Mimicking Human Camera Operators

Jianhui Chen[1,2]
[1]University of British Columbia
jhchen14@cs.ubc.ca

Peter Carr[2]
[2]Disney Research
peter.carr@disneyresearch.com

Abstract

Filming team sports is challenging because there are many points of interest which are constantly changing. Unlike previous automatic broadcasting solutions, we propose a data-driven approach for determining where a robotic pan-tilt-zoom (PTZ) camera should look. Without using any pre-defined heuristics, we learn the relationship between player locations and corresponding camera configurations by crafting features which can be derived from noisy player tracking data, and employ a new calibration algorithm to estimate the pan-tilt-zoom configuration of a human operated broadcast camera at each video frame. Using this data, we train a regressor to predict the appropriate pan angle for new noisy input tracking data. We demonstrate our system on a high school basketball game. Our experiments show how our data-driven planning approach achieves superior performance to a state-of-the-art algorithm and does indeed mimic a human operator.

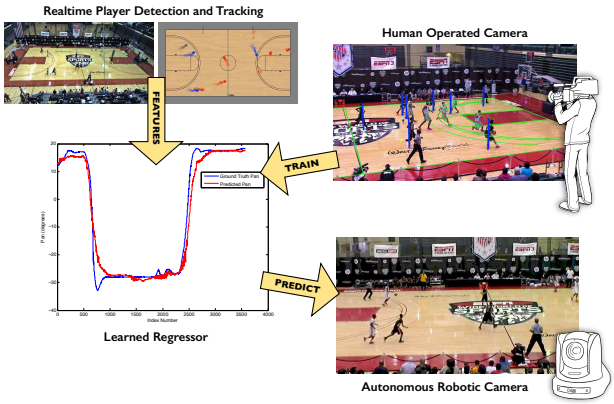


Figure 1. **Mimicking**. Using a stationary machine vision camera, we extract a fixed length feature vector from tracking data to describe the game state. Meanwhile, we estimate the pan-tilt-zoom configuration for each video frame of a human operated camera. Using these data, we train a regressor to model the relationship between camera pan angle and game state. At test time, we use the regressor to generate target pan angles for an autonomous robotic camera based on the input tracking data.

1. Introduction

Automatic broadcasting makes small events, such as lectures and amateur sporting competitions, available to a much larger audience. These systems must be able to sense the environment, decide where the cameras should look, and ensure the cameras remain fixated on the intended targets [7]. In this paper, we focus on planning where the cameras should look. Most existing autonomous camera systems function in an object tracking paradigm (such as following the lecturer) and implement camera planning as smoothing noisy tracking data. In practice, the resulting output video often looks robotic. Human operators, in contrast, are trained to anticipate action and to frame their shots with sufficient 'lead room' [18]. Motivated by this distinction, we formulate camera planning as a supervised learning problem: given sensor data of the environment and corresponding exemplar camera work by a human expert, we learn a regressor which can predict an appropriate camera configuration for a new situation. Basketball is a good scenario for a proof of concept because the current situation can be precisely described (to a large extent) by the positions of the players. Furthermore, the repetitive nature of the game ensures a representative training set can be collected efficiently.

Generally, the camera configuration information is not directly available from the recorded video. Instead, we must estimate the pan, tilt and zoom settings of the camera at each frame from recorded video. Previous approaches [10, 17] have estimated camera parameters by searching the images for known geometric features. Alternative methods have looked for matching key points between the image and a set of reference frames [11]. In both cases, robustly estimating the parameters of a PTZ camera for long video sequences has not been addressed.

Long-term multi-object tracking remains an unsolved problem in computer vision, especially within a realtime constraint. As a result, any practical planning algorithm for autonomous camera systems must be robust to tracking errors, such as coping with a varying number of objects.

Therefore, a robust feature extraction phase which can mitigate the effects of noise is critical for successful camera planning.

Many machine learning methods (such as SVM, neural networks and KNN classifiers) have been used during feature extraction in previous camera planning algorithms [7]. However, to the best of our knowledge, no one has formulated camera planning as a supervised regression problem (and certainly not for automatically broadcasting sporting events). Furthermore, we have proposed a novel camera model to estimate pan-tilt-zoom settings at every video frame, as well as a host of new features which can be derived from noisy tracking data. In summary, our paper has three main contributions:

- implementing camera planning by solving a supervised regression problem,
- estimating the per-frame pan-tilt-zoom settings of a PTZ camera via video analysis,
- extracting robust features from noisy multi-object tracking data.

We first summarize recent work in Section 2. We then explain in Section 3 how camera planning can be posed as a regression problem. In Section 4, we investigate the performance of different regression algorithms and feature combinations. Finally, we validate our method in Section 5 by comparing to a baseline automatic planning algorithm, as well as the actual camera work performed by the human operator.

2. Related Work

Automatic Broadcasting As illustrated in a recent survey [7], several previous automatic sports broadcasting systems operate in an offline manner by cropping subregions from recorded video after tracking players and/or the ball. For example, Chen *et al*. [6] determined important subregions by considering user-defined attentional interests (such as including star players). In contrast, Daigo *et al*. [8] developed an online system which controlled a robotic camera by tracking audience face directions and a rough region of where players were located on a basketball court.

Sports Scene Understanding Sports scene understanding refers to how the current game situation is described in an efficient manner. Typically, it consists of the locations of players and the ball, such as in [2]. However, higher-level understanding is also possible. Kim *et al*. [12] proposed a global motion vector field on the ground plane to predict the motion of the broadcast camera. Generally, it mimicked the human operators by setting heuristic rules and learning from recorded video.

Video Calibration In most sports, the camera parameters are estimated by determining the homography $\mathtt{H}$ between the image plane and the ground plane. For example, Gupta *et al*. [10] leveraged ellipses along with lines and points to estimate $\mathtt{H}$. The combination of geometry, appearance and motion information enabled them to track long sequences of broadcast video in hockey. To overcome the lack of distinct image features in football, Hess *et al*. [11] proposed non-distinctive local features which increased the number of matched patches by limiting the possible matchings. The pan, tilt and zoom parameters can be estimated from homographies [1].

3. Method

We model camera planning as a structured regression problem (see Fig. 1)

$$\hat{\mathbf{y}}_t = h(\mathbf{x}_t) \tag{1}$$

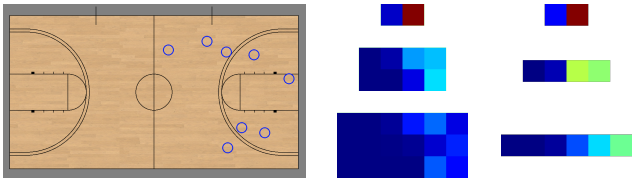
where $\hat{\mathbf{y}}_t$ is the planned pan-tilt-zoom state of the camera for a particular time, $h(\cdot)$ is the learned regressor, and $\mathbf{x}_t$ is a feature vector extracted from the current tracking data (see Sec. 3.1). To learn the regressor, the work of an expert human camera operator is analyzed (see Sec. 3.2) to generate exemplar pan-tilt-zoom states $\{\mathbf{y}_t\}$ for the observed tracking features $\{\mathbf{x}_t\}$. Using these paired data $\{(\mathbf{y}_t, \mathbf{x}_t)\}$, we investigate various machine learning algorithms (see Sec. 4) to generate a suitable camera planning algorithm $h(\cdot)$.

Our experimental setup is from a high school basketball match recorded by two cameras. The broadcast camera (1280×720 at 60fps) is operated by a human expert. The machine vision camera (1936×1456 at 25fps) remains stationary so that a computer can detect and track players automatically. We manually segmented the recorded video to remove all non-continuous periods of play (such as timeouts and free throws). The resulting dataset is about 17 minutes in duration (16 of which were used for training, and 1 held out for testing). For convenience, we linearly interpolated the tracking data to 60fps. Because the main broadcast camera in basketball maintains a wide shot and mostly constant tilt angle, we focus on predicting an appropriate pan angle, leaving tilt and zoom as future work (which may be more important in sports such as soccer).

3.1. Features

We detect players within the video of the stationary machine vision camera using a method similar to [4], which analyzes background subtraction results in terms of 3D cylinders. To minimize the impact of missed and false detections, we analyze the data in $\tau = 12$ frame chunks (≈ 0.5s), and greedily fit constant velocity models to the detection data using RANSAC [13]. As a result, temporal chunk t contains a set $\mathcal{T}_t = \{T_1, T_2, ...T_{N_t}\}$ of short constant velocity trajectories. Unlike [6], we do not have an estimate of the players' identities.

(a)



(b) (c) (d)

Figure 2. **Feature Extraction**. Players are detected in a stationary camera (a) with corresponding world locations (b). The heat map feature (c) is computed by quantizing the basketball court into 1×2, 2×4 and 3×6 grids, and soft binning each detected player location. The spherical map (d) is computed by projecting each (x, y) player location onto a unit sphere located at the broadcast camera's center of projection, and soft binning the pan angle into three different quantization resolutions.

In order to formulate camera planning as a regression problem, we must extract a fixed length feature vector $\mathbf{x}_t$ from each set $\mathcal{T}_t$ of noisy player trajectories (see Fig. 2). Here, we proposed three possible features.

Centroid We compute a 2-dimensional feature vector $\mathbf{x}_t^{\text{centroid}}$ by computing the average (x, y) location of all players during temporal chunk t.

Heat Map Similar to HOG and SIFT, we divide the basketball court into a 2D grid and count the number of players present within each cell to generate $\mathbf{x}_t^{\text{heat map}}$. To minimize quantization effects, we linearly interpolate each player's count between the four neighboring cells. Additionally, the heat map describes the player distribution at different scales by changing the resolution of grid. In practice, we employ three resolutions: 2×1, 4×2 and 6×3, resulting in a 28-dimension feature vector.

Spherical Map Because the regressor predicts a pan angle for the PTZ camera, there is an inherent non-linear spherical projection between the world coordinate system, and the pan-tilt-zoom domain of the camera. Therefore, we generate an equivalent heat map $\mathbf{x}_t^{\text{spherical map}}$ on the unit sphere of the PTZ camera. There are two key differences: (1) we project each player location onto the unit sphere, and (2) since we are only interested in predicting pan angle, we only quantize the pan axis. As a result, the spherical heat map is generated for resolutions 1×2, 1×4, and 1×8. Again, these scales are stacked to build a 14-dimension feature vector. Unlike the heat map in the world coordinate system, the spherical heat map is specific to a particular camera location $\mathbf{C}$. Effectively, the spherical map is a polar quantization of the player positions on the basketball court.

Ball Currently, we do not use any direct information about the ball because estimating its 3D location in monocular video is ill-posed (unless the ball is in contact with the floor). Moreover, players are coached to be in the right place at the right time. As a result, the spatial formation of the offensive and defensive players usually gives strong clues about the ball's location [21]. Our experiments illustrate a similar pattern: the learned regressor often follows the ball, even though it has no direct information about the ball. Of course, if reliable ball data was available as a feature, we would expect improved learning performance.

3.2. Labels

In addition to feature vectors $\{\mathbf{x}_t\}$, we also require corresponding ground truth labels $\{\mathbf{y}_t\}$, which in this application are camera pan angles. Therefore, we now describe how we estimate the pan angle of the human operated camera at each video frame.

The pinhole model is frequently used to describe the projective aspects of a camera [19]

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I}| - \mathbf{C}] \tag{2}$$

where $\mathtt{K}$ is the intrinsic matrix, $\mathtt{R}$ is a rotation matrix from world to camera coordinates, and $\mathbf{C}$ is the camera's center of projection. Assuming square pixels and a principle point at the image center, the focal length f is the only degree of freedom in the intrinsic matrix $\mathtt{K}$.

Generally, a PTZ camera system has two separate components: a camera and a robotic head. The rotation matrix $\mathtt{R}$ changes as robotic head moves. Thus, we factor $\mathtt{R}$ into two rotation matrices $\mathtt{Q}$ and $\mathtt{S}$ [3]

$$\mathtt{R} = \mathtt{Q}\mathtt{S}. \tag{3}$$

The rotation matrix $\mathtt{S}$ represents the rotation from the world coordinate system to pan-tilt motor coordinate system, and remains constant regardless of the actual pan-tilt settings. We model the rotation matrix using the Rodrigues notation $\mathtt{S} = [s_x, s_y, s_z]^{\mathsf{T}}$. The matrix $\mathtt{Q}$ represents the 3D rotation for a specific pan-tilt (θ, ϕ) setting

$$\mathtt{Q} = \mathtt{Q}_\phi \mathtt{Q}_\theta \tag{4}$$

where

$$\mathtt{Q}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \mathtt{Q}_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{5}$$

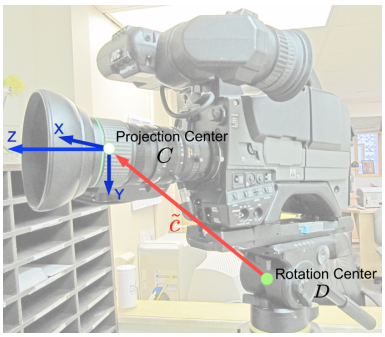


Figure 3. **Displacement**. When a PTZ camera is realized by mounting a large camera on a tripod, the projection center and the rotation center are usually significantly far apart. As a result, it is necessary to model this displacement.

Most PTZ models assume the rotation center is the same as the projection center. However, it is only an approximation [1]. Some cameras obviously do not obey this assumption. For example, Figure 3 shows a professional broadcast camera. The camera is mounted on a tripod, and rotates around the tripod head (green circle). The projection center (white circle) is significantly far from the rotation center. Therefore, we refine the camera model to account for this displacement

$$\mathtt{P} = \mathtt{K}\tilde{\mathtt{C}} \begin{bmatrix} \mathtt{R} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathtt{I} & -\mathbf{D} \\ 0 & 1 \end{bmatrix}. \tag{6}$$

Here, $\mathbf{D}$ is the center of rotation, and $\tilde{\mathtt{C}}$ is the translation from the center of rotation to the center of projection

$$\tilde{\mathtt{C}} = \begin{bmatrix} 1 & 0 & 0 & \tilde{c}_x \\ 0 & 1 & 0 & \tilde{c}_y \\ 0 & 0 & 1 & \tilde{c}_z \end{bmatrix}. \tag{7}$$

The center of projection changes significantly whenever the camera zooms in or out. Therefore, we model the displacement between the center of rotation and the center of projection as a linear function of f

$$\begin{aligned} \tilde{c}_x &= \lambda_1 + \lambda_4 f, \\ \tilde{c}_y &= \lambda_2 + \lambda_5 f, \\ \tilde{c}_z &= \lambda_3 + \lambda_6 f. \end{aligned} \tag{8}$$

In our PTZ camera model, there are twelve time invariant parameters $\Phi = [D_x, D_y, D_z, s_x, s_y, s_z, \lambda_1, ..., \lambda_6]^\mathsf{T}$ and three per-frame parameters $[\theta, \phi, f]^\mathsf{T}$. Constrained by the common parameters Φ, our model can estimate pan, tilt and focal length from at least two correspondences. Okubo *et al*. [16] employed a similar idea, but included an additional rotation between the rotation center and projection center.

Parameter Estimation We begin by independently estimating the calibration matrix $\mathtt{P}_j$ of each video frame j using the standard pinhole model (2). Afterwards, we employ

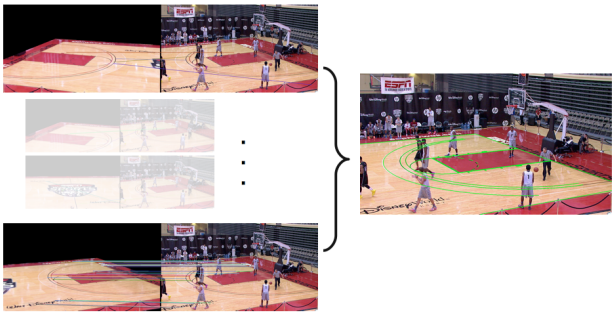


Figure 4. **Keyframe Matching**. The left column shows the warped overview images from different keyframes and the corresponding matched points. The right column show the final calibration.

Levenberg-Marquardt optimization to estimate the time invariant parameters Φ and per-frame pan-tilt-zoom settings by minimizing the projection error of key points

$$\Phi, \{\theta_j, \phi_j, f_j\} = \arg\min \sum_i \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|^2. \tag{9}$$

Here, $\mathbf{m}_i$ is the observed image location of known 3D point $\mathbf{M}_i$ (corners of basketball court markings), and $\hat{\mathbf{m}}_i$ is the projection of $\mathbf{M}_i$ by $\mathtt{P}(\Phi, \theta_j, \phi_j, f_j)$. We find locations of key points in the input video frame by searching for matching key points from multiple manually calibrated keyframes. To perform the matching, we employ two homographies

$$\mathbf{m}_{i,k} = \mathtt{H}\mathtt{H}_k^{-1}\mathbf{M}_i \tag{10}$$

where $\mathbf{M}_i$ is the 3D position of the key point, $\mathtt{H}_k$ is the homography mapping $\mathbf{M}_i$ to a manually calibrated keyframe k, $\mathbf{m}_{i,k}$ is the corresponding key point in the input video frame, and $\mathtt{H}$ is the homography between keyframe k and the input video frame.

We use SIFT matching between the keyframe and input video frame to estimate $\mathtt{H}$. To filter out noisy correspondences, we employ the method of Hess *et al*. [11] and synthesize an overhead view image of the basketball court by combining multiple calibrated frames. A database of six keyframes was constructed using the method of [5] to obtain accurate camera parameters. Finally, we warp the overhead image to the vantage point of each keyframe using the homography matrix $\mathtt{H}_k$. Because the warped overhead image has almost no clutter (such as players), matches to an input video frame typically have few outliers. We calculate the homography matrix $\mathtt{H}$ between the input video frame and each keyframe using SIFT [14]. We greedily select the keyframe with the minimum sum of squared differences (SSD) between the warped overhead image and the input image based on where we expect to see court markings. Figure 4 shows an example of keyframe matching. For a long video sequence, we linearly interpolate $[\theta, \phi, f]^\mathsf{T}$ for

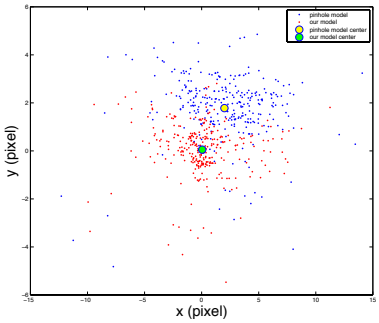


Figure 5. **Reprojection Error Distribution**. The reproduction errors of our model have smaller variance and bias.

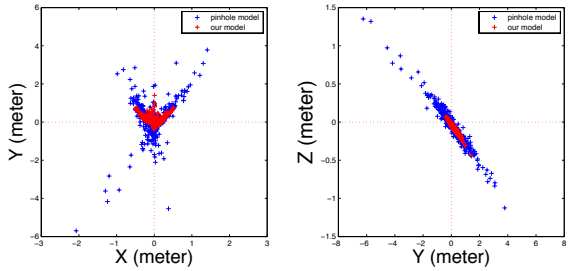


Figure 6. **Projection Center Distribution**. The distribution of the projection center C relative to its average $\bar{C}$ for the two calibration methods. We constrain the projection center's location relative to the center of rotation which results in more plausible calibration estimates.

frames where calibration fails (typically because of motion blur when the camera is moving quickly).

Figure 5 shows the projection error distribution of our revised PTZ model and the typical pinhole model. The data is from 26 images in which the correspondences were manually identified. The RMS error of our method is 3.1px, whereas the pinhole model is 4.6px. Since our method assumes $\tilde{\mathbf{c}}$ is a linear function of focal length, the $\tilde{c}_x$ and $\tilde{c}_y$ components can account for linear variations in the principal point. As a result, our model can also model the shift of the principle point as f varies. Figure 5 illustrates how our average projection error is very close to $(0, 0)$ while the pinhole model is significantly far from $(0, 0)$ implying the principal point is not always exactly at the center of the image. We also tested our model on two additional basketball broadcast datasets [3, 15]. In both cases, our model outperforms the standard pinhole model (see Table 1).

Figure 6 shows the distribution of $\mathbf{C}$ relative to its average location $\bar{\mathbf{C}}$ for both the pinhole model and our model. The data is from 389 calibrated frames which were uniformly sampled from the recorded video. Because the projection center cannot be outside the camera body, the results of the pinhole model look unrealistic. Our model, on the other hand, constrains the projection center to a reasonable range.

Dataset	#points	Pinhole RMS	Pinhole median	Our model RMS	Our model median
Ours	296	4.6	2.1	3.1	1.6
CMU [3]	233	4.2	3.2	3.9	2.8
NBA [15]	412	2.6	1.8	2.0	1.4

Table 1. **Reprojection Error Comparison**. Our model has lower reprojection error (measured by RMS and median pixel distance) on all three basketball datasets.

Method	Feature					
	Centroid		Heat Map		Spherical Map	
	Train	Test	Train	Test	Train	Test
LSQ	6.6	**3.9**	7.1	5.1	7.4	4.8
SVR	6.2	4.3	3.2	3.1	3.9	2.9
RF	6.1	4.6	0.9	**3.0**	1.8	**2.7**

Table 2. **Regression Methods**. The RMS error (in degrees) of predicted pan angles for various regression methods and feature combinations. The lowest test error for each feature is emphasized. Training error is reported on the entire training set after selecting optimal parameters.

4. Learning

Having extracted a set of tracking features $\{\mathbf{x}_t\}$ and corresponding exemplar pan angles $\{y_t\}$, we now describe how we learned a regressor $h(\cdot)$ to predict pan angles $\{\hat{y}_t\}$ for new game situations $\{\mathbf{x}_t\}$. We considered three well established techniques: linear least squares regression (LSQ), support vector regressor (SVR) and random forest regression (RF). For each learning method, we investigated the three feature representations discussed in Section 3.1. We used about 16 minutes of video (at 60fps) to generate 60,496 training examples. We employed ten-fold cross validation to determine the optimal soft margin in SVR, and out-of-bag testing to decide the optimal number of trees and tree depth in RF[1]. An additional one minute of video was held out and set aside for use as 3,562 testing examples. Because our goal is to mimic a human operator, the ground truth labels for the test set are the pan angles of the human operator. The centroid feature is our baseline, as it is effectively the method of [3].

Table 2 shows the RMS error for each regression method and feature combination. For the low-dimensional centroid feature, least squares regression achieved the best performance with an average RMS error of $3.9°$. However, for the high-dimensional heat map and spherical map features, the more complex regression algorithms were able to achieve substantially lower RMS errors of $3.0°$ and $2.7°$ respectively (performance improvements of $\sim 20\%$ and $\sim 30\%$

[1] See the supplemental material for additional details.

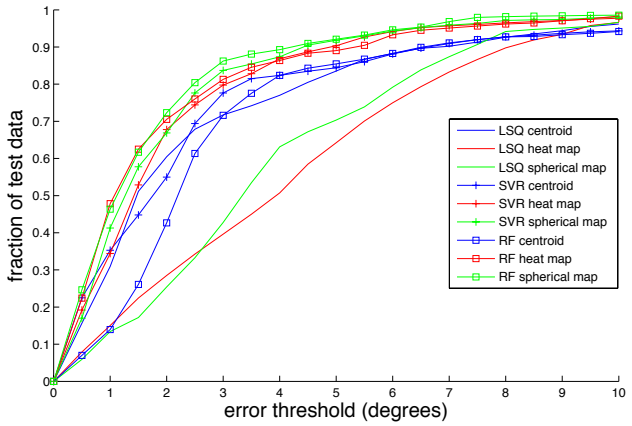
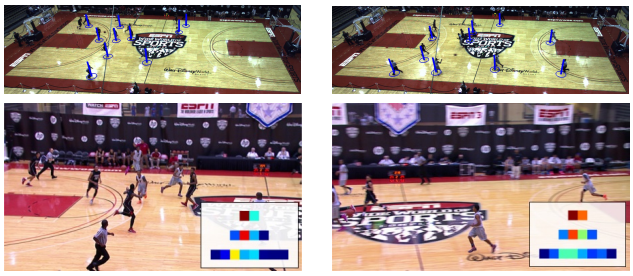


Figure 7. **Error Distribution**. The cumulative fraction of the test data where the prediction error is less than a specified threshold. Better predictors will populate the top-left corner of the plot.

compared to predictions based on the centroid feature using LSQ). In both cases, RFs achieved slightly better performance relative to SVR. Figure 7 shows the cumulative fraction of test data that fell within a specified error tolerance for each regression algorithm and feature combination. Methods which have a substantial number of small errors will quickly approach a high fraction (i.e. the top left corner of the plot). RF using spherical maps achieved the the best result, but RF using heat maps and SVR using spherical maps were almost equally as good. However, all methods clearly have a small number of occasions (less than 10% of the test data) where there are large $>10^\circ$ discrepancies between the predicted pan angles $\{\hat{y}_t\}$ and the actual pan angles $\{y_t\}$ of the human operator.

There are several factors which may lead to inaccurate predictions, such as errors in detecting and tracking players. However, in our test sequence, we have considered the human operator's actions as ground truth, which results in two implicit assumptions: (1) there is a single optimal pan angle y_t^* for a particular situation $\mathbf{x}_t$, and (2) the human operator never makes a mistake — *i.e.* $y_t \approx y_t^*$. On closer inspection of the data, we find neither of these assumptions is always true.

Multivalued Function Figure 8 shows two similar distributions of players. The computed spherical map features (inset) are roughly equivalent. However, in the first situation the camera is panned to the left, while in the other the camera is panned significantly to the right. Although the features $\mathbf{x}_a \approx \mathbf{x}_b$ are similar, the pan angles $y_a \neq y_b$ are not. As a result, $h(\mathbf{x}_t)$ is not strictly a single valued function. The same formation of players may have multiple possible correct pan angles — *i.e.* $h(\mathbf{x}_t) \mapsto \{y_t, y_t', y_t'', \dots\}$.



(a) (b)

Figure 8. **Multivalued Function**. (a) and (b) show two roughly similar distributions of players at different times in the game. The resulting spherical maps (inset) are also quite similar. However, the pan angles of the human operated camera are substantially different.

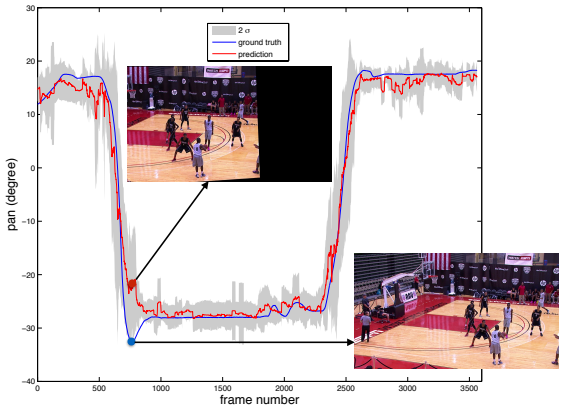


Figure 9. **Human Variability**. The gray area shows a two standard deviation confidence interval (average uncertainty is $\pm 2.2^\circ$). The inset image in the lower right shows the human operator with a poorly framed shot (half of the image is of an unoccupied area of the basketball court). The inset in the top left show the predicted framing from the learned regressor. In this situation, the prediction error is quite large, but the discrepancy is the result of a poor ground truth label, and not a bad prediction.

Human Variability Our approach contains an underlying assumption that the human operator has perfect control of the camera. Figure 9 shows the empirical variance learned by the RF regressor. The variance arises from two sources: player tracking errors which result in noisy feature representations, and variation in how the human has operated the camera during similar formations of players. Frequent small variations arise from control errors — *i.e.* the operator is putting the camera in roughly the same configuration, but not exactly the same configuration. Larger variations occur when a camera operator incorrectly anticipates how the action will unfold, and ends up with an incorrect framing. As a result, our ground truth (what the human operator did with the camera) is only an approximation of the what ideally should have been done with the camera. Furthermore, large errors in our predicted pan angles $\{\hat{y}_t\}$ may in fact be

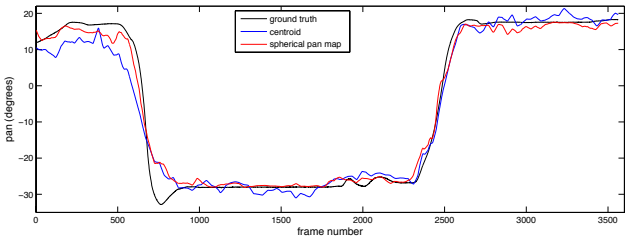


Figure 10. **Smoothed Predictions**. The raw predictions are smoothed using a first-order Savitzky-Golay filter. These revised predictions are used to generate the synthetic video for qualitative analysis.

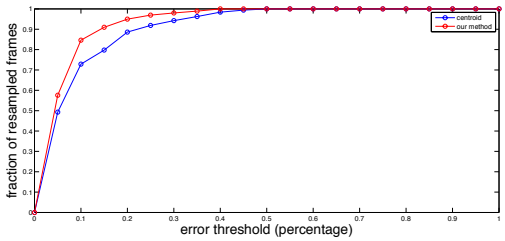


Figure 11. **Image Overlap**. We asses how well each algorithm is able to mimic the human operator by counting the number of missed pixels when synthesizing how the autonomous algorithms would have filmed the same scene. An algorithm which closely mimics the human should populate the top-left corner of the plot.

valid predictions that simply coincided with human operator error (such as the circumstance depicted in Figure 9). Therefore, in the next section, we investigate an alternative comparison technique which evaluates what the human operator did with the camera versus what an autonomous planning algorithm would have done with the camera in the same situation.

5. Evaluation

Smooth motion is critical for aesthetic camera work [18]. Therefore, we use a first-order Savitzky-Golay filter [20] of 33 frames (0.5s) to smooth the predicted pan angles. Figure 10 shows the smoothed prediction of baseline (centroid) and our method (spherical map). The prediction error of our method (1.7°) is substantially smaller than the baseline (3.0°).

Finally, we evaluate the various prediction algorithms using *re-cinematography* [9], which generates new video by resampling previously recorded video. When resampling the recorded broadcast, we fix the focal length and tilt angle as ground truth, and set the pan angle to the predicted value $\hat{y}_t$. Since the prediction $\hat{y}_t$ is generally different from the ground truth y_t, the resampled video will have missing pixels because the resampled frame will go beyond the bounds of the recorded video. As a result, we can gauge how well an algorithm mimicked the human operator by the magnitude of missing pixels in the synthesized video. Figure 11 shows the cumulative fraction of missing pixels present in a resampled video. Our method has significantly smaller errors compared to the baseline. For qualitative comparisons, please consult the supplementary material.

6. Discussion

In this work, we proposed a data-driven method to predict the pan angle of PTZ camera from the distribution of players on a basketball court. To the best of our knowledge, we are the first to pose camera planning as a supervised regression problem. More importantly, our method provides realtime predictions which closely resemble the work of a human operator for the same game situation. In order to learn from exemplar recorded video, our method also includes a new calibration algorithm for obtaining reliable pan-tilt-zoom camera parameters from long video sequences.

We have demonstrated our approach on basketball using noisy realtime player tracking data. Although ball detection would be useful, our results indicate that it is not absolutely necessary (the formation of the players often implies the location of the ball carrier [21]). Figure 12 illustrates how the learned regressor is able to keep the ball in view during the setup of a half court press. In contrast, the centroid method focuses on a vacant region of the court because the distribution of players is bimodal.

Currently, we have only evaluated our method on basketball. However, we expect a similar approach will work for other sports as well (possibly using additional features). In future work, we plan to evaluate our method on different sports and different competitive levels. We will also investigating mimicking the auxiliary cameras which are used for cut-aways in a multi-camera production.

References

[1] L. Alvarez, L. Gomez, P. Henriquez, and J. Sánchez. Real-time camera motion tracking in planar view scenarios. *Journal of Real-Time Image Processing*, 2013. 2, 4

[2] Y. Ariki, S. Kubota, and M. Kumano. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, pages 851–860. IEEE, 2006. 2

[3] P. Carr, M. Mistry, and I. Matthews. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 193–202. ACM, 2013. 3, 5

[4] P. Carr, Y. Sheikh, and I. Matthews. Monocular object detection using 3d geometric primitives. In *ECCV, 2012*, pages 864–878. Springer, 2012. 2

[5] P. Carr, Y. Sheikh, and I. Matthews. Point-less calibration: Camera parameters from gradient-based alignment to edge images. In *WACV, 2012*, pages 377–384. IEEE, 2012. 4

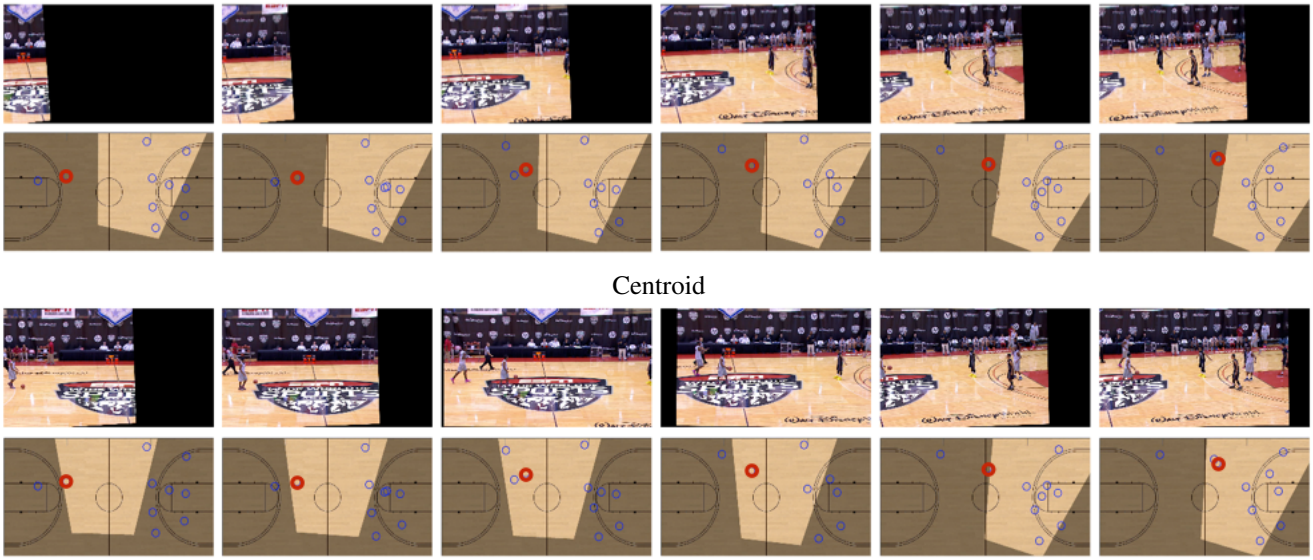


Figure 12. **Half Court Press**. During a half court press, two guards bring the ball up to the half court, while the remainder of the players setup in the offensive zone. Because the distribution of players is bimodal, the centroid (top) is not a good approximation of where the camera should look. Instead, the learned regressor (bottom) generally follows the ball carrier (highlighted in red) with appropriate lead room [18], even though the regressor has no direct information about the ball — only noisy player tracking data is provided.

[6] F. Chen, D. Delannay, and C. De Vleeschouwer. An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study. *Multimedia, IEEE Transactions on*, 13(6):1381–1394, 2011. 2, 3

[7] J. Chen and P. Carr. Autonomous camera systems: A survey. In *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014. 1, 2

[8] S. Daigo and S. Ozawa. Automatic pan control system for broadcasting ball games based on audience's face direction. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 444–447. ACM, 2004. 2

[9] M. L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 5(1):2, 2008. 7

[10] A. Gupta, J. J. Little, and R. J. Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 32–39. IEEE, 2011. 1, 2

[11] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *CVPR, 2007*, pages 1–8. IEEE, 2007. 1, 2, 4

[12] K. Kim, D. Lee, and I. Essa. Detecting regions of interest in dynamic scenes with camera motions. In *CVPR, 2012*, pages 1258–1265. IEEE, 2012. 2

[13] J. Liu, P. Carr, R. T. Collins, and Y. Liu. Tracking sports players with context-conditioned motion models. In *CVPR, 2013*, pages 1830–1837. IEEE, 2013. 2

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 5

[15] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy. Learning to track and identify players from broadcast sports videos. *PAMI*, 35(7):1704–1716, 2013. 5

[16] H. Okubo, Y. Yamanouchi, H. Mitsumine, T. Fukaya, and S. Inoue. Accurate camera calibration method specialized for virtual studios. In *Electronic Imaging 2008*, pages 68040F–68040F. International Society for Optics and Photonics, 2008. 4

[17] K. Okuma, J. J. Little, and D. G. Lowe. Automatic rectification of long image sequences. In *ACCV*, 2004. 1

[18] J. Owens. *Television Sports Production*. Focal Press, fourth edition, 2007. 1, 7, 8

[19] J. Puwein, R. Ziegler, L. Ballan, and M. Pollefeys. Ptz camera network calibration from moving people in sports broadcasts. In *WACV, 2012*, pages 25–32. IEEE, 2012. 3

[20] A. Savitzky and M. J. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 7

[21] X. Wang, V. Ablavsky, H. B. Shitrit, and P. Fua. Take your eyes off the ball: Improving ball-tracking by focusing on team play. *CVIU*, 119:102–115, 2014. 3, 7