison. To demonstrate the effectiveness of our camera calibration, we mainly compare the presented system to Hu et al.'s [6] approach because both methods are fully automatic and attempt to handle broadcast basketball videos. All their results shown in this paper were provided by the authors. Similarly, we place the standard basketball court at a fixed position and render their rectified video frames with half transparency to exhibit the calibration quality. As shown in Figure 9 and our accompanying video, the approach of [6] cannot handle video frames that cover insufficient court line features. Statistically, in this example, only 18% (101 / 565) of video frames were successfully calibrated. Their calibration also suffers from temporal incoherence in consecutive frames because features in each frame are detected individually.

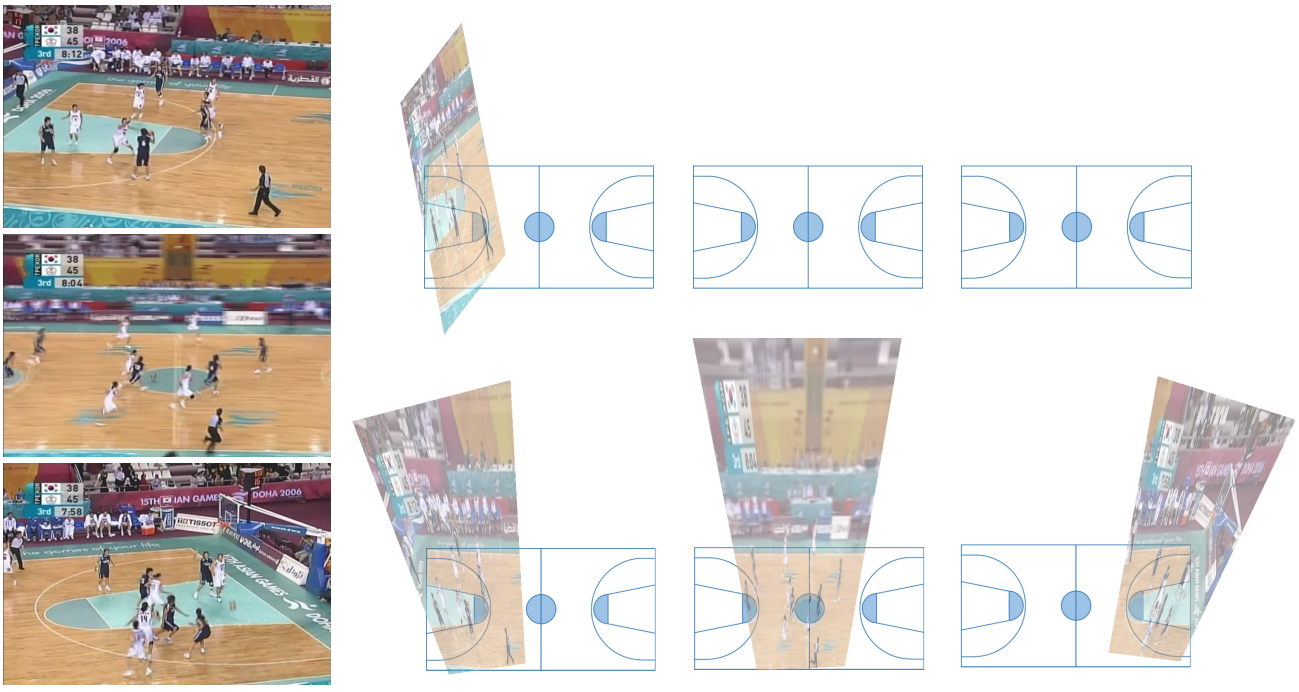


Fig. 9. (Left) Original video frames. (Top right) Calibrated video frames achieved by [6]. (Bottom right) Calibrated video frames achieved by our method. The images indicate that the method of [6] cannot calibrate both middle and right court views because of the undetected court line features. In contrast, our system can calibrate all video frames due to court reconstruction.

In contrast, 100% of video frames are calibrated by using our system, and these calibrated frames are temporally coherent. Accordingly, only our system can rectify player trajectories that are good enough to achieve basketball video retrieval. Figure 10 shows the player trajectories rectified by Hu et al.'s method [6], our system, and by a user. These trajectories indicate that our result is close to the one perceived by humans.

Our system is superior to that of [6] due to the consideration of all video frames. In other words, Hu et al.'s [6] method directly maps the detected features in a frame to the standard basketball court. It fails whenever court features in a frame are not detected. In contrast, we track KLT features to build the connection between video frames and reconstruct the basketball court. The calibration is actually achieved based on the four corners of the reconstructed court. As a result, while court features in a frame are occluded, the calibration from that frame to the standard court can still be achieved via other frames in the video. These conditions verify that our camera calibration is robust.

Basketball video retrieval. Our stroke-based system allows users to retrieve basketball videos by specifying player trajectories. This additional information helps separate basketball videos that have the same pre-defined event. For the examples shown in Figure 11, the players cut in from different positions and make scores. Retrieving a single video by using a conventional text-based query system is

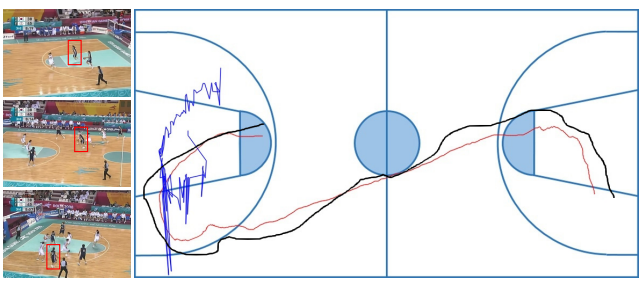


Fig. 10. (Left) Key frames of a basketball video. The goal in this example is to rectify the moving trajectory of the player indicated by a red rectangle. (Right) The trajectories rectified by using Hu et al.'s method [6] (blue), our system (red), and by a user (black). As indicated, the player trajectory rectified by our system is complete, temporal coherent, and close to the trajectory perceived by humans.

difficult because both events are identical. In addition, our basketball video retrieval supports the partial matching of a player trajectory and a given stroke. This partial matching is helpful because users usually have no idea when a trajectory begins and ends in a video and cannot provide an exact stroke query. Another advantage of this partial matching is the control of strictness. Considering that our similarity measure compares the stroke query to only a segment of player trajectory, stroke length involves query strictness. Users can apply a short stroke to indicate

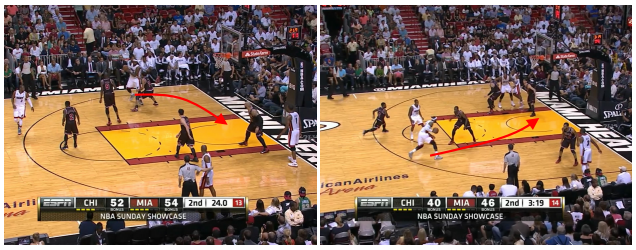


Fig. 11. Players normally cut in from different positions. The presented stroke query helps separate the videos even though they have the same pre-defined event.

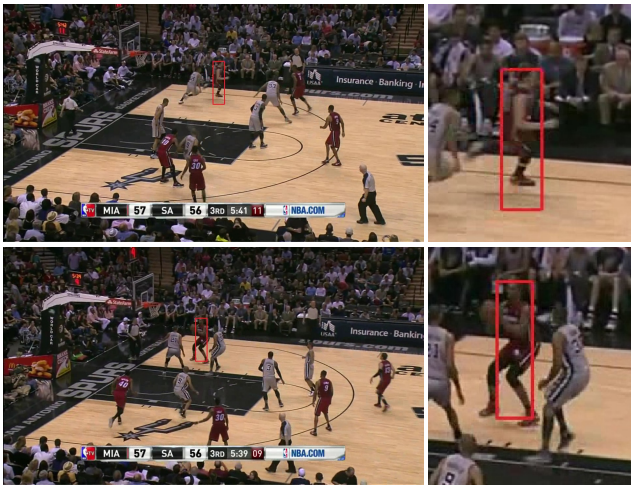


Fig. 12. Retrieved video may be unexpected due to the tracking error. The tracked players of *a single trajectory* are indicated using rectangles. The two frames of a court view shot are shown in the left and the corresponding zoom-in views are highlighted in the right.

where the player makes scores and apply a long one to specify how the player cuts into the restricted area. We show examples in Figure 7.

We have presented our system to college basketball players. They indicated that our video retrieval system is beneficial to basketball coaches and players. That is, most coaches draw curves on a tactical board to convey tactics, but such curves are too abstract to players. With the help of our system, coaches can draw strokes to retrieve the video that shows a real example completed by professional players to provide better explanation. Besides, players can experience by real example how the opposing team may react for tactics. Overall, the players expressed high preference to our system because strokes are sufficient to define a tactical event in most basketball videos. In particular, they agree that our system is cost effective to find the demanded results.

5.3 Limitations

Our camera calibration technique relies on the basketball court reconstruction. It works only for court view shot videos due to sufficient background features for scene matching. For the videos with a close-up view, in which players normally occupy a large area of the frame, our system cannot obtain corresponding background features and fails in this circumstance. In addition, our camera calibration is based on the four corners of a reconstructed basketball court. Court lines, such as free throw lines, three point lines, and center circle, in a video are not considered and may not perfectly aligned with the real court features after calibration. Given that accumulation errors are inevitable, the calibration is visually good but not geometrically correct. We plan to adopt these features to enhance the calibration quality in the near future.

Our system does not guarantee the retrieved basketball videos to always fit the query due to imperfect player tracking techniques. Figure 12 and the accompanying video show a failure example. Besides, DPM detection only outputs bounding boxes, which are usually not accurate in telling the exact player positions. The calibrated player trajectories could suffer from noise. Finally, without tracking ball trajectories, our system is not sufficient to retrieve videos containing complex ball passing.

6 Conclusions

We have presented a fully automatic method of calibrating camera motions in basketball videos. The system is robust to videos that contain zoom-in, zoom-out, fast camera panning, and flashes. The success of this approach comes from the consideration of all frames in a court view shot video rather than each frame individually. Therefore, when certain court features are not available in some frames due to player occlusions and illumination changes, they can still be obtained from other frames for calibration. Although our system is presented to calibrate camera motions in basketball videos, it has the potential to handle many other sport videos. Our supplemental result shows a volleyball example.

This calibration technique has been applied to retrieve basketball videos by giving stroke and event queries. Coaches, players, and spectators are expected to benefit from our technique because the system supports precise queries to prevent unwanted results during retrieval. Our camera calibration is also useful in many applications, including tactical analysis, wide open detection, and player statistical data visualization, because player positions are transformed to the same coordinate system. Accordingly, we intend to discover more interesting applications based on this technique in future.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. We are also grateful to Gerardo

Figueroa for narrating the demo video and the National Basketball Association Entertainment for kindly providing us the dataset. This work was supported in part by the National Science Council (101-2628-E-009-020-MY3, 100-2628-E-006-031-MY3, and 100-2221-E-006-188-MY3).

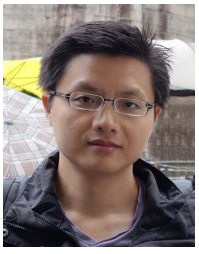
REFERENCES

[1] M. Varga, *Practical Image Processing and Computer Vision*. John Wiley & Sons Australia, Limited, 2008.
[2] D. Farin, S. Krabbe, P. H. N. de With, and W. Effelsberg, "Robust camera calibration for sport videos using court models." in *Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, 2004, pp. 80–91.
[3] D. Farin, J. Han, and P. de With, "Fast camera calibration for the analysis of sport sequences," in *IEEE International Conference on Multimedia and Expo.*, 2005, pp. 482–485.
[4] J. Han, D. Farin, and P. H. N. de With, "A real-time augmented-reality system for sports broadcast video enhancement," in *International Conference on Multimedia*, 2007, pp. 337–340.
[5] X. Yu, N. Jiang, L.-F. Cheong, H. W. Leong, and X. Yan, "Automatic camera calibration of broadcast tennis video with applications to 3d virtual content insertion and ball detection and tracking," *Comput. Vis. Image Underst.*, vol. 113, no. 5, pp. 643–652, 2009.
[6] M.-C. Hu, M.-H. Chang, J.-L. Wu, and L. Chi, "Robust camera calibration and player tracking in broadcast basketball video." *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp. 266–279, 2011.
[7] P. Carr, Y. Sheikh, and I. Matthews, "Point-less calibration: Camera parameters from gradient-based alignment to edge images," in *IEEE Workshop on the Applications of Computer Vision*, 2012, pp. 377–384.
[8] J. Puwein, R. Ziegler, L. Ballan, and M. Pollefeys, "Ptz camera network calibration from moving people in sports broadcasts." in *IEEE Winter conference on Applications of Computer Vision*, 2012, pp. 25–32.
[9] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, "Learning to track and identify players from broadcast sports videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1704–1716, 2013.
[10] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593 – 600.
[11] 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. Comun. Image Represent.*, vol. 20, no. 3, pp. 204–216, 2009.
[12] P. Parisot and C. De Vleeschouwer, "Graph-based filtering of ballistic trajectory," in *IEEE International Conference on Multimedia and Expo.*, 2011, pp. 1–4.
[13] K. Kumar, P. Parisot, and C. De Vleeschouwer, "Demo: Spatio-temporal template matching for ball detection," in *ACM/IEEE International Conference on Distributed Smart Cameras*, 2011, pp. 1–2.
[14] C. Verleysen and C. D. Vleeschouwer, "Recognition of sport players' numbers using fast-color segmentation," in *IS&T/SPIE | Electronic Imaging*, 2012.
[15] Y. Liu, X. Liu, and C. Huang, "A new method for shot identification in basketball video," *Journal of Software*, vol. 6, no. 8, pp. 1468–1475, 2011.
[16] F. Chen and C. De Vleeschouwer, "Automatic production of personalized basketball video summaries from multi-sensored data." in *International Conference on Image Processing*, 2010, pp. 565–568.
[17] ——, "Personalized production of basketball videos from multi-sensored data under limited display resolution," *Comput. Vis. Image Underst.*, vol. 114, no. 6, pp. 667–680, 2010.
[18] K. Okuma, J. J. Little, and D. G. Lowe, "Automatic rectification of long image sequences." in *Asian Conference on Computer Vision*, 2002.
[19] R. Hess and A. Fern, "Improved video registration using non-distinctive local image features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
[20] A. Gupta, J. J. Little, and R. J. Woodham, "Using line and ellipse features for rectification of broadcast hockey video," in *Canadian Conference on Computer and Robot Vision*, 2011, pp. 32–39.
[21] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval." *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 56–63, 1999.
[22] C.-W. Su, H.-Y. M. Liao, H.-R. Tyan, C.-W. Lin, D.-Y. Chen, and K.-C. Fan, "Motion flow-based video retrieval." *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1193–1201, 2007.
[23] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *ACM international conference on Image and video retrieval*, 2007, pp. 494–501.
[24] T. Urruty, F. Hopfgartner, D. Hannah, D. Elliott, and J. M. Jose, "Supporting aspect-based video browsing: analysis of a user study." in *ACM International Conference on Image and Video Retrieval*, 2009.
[25] W. Hu, N. Xie, Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *Trans. Sys. Man Cyber Part C*, vol. 41, no. 6, pp. 797–819, 2011.
[26] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[27] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2427–2434.
[28] J. Shlens, "A tutorial on principal component analysis," in *Systems Neurobiology Laboratory, Salk Institute for Biological Studies*, 2005.
[29] K. Madsen, H. B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems (2nd ed.)," p. 60, 2004.
[30] A. Hanjalic, "Shot-boundary detection: unraveled and resolved?" *IEEE Trans. Circuits Syst. Video Techn.*, vol. 12, no. 2, pp. 90–105, 2002.
[31] G. Andrienko, N. Andrienko, and S. Wrobel, "Visual analytics tools for analysis of movement data," *SIGKDD Explor. Newsl.*, vol. 9, pp. 38–46, 2007.
[32] Tesseract-ocr. [Online]. Available: http://code.google.com/p/tesseract-ocr/

Pei-Chih Wen received the BS and MS degrees from the Department of Computer Science, National Chiao Tung University, Taiwan, in 2011 and 2013, respectively. His research interests include computer graphics and computer vision.

Wei-Chih Cheng received the BS degree from the Department of Computer Science, National Taiwan Normal University, Taiwan, in 2013. He is currently pursuing the master's degree in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include computer graphics and computer vision.

Yu-Shuen Wang received the BS and PhD degrees from the Department of Computer Science and Information Engineering, National Cheng-Kung University, in 2004 and 2010, respectively. He is currently an assistant professor of the Department of Computer Science at National Chiao Tung University (http://people.cs.nctu.edu.tw/ yushuen/). He leads the Computer Graphics and Visualization Lab at the Institute of Multimedia Engineering. His research interests include computer graphics, computational photography, and visualization.



Hung-Kuo Chu is an assistant professor at the Department of Computer Science, National Tsing Hua University. He received BS and Ph.D degrees from Department of Computer Science and Information Engineering, National Cheng-Kung University. His research interests focus on Shape Understanding, Smart Manipulation, Perception-based Rendering, Recreational Graphics and Human Computer Interaction.

Nick C. Tang received the B.S., M.S., and Ph.D. degrees from Tamkang University, Taipei, Taiwan, in 2003, 2005, and 2008, respectively. He is currently a Post-Doctoral Fellow with the Institute of Information Science, Academia Sinica, Taipei. His research interests include image and video analysis, computer vision, computer graphics, and their applications.

Hong-Yuan Mark Liao received the Ph.D. degree in electrical engineering from Northwestern University, Evanston, IL, USA, in 1990. In 1991, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Distinguished Research Fellow. He has worked in the fields of multimedia signal processing, image processing, computer vision, pattern recognition, video forensics, and multimedia protection for more than 25 years. He was a recipient of the Young Investigators Award from Academia Sinica in 1998, the Distinguished Research Award from the National Science Council of Taiwan, in 2003, 2010, and 2013, respectively, the National Invention Award of Taiwan in 2004, the Distinguished Scholar Research Project Award from the National Science Council of Taiwan in 2008, and the Academia Sinica Investigator Award in 2010. His professional activities include the Co-Chair of the 2004 International Conference on Multimedia and Exposition (ICME), the Technical Co-Chair of the 2007 ICME, the General Co-Chair of the 17th International Conference on Multimedia Modeling, the President of the Image Processing and Pattern Recognition Society of Taiwan (20062008), an Editorial Board Member of the IEEE SIGNAL PROCESSING MAGAZINE (20102013), and an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING (20092013), the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (20092012), and the IEEE TRANSACTIONS ON MULTIMEDIA (19982001). He also serves as the IEEE Signal Processing Society Region 10 Director (Asia-Pacific Region).