

Learning Sports Camera Selection from Internet Videos

Jianhui Chen * Keyu Lu † Sijia Tian * James J. Little *

*University of British Columbia †National University of Defense Technology

{jhchen14, candice, little}@cs.ubc.ca keyu.lu@nudt.edu.cn

Abstract

This work addresses camera selection, the task of predicting which camera should be “on air” from multiple candidate cameras for soccer broadcast. The task is challenging because of the scarcity of learning data with all candidate views. Meanwhile, broadcast videos are freely available on the Internet (e.g. Youtube). However, these videos only record the selected camera views, omitting the other candidate views. To overcome this problem, we first introduce a random survival forest (RSF) method to impute the incomplete data effectively. Then, we propose a spatial-appearance heatmap to describe foreground objects (e.g. players and balls) in an image. To evaluate the performance of our system, we collect the largest-ever dataset for soccer broadcasting camera selection. It has one main game which has all candidate views and twelve auxiliary games which only have the broadcast view. Our method significantly outperforms state-of-the-art methods on this challenging dataset. Further analysis suggests that the improvement in performance is indeed from the extra information from auxiliary games.

1. Introduction

The sports market in North America was worth 69.3 billion in 2017 and is expected to reach 78.5 billion by 2021. Our work focuses on soccer game which has about 40% shares of the global sports market. As the biggest reason for market growth, media rights (game broadcasts and other sports media content) are projected to increase from 19.1 billion in 2017 to 22.7 billion in 2021 [1]. Computational broadcasting is a promising way to offer consumers with various live game experiences and to decrease the cost of media production. Automatic camera selection is one of the key techniques in computational broadcasting.

Machine learning has produced impressive results on view point selection. These include automatic [7, 33, 32, 22, 8] and semi-automatic [16] methods from various inputs such as first-person cameras, static and pan-tilt-zoom (PTZ) cameras. The underlying assumption of these meth-

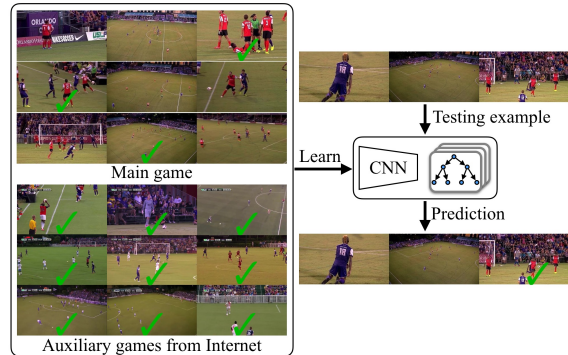


Figure 1: Learning camera selection from Internet videos. The goal of our work is to select one camera from multiple candidate cameras for sports broadcast. The ideal way is trained from a dataset that has all candidate views such as the main game shown in the figure. However, it is hard to acquire this kind of data (including candidate videos and broadcasting videos) because they are generally not available to researchers (owned by broadcasting companies). Our method uses publicly available Internet videos as auxiliary data to train a model with state-of-the-art prediction accuracy. Best viewed in color.

ods is that large training data are easily available.

Motivation For sports camera selection, amounts of large training data are not directly available. As a result, most previous methods are trained on a single game (main game) because researchers can not acquire the data that are owned by broadcasting companies [6, 8]. On the other hand, broadcast videos are widely available on the Internet (e.g. Youtube). These games (auxiliary games) provide a large number of positive examples. Using these Internet videos can scale up the training data with negligible cost.

In practice, arbitrarily choosing auxiliary games does not necessarily improve the performance, when main games are from minor leagues while auxiliary games are from premier leagues. So, the main game and the auxiliary games should be similar in terms of camera locations and the action of players. Although a universal camera selection model should be the final goal, a model for a specific team

is also valuable. For example, teams in minor leagues can reduce the cost of live broadcasting for host games. Targeting these applications, we constrain the main games and auxiliary games to be from the same stadium at the current stage.

The main challenge of using auxiliary games is the missing views in the video composition. Omitting non-broadcast views is the default setting for TV channels and live streams on the Internet. As a result, the amount of complete and incomplete data is highly unbalanced. To overcome this challenge, we introduce the random survival forest method (RSF) [24] from statistical learning to impute the missing data. To the best of our knowledge, we are the first to use Internet videos and RSF to solve camera selection problems.

The second challenge is from the potentially negative impact of background information in auxiliary games. Auxiliary games are very different in lighting, fan celebration and stadium decoration. In practice, camera operators are trained to capture interesting players and keep the ball visible [30]. Inspired by this observation, we propose a spatial-appearance heatmap to represent foreground objects locations and their appearances jointly.

Our main contributions are: (1) Using Internet data and random survival forests to address the data scarcity problem in camera selection for soccer games. (2) Proposing a spatial-appearance heatmap to effectively represent foreground objects. With these novel techniques, our method significantly outperforms state-of-the-art methods on a challenging soccer dataset. While we present results on soccer games, the technique developed in this work can apply to many other team sports such as basketball and hockey.

2. Related Work

Data Scarcity and Imputation The availability of a large quantity of labeled training data is critical for successful learning methods. This assumption is unrealistic for many tasks. As a result, many recent works have explored alternative training schemes, such as unsupervised learning [44], and tasks where ground truth is very easy to acquire [2]. We follow this line of work with additional attention to data imputation approaches.

Data imputation fills in missing data from existing data [14]. The missing data falls into three categories: missing at random (MAR), missing completely at random (MCAR) and missing not at random (MNAR). The missing data in our problem is MNAR because the missing data is related to the value itself (all missing data is unselected by human operators). Our solution is adapted from the state-of-the-art random survival forests method [24, 34].

Camera Viewpoint Prediction In single camera systems, previous techniques have been proposed to predict camera angles for PTZ cameras [7] and to generate natural-

looking normal field-of-view (NFOV) video from 360° panoramic views [33, 32, 22]. In multi-camera systems, camera viewpoint prediction methods select a subset of all available cameras [39, 12, 35, 3, 17, 18, 43, 40, 26]. In broadcast systems, semi-automatic [16] and fully automatic systems have been developed in practice. For example, Chen *et al.* [6] proposed the automated director assistance (ADA) method to recommend the best view from a large number of cameras using hand-crafted features for field hockey games. Chen *et al.* [8] modeled camera selection as a regression problem constrained by temporal smoothness. They proposed a cumulative distribution function (CDF) regularization to prevent too short or too long camera durations. However, their method requires a real-valued label (visual importance) for each candidate frame. Our problem belongs to multiple dynamic camera systems.

Video Analysis and Feature Representation Team sports analysis has focused on player/ball tracking, activity recognition, localization, player movement forecasting and team formation identification [23, 15, 42, 29, 36, 27, 19]. For example, activity recognition models for events with well defined group structures have been presented in [23]. Attention models have been used to detect key actors [31] and localize actions (*e.g.* who is the shooter) in basketball videos. Gaze information of players have been used to select proper camera views [3] from first-person videos.

Hand-crafted features [7, 15], deep features [37] and semantic features [5] have been used to describe the evolution of multi-person sports for various tasks. Most deep features are extracted from the whole image using supervised learning [8]. On the other hand, object-level (*e.g.* image patches of players) features are difficult to learn because of the lack of object-level annotations. Our object appearance features are learned from a siamese network [9] without object-level annotations.

3. USL Broadcast Dataset

We collect a dataset from United Soccer League (USL) 2014 season. The dataset has one main game and twelve auxiliary games. The main game has six videos. Two videos are from static cameras which look at the left and right part of the playing field, respectively. Three other videos are from pan-tilt-zoom (PTZ) candidate cameras (1280×720). Among them, one camera was located at mid-field, giving an overview of the game. The other two cameras are located behind the left and right goals respectively, providing detailed views. Figure 2 visualizes the camera locations and shows image examples. The sixth video is the broadcast video composited by a professional director from the three PTZ videos. We manually remove commercials and “re-plays” and synchronize this video with other videos. The length of the main game is about 94 minutes. Our system only uses information from the three PTZ cameras to select

Dataset	Year	Game type	Length (min.)	# Game	# Camera	Camera type	Ground truth
APIDIS [12]	2011	basketball	15	1	5	static	non-prof.
ADS [6]	2013	field hockey	70	1	3	PTZ	prof.
OMB [16]	2013	basketball	~ 16	1	2	PTZ	non-prof.
VSR [41]	2014	soccer	9	1	20	static	non-prof.
CDF [8]	2018	soccer	47	1	3	PTZ	hybrid
USL (Ours)	2018	soccer	94 + 108	1+12	3	PTZ	prof.

Table 1: Dataset comparison. In our dataset, the 108 minutes data (column four) is sparsely sampled from total 1,080 minutes data.

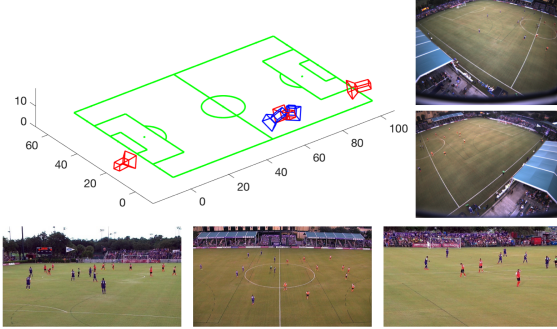


Figure 2: Camera settings of the main game and image examples. Blue: static cameras; red: PTZ cameras.

the broadcast camera. Static cameras are only used in one of the baselines.

Twelve auxiliary games are collected from Youtube. These games are hosted in the same stadium as the main game. They are typically 1.5 hours long. Unlike the main game, each auxiliary game only has a composited broadcast video (640×360). Figure 1(bottom left) shows image examples from the auxiliary games.

In the main game, we manually annotate the ball locations on the static cameras and two detailed view PTZ cameras, at 1fps. In the auxiliary games, we manually check the classified camera IDs around detected camera transition points (details in Section 5). In all games, we detected bounding boxes of players and balls using a method similar to [11]. Table 1 compares our dataset with previous camera view selection datasets. To the best of our knowledge, ours is the first dataset with dense annotations for such long dynamic multi-camera image sequences. We put more dataset details in the supplementary material.

4. Method

4.1. Problem Setup

We have two sources of data. One is from complete games which have videos and selections. Another is from auxiliary games which only have one broadcast video and selections. We model the problem as a classification task given hybrid data $D = \{D_{com}, D_{incom}\}$ in which D_{com} is the *complete* data and D_{incom} is the *incomplete* data. Let

$D_{com} = \{X_{com}, Y\}$ where X_{com} is the feature representation of all candidate views and $Y \in \{1, 2, 3\}$ is the corresponding label. X can be an arbitrary feature representation for an image. Let $D_{incom} = \{\{X_{obs}, X_{mis}\}, Y\}$ where X_{obs} is the *observed* data and X_{mis} is the *missing* data (e.g. unrecorded views). Our goal is to learn a classifier from the whole data to predict the best viewpoint from multiple candidate viewpoints (e.g. an unseen X_{com}):

$$y_t = f(\mathbf{x}_t). \quad (1)$$

We do instantaneous single frame prediction and \mathbf{x}_t is a feature representation from all camera views. During training, \mathbf{x}_t is either a raw feature extracted from the main game, or a raw plus imputed feature from an auxiliary game. We only test on the main game.

Our primary novelty is to use auxiliary data from the Internet which augments the training data with lots of positive examples. On the other hand, this choice creates considerable challenges because of the missing data.

Assumptions and Interpretation Our method has three assumptions. First, $X_{inputed} = \{X_{obs}, \hat{X}_{mis}\}$ (\hat{X} means the inferred values) and $X_{inputed}$ has a similar distribution as X_{com} . This assumption is reasonable since both types of games are collected from the same stadium with the same host team. Also, we expect the broadcast crew to have consistent behaviors across games to some extent. Second, images from different viewpoints are correlated at a particular time instance. Camera operators (from different viewpoints) cooperate to tell the story of the game. For example, often the focus of attention of the cameras is the same (*i.e.* joint attention). In this case, the observed data X_{obs} has a strong indication of the missing data X_{mis} . Third, our method models the viewpoint prediction problem as single frame prediction problem without using temporal information. Single-frame prediction is the focus of our work. We will briefly show the adaptation of our method to a temporal model in the experiment.

4.2. Random Survival Forest

With these assumptions, we first impute missing data in training. We randomly draw imputed data from the joint

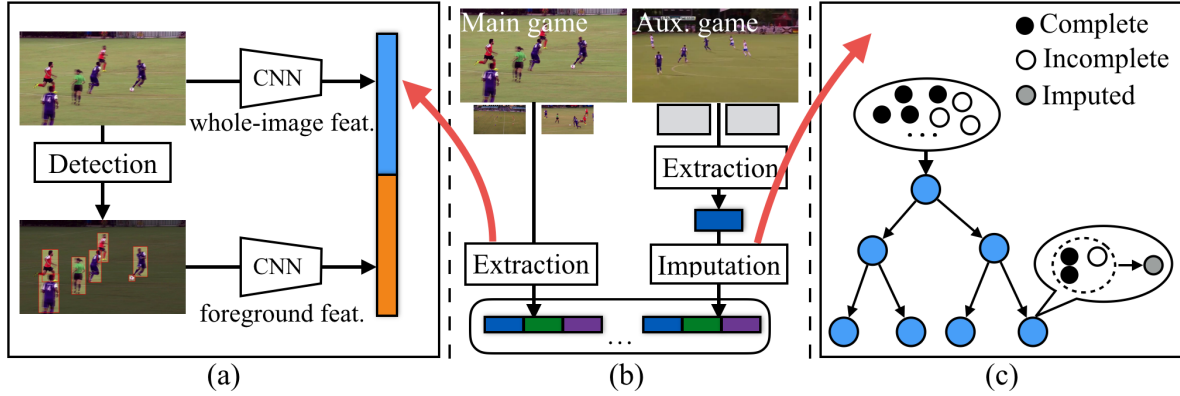


Figure 3: Main components of our method. (a) Feature extraction. Two CNNs are used to extract whole-image and foreground features. (b) Training process. We first extract features from both main game and auxiliary game frames. The feature of auxiliary games is imputed for the missing data. Both data are then used to train the final model. (c) Data imputation (Section 4.2). Best viewed in color.

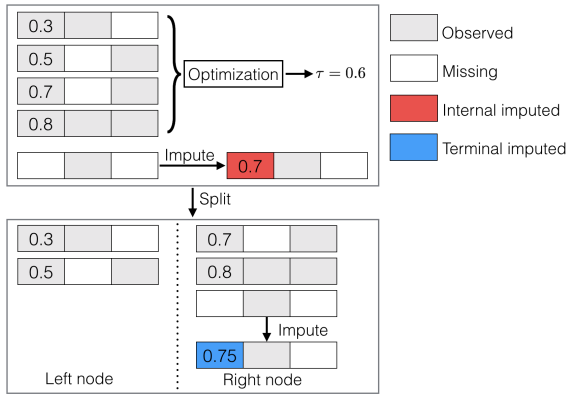


Figure 4: A two-level random survival tree. Each row represents a three-dimensional feature. The first dimension of the fifth feature is imputed. τ is the decision boundary. Labels are omitted for clarity. Best viewed in color.

posterior distribution of the missing data given the observed data [38].

$$X_{mis} \sim p(X_{mis}|X_{obs}, Y) \quad (2)$$

with

$$p(X_{mis}|X_{obs}, Y) = \int p(X_{mis}|X_{obs}, \theta)p(\theta|X_{obs}, Y)d\theta, \quad (3)$$

where θ is the model which is decision trees in our method and Y is the label. Please note this process is in training phase so that Y is available. However, it is often difficult to draw from this predictive distribution due to the requirement of integrating over all θ . Here we introduce random survival forests to simultaneously estimate θ and draw imputed values.

A random survival forest (RSF) is an ensemble of random survival trees, which was originally designed to identify a complex relationship between long-term survival and

attributes of persons (e.g. body mass, kidney function and smoking). Each decision tree recursively splits training data into sub-trees until the stopping criteria is satisfied. The statistics (e.g. mean values of labels for regression) of training examples in the leaf nodes are used as the prediction [10]. A survival tree imputes missing data as below.

1. In internal nodes, only observed data is used to optimize tree parameters such as the decision boundary by minimizing the cross entropy loss. This step estimates the model θ from the distribution $p(\theta|X_{obs}, Y)$ in (3).
2. To assign an example with missing data to the left or right sub-trees, the missing value is “imputed” by drawing a random value from a uniform distribution $U(x|a, b)$ where (a, b) are the lower/upper bounds of X_{obs} of the target dimension. This step draws samples from $p(X_{mis}|X_{obs}, \theta)$ in (3).
3. After the node splitting, imputed data are reset to missing and the process is repeated until terminal nodes are reached.
4. Missing data in terminal nodes are then imputed using non-missing terminal node data from all the trees. For categorical variables, a majority vote is used; a mean value is used for continuous variables.

Figure 4 shows the data imputation in a two-level random survival tree. Specific details of RSF can be found in [24, 34]. With the RSF method, we impute the missing data with substituted values to obtain the new data $\{X_{imputed}, Y\}$. To the best of our knowledge, we are the first to introduce RSF from statistical learning to solve vision problems. Besides, we will experimentally show that it outperforms other alternatives in our problem.



Figure 5: Spatial-appearance heatmap. Left: one player on a 4×4 grid; right: an example of detected objects and corresponding heatmap.

4.3. Foreground Feature

Besides the whole-image feature from a CNN, we also represent foreground objects in an image using a spatial-appearance (SA) heatmap which encodes object appearances in a quantized image space. First, we quantized the image space into a 16×9 grid. Then, we represent the location of each player using five points (four corners and one center point) of its bounding box. Each point contributes “heat” to its located and neighboring cells. In the conventional heatmap, the “heat” is pre-defined values such as the number of players [7]. In our heatmap, the “heat” is the object appearance feature that is learned from the data.

Figure 5 (left) illustrates how the SA heatmap is computed on a 4×4 grid. The bottom right corner of the bounding box contributes the weighted “heats” to C_1, C_2, C_3 and C_4 . The weights are the barycentric coordinates of the corner with respect to four cell centers. We use the heatmap as input to train a binary classification CNN and its second-last fully connected layer is used as the foreground feature.

Appearance Feature Learning Given the detected bounding boxes of the objects [11], we use a siamese network [9] to learn object appearance features. We train the siamese network using the player tracking information between frames and extract features from image patches of players in testing. To train the network, we obtain positive (similar) examples from tracked players [28] in consecutive frames (*e.g.* from frame 1 to frame 2). The underlying assumption is that the tracked players in consecutive frames have similar appearance, pose and team membership. Any player not part of a track is likely to be dissimilar. The siamese network minimizes the contrastive loss [20]:

$$L_c(\mathbf{x}_i, \mathbf{x}_j, y_{i,j}) = y_{i,j} D(\mathbf{x}_i, \mathbf{x}_j)^2 + (1 - y_{i,j}) \max(\delta - D(\mathbf{x}_i, \mathbf{x}_j), 0)^2, \quad (4)$$

where \mathbf{x}_i and \mathbf{x}_j are sub-images, $y_{i,j}$ are similar/dissimilar labels, $D(\cdot)$ is the L_2 norm distance and δ is a margin (1 in this work). The loss function minimizes the distance between paired examples when $y_{i,j} = 1$, and maximizes the distance according to the margin δ when $y_{i,j} = 0$.

5. Implementation

Label Estimation of Internet Videos We pre-process Internet videos for training labels. Given a raw video, we first detect shot boundaries using [13]. We call the consecutive frames at the shot boundary *boundary frames* for simplicity. Given boundary frames, we train a CNN to classify their camera IDs to four categories (*i.e.* left, middle, right and other-view). The other-view images are commercials, replay logos or frames that are captured from other view-points. To train the camera-ID CNN, we first randomly sample 500 training frames from each PTZ video of the main game. For the other-view, we sample the same number of images from a non-sports video. Then, we apply the trained model to classify boundary frames. The classification result is manually checked and refined. The refined boundary frames are used to re-train the CNN. This process is repeated for each video. After five games, the prediction accuracy is about 85%. We found this performance is sufficient to lighten the workload of human annotation. Initialized by the CNN then manually corrected, we collect 1,634 pairs of boundary frames from twelve videos.

Feature Extraction Each frame is represented by two types of features: the whole-image feature and the foreground feature. The whole-image feature (16 dimensions) is from a binary classification CNN to classify if an image is selected or not by human operators. The foreground feature (16 dimensions) is described in Section 4.3. We balance the number of positive and negative examples in training. For the main game, we choose the positive candidate view and one of the negative camera views at sampled times. For the auxiliary games, we randomly sample negative examples from the main game.

Data Imputation and Final Model Training In data imputation, we randomly sampled 4,000 frames around camera shot boundaries (within 2 seconds). The imputed data are verified by a model trained from the complete examples (about 2,100 data passed verification). We use the random forest method to fuse features from all candidate cameras since it is relatively easy to train. In the final model, about 6,000 examples are uniformly downsampled (1fps) from the main game. The dimension of the feature is $96 (16 \times 3 \times 2)$ for two types of features from three candidate cameras). The parameters of the random forest are: tree number 20, maximum depth 20 and minimum leaf node number 5. More details of the implementation are provided in the supplementary material.

6. Evaluation

We evaluate our method on the the USL dataset. To effectively use the data, we test on the main game using 3-fold

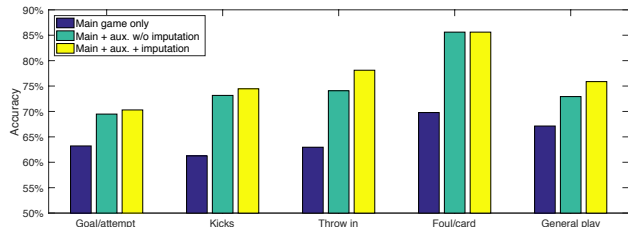


Figure 6: Prediction accuracy with and without auxiliary games (grouped by events).

leave-one-sequence-out cross-validation. We use the camera selection from human operators as the ground truth and report the classification accuracy. We also report the prediction accuracy on the dataset from [8] (the first 4 datasets in Table 1 are not publicly available).

For comparison, we also implement six baselines. **Baseline 1:** constantly select one camera with the highest prior in the training data. This baseline always selects the overview (middle) camera. **Baseline 2:** select the camera that is closest to the human-annotated grounded truth location of the ball. **Baseline 3:** predict the camera using the team occupancy map introduced in [4]. The team occupancy map describes the player distribution on the playing ground using tracked players from the static cameras. **Automated director assistant (ADA)** [6]: it learns a random forest classifier using player distribution and game flow at a time instance. Our implementation augments temporal information by concatenating the features in a 2-second sliding window, making the predictions more reliable. **C3D** [37]: it is a deep CNN modified from the original C3D network. First, images from three cameras pass through the original C3D network, separately. Then their f_{c6} activations are concatenated and fed into a fully connected network ($1024 \times 32 \times 3$) with *Softmax* loss. **RDT+CDF** [8]: it uses the recurrent decision tree (RDT) method and a cumulative distribution function (CDF) regularization to predict camera selections in a sequence. Because [8] requires real-valued labels in training, we only compare with it on the dataset from [8].

6.1. Main Results

Table 2 shows the main results of our method. First, auxiliary data provides significant performance improvement (about 9.4%). The improvement is from two stages: feature extraction and data imputation. Figure 6 shows details of the improvement by separating these two stages and grouping the frames into different events. Overall, the main improvement is from the feature extraction stage (about 6.6%). Data imputation provides an extra 2.8% improvement, which is significant in “throw in” and “general play”. Second, the foreground feature improves performance, especially when the auxiliary games are used. The main reason might be that the foreground feature excludes the

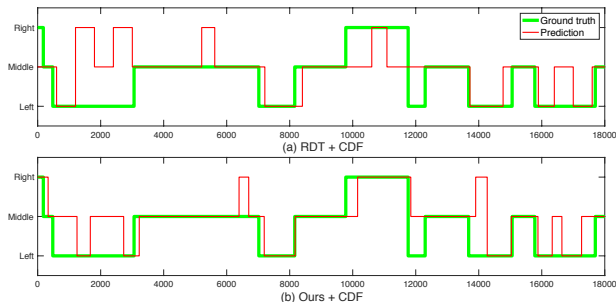


Figure 7: Camera selection on a 5-minute testing sequence. (a) Result of [8]. (b) Ours. The green line is from human operators and the red line is from algorithms. Best viewed in color.

negative impact of backgrounds (*e.g.* different fan groups, weather and light conditions) of auxiliary games.

Table 3 shows the comparison with the baselines. Our method outperforms all the methods by large margins. Baselines 1-3 do not work well on this dataset mainly because they omit the image content from the PTZ cameras. This result suggests that heuristic techniques using ball and player locations are not enough for dynamic PTZ camera selection. ADA [6] seems to have substantial challenges on this dataset. It is partially because of hand-crafted features (such as player flow) are quite noisy from fast-moving PTZ cameras. C3D [37] works reasonably well as it learns both appearance and motion features end-to-end. Its performance is slightly better than our whole-image-feature model. However, our full model is significantly more accurate (11.6 %) than C3D. It is worth noting that training C3D with auxiliary data is very difficult because the input of C3D is consecutive frames.

Combined with a Temporal Model To test the capability of our method with temporal models, we conducted experiments on the dataset from [8]. This dataset has 42 minutes (60 fps) data for training and a 5-min sequence for testing. In the experiment, we feed the selection probability to the cumulative distribution function (CDF) method from [8]. The CDF method prevents too short (brief glimpse) and too long (monotonous selection) camera selections. The experiment shows our method is more accurate than [8] (70% vs. 66%). Figure 7 shows a visualization of the camera selection. Video results are in the supplementary material for visual inspection.

6.2. Further Analysis

Data Imputation Accuracy Because the missing data in the auxiliary videos has no ground truth, we analyze the accuracy of our data imputation method using the main game data. We use the last 1, 100 frames as testing data by masking the features from the un-selected cameras as missing

Feature	Main				Main + aux.			
	L	M	R	All	L	M	R	All
whole-image	53.4	74.4	57.5	63.2	62.8	77.8	61.4	68.5
foreground	45.9	84.1	39.5	59.7	53.1	86.2	41.3	63.2
both	58.3	78.0	58.7	66.5	70.0	85.2	68.9	75.9

Table 2: Selection accuracy using different features and training data. “Main” and “Main+aux.” mean the training data is from the main game only and is with auxiliary videos, respectively. L, M and R represent the camera on the left, middle, and right side, respectively. The highest accuracy is highlighted by bold.

	Accuracy (%)	Δ
Constant selection	40.9	35.0
Closest to ball (GT.)	37.6	38.3
Team occupancy map [4]	49.8	26.1
ADA [6]	54.1	21.8
C3D [37]	64.3	11.6
Both feature w/ aux. (Ours)	75.9	–
Both feature w/o aux.	66.5	9.4
whole-image feature w/ aux.	68.5	7.4
foreground feature w/ aux.	63.2	12.7

Table 3: Comparison against various baselines and analysis of the effects of various components.

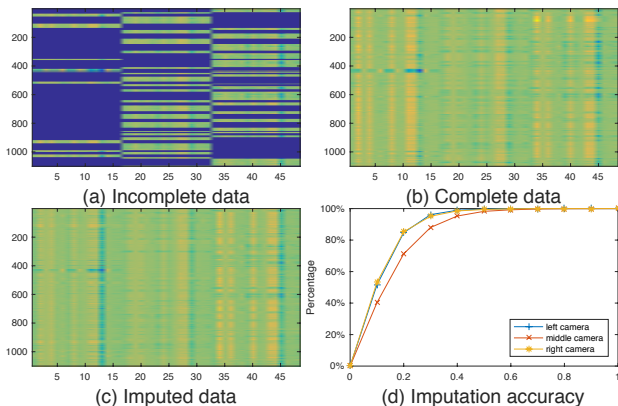


Figure 8: Imputation test on the main game data. (a),(b),(c) Color-coded visualization of imputed feature sequence. The x-axis is the feature dimension. The y-axis is the frame index. Colors visualize the feature values. Blue color blocks indicate the missing data. (d) The imputation accuracy as a function of the error thresholds. Best viewed in color.

data. A random survival forest model is trained from the rest of the data. The error is measured by absolute errors normalized by the range of feature in each dimension. This error metric is a good indication of the performance of imputed data in the final model. The final model (*i.e.* a random forest) uses the sign of the difference between the feature value and the decision boundary in internal nodes to guide the prediction. Figure 8(a)(b)(c) visualizes the incomplete

data, complete data and imputed data, respectively. The imputed data is visually similar to the ground truth. Figure 8 (d) shows the imputation accuracy as a function of the error thresholds. When the error threshold is 0.2, about 80% of the data are correctly predicted. Although the accuracy is tested on the main game, it suggests a reasonably good prediction on the auxiliary games.

To evaluate the performance of RSF on the real data, we also imputed the missing values using nearest neighbor (NN), OptSpace [25] and a neural autoencoder. Table 4 shows that RSF outperforms all of them with a safe margin.

Foreground Feature Aggregation To compare the performance of the SA heatmap with other alternatives, we conducted experiments on the main game data. All the methods use the same appearance feature from the siamese network as input. Table 5 shows that SA heatmap outperforms other alternatives mainly because it encodes both location and appearance information.

Qualitative Results Figure 9 shows predicted image sequences with the ground truth and contributing sequences. The contributing sequence is from the most dominant contributing examples in the leaf nodes for each prediction. Figure 9(b) (last column) shows an example of incorrect predictions. The ground truth camera is kept as the middle camera. Our prediction switches to the right camera. By inspecting the video, we found the human operator’s selection has better temporal consistency while ours tends to provide more information in single frames.

Discussion In real applications, more than three candidate cameras are used. However, we found most of the shots are from the three cameras that cover the left goal area, the middle field and the right goal area. We also qualitatively verified that the camera setting in the proposed dataset is representative for soccer games from [41] and [21]. It indicates that our method can be applied to many real situations, especially in small-budget broadcasting.

Although we collected the largest-ever training data from the Internet, the testing data is from one game. We mitigate this limitation by using dense testing (3-fold cross-

	Acc. (%)	Δ
RSF	75.9	–
NN	72.2	3.7
OptSpace [25]	68.6	7.3
Autoencoder	73.9	2.0

Table 4: Comparison of RSF with alternatives.

	loc.	appe.	Acc. (%)	Δ
SA heatmap	✓	✓	59.7	–
Avg pool.		✓	41.8	17.9
Max pool.		✓	42.4	17.3
Heatmap in [8]	✓		48.4	11.3

Table 5: Comparison of SA heatmap with alternatives.

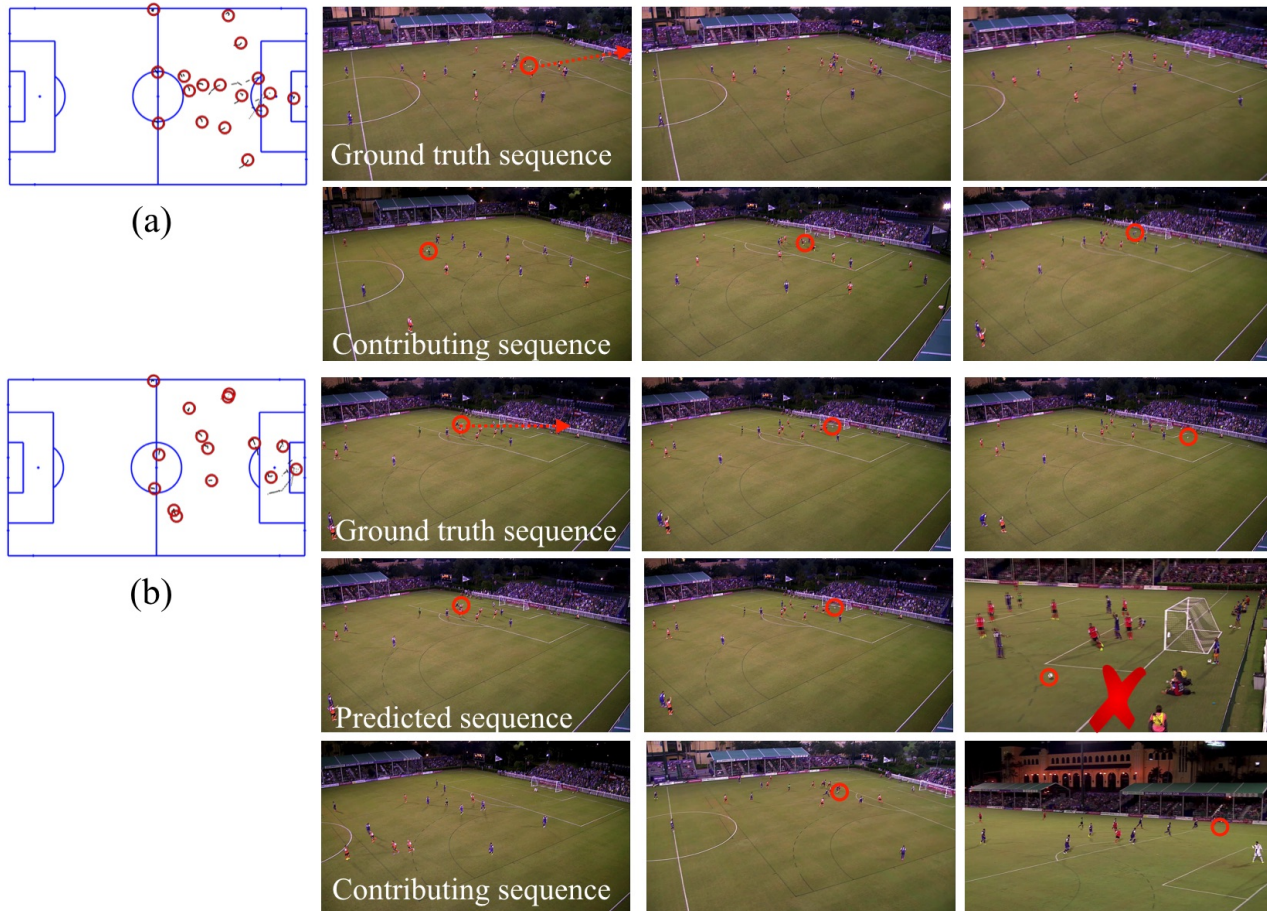


Figure 9: Qualitative results. The ground truth row shows the ground truth image sequences (about 3 seconds). The predicted sequence row shows the predictions from our method (omitted if all predictions are correct such as in (a)). The contributing sequence row shows the most dominant training example in the leaf node. In each sequence, player trajectories are visualized on the playing field template. The ball locations and their trajectories (dashed lines) are overlaid on the original images. The red cross mark indicates incorrect predictions. Best viewed in color.

validation). We leave large-scale camera selection as future work.

7. Summary

In this work, we proposed a framework for sports camera selection using Internet videos to address the data scarcity problem. With effective feature representation and data imputation, our method achieved the state-of-the-art performance on a challenging soccer dataset. Moreover, some

of our techniques such as foreground feature extraction are generic and can be applied to other applications. The proposed method mainly focuses on camera selection in single frames at the current stage. In the future, we would like to explore temporal information for camera selection.

Acknowledgements: This work was funded partially by the Natural Sciences and Engineering Research Council of Canada. We thank Peter Carr from Disney Research and Tian Qi Chen from University of Toronto for discussions.

References

- [1] At the gate and beyond: Outlook for the sports market in North America through 2021. Technical report, PricewaterhouseCoopers, United States, 2017.
- [2] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.
- [3] I. Arev, H. S. Park, Y. Sheikh, J. Hodgins, and A. Shamir. Automatic editing of footage from multiple social cameras. *ACM Transactions on Graphics (TOG)*, 33(4):81, 2014.
- [4] A. Bialkowski, P. Lucey, P. Carr, S. Denman, I. Matthews, and S. Sridharan. Recognising team activities from noisy data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [5] A. Bialkowski, P. Lucey, P. Carr, I. Matthews, S. Sridharan, and C. Fookes. Discovering team structures in soccer from spatiotemporal data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 28(10):2596–2605, 2016.
- [6] C. Chen, O. Wang, S. Heinzle, P. Carr, A. Smolic, and M. Gross. Computational sports broadcasting: Automated director assistance for live sports. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2013.
- [7] J. Chen, H. M. Le, P. Carr, Y. Yue, and J. J. Little. Learning online smooth predictors for realtime camera planning using recurrent decision trees. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [8] J. Chen, L. Meng, and J. J. Little. Camera selection for broadcasting soccer games. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [10] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2–3):81–227, 2012.
- [11] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.
- [12] F. Daniyal and A. Cavallaro. Multi-camera scheduling for video production. In *Conference for Visual Media Production (CVMP)*, 2011.
- [13] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing (TIP)*, 12(7):796–807, 2003.
- [14] C. K. Enders. *Applied missing data analysis*. Guilford Press, 2010.
- [15] P. Felsen, P. Agrawal, and J. Malik. What will happen next? forecasting player moves in sports videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [16] E. Foote, P. Carr, P. Lucey, Y. Sheikh, and I. Matthews. One-man-band: A touch screen interface for producing live multi-camera sports broadcasts. In *ACM International Conference on Multimedia (ACMMM)*, 2013.
- [17] K. Fujisawa, Y. Hirabe, H. Suwa, Y. Arakawa, and K. Yasumoto. Automatic content curation system for multiple live sport video streams. In *IEEE International Symposium on Multimedia (ISM)*, pages 541–546, 2015.
- [18] V. R. Gaddam, R. Langseth, H. K. Stensland, P. Gurdjos, V. Charvillat, C. Griwodz, D. Johansen, and P. Halvorsen. Be your own cameraman: Real-time support for zooming and panning into stored and live panoramic video. In *ACM Multimedia Systems Conference (MMSys)*, 2014.
- [19] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem. Soccer-net: A scalable dataset for action spotting in soccer videos. In *CVPR Workshop on Computer Vision in Sports*, 2018.
- [20] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [21] N. Homayounfar, S. Fidler, and R. Urtasun. Sports field localization via deep structured models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [22] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360° sports videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [23] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [24] H. Ishwaran, U. B. Kogalur, E. H. Blackstone, and M. S. Lauer. Random survival forests. *The Annals of Applied Statistics*, pages 841–860, 2008.
- [25] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.
- [26] F. Lefèvre, V. Bombardier, P. Charpentier, N. Krommenacker, and B. Petat. Automatic camera selection in the context of basketball game. In *International Conference on Image and Signal Processing (ICISP)*, 2018.
- [27] K. Lu, J. Chen, J. J. Little, and H. He. Lightweight convolutional neural networks for player detection and classification. *Computer Vision and Image Understanding (CVIU)*, 2018.
- [28] 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 (TPAMI)*, 35(7):1704–1716, 2013.
- [29] P. Lucey, A. Bialkowski, P. Carr, S. Morgan, I. Matthews, and Y. Sheikh. Representing and discovering adversarial team behaviors using player roles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [30] J. Owens. *Television sports production*. CRC Press, 2015.
- [31] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei. Detecting events and key actors in multi-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [32] Y.-C. Su and K. Grauman. Making 360° video watchable in 2D: Learning videography for click free viewing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [33] Y.-C. Su, D. Jayaraman, and K. Grauman. Pano2vid: Automatic cinematography for watching 360° videos. In *Asian Conference on Computer Vision (ACCV)*, 2016.
- [34] F. Tang and H. Ishwaran. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2017.
- [35] L. Tessens, M. Morbee, H. Aghajan, and W. Philips. Camera selection for tracking in distributed smart camera networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(2):23, 2014.
- [36] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding (CVIU)*, pages 3–18, 2017.
- [37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [38] H. C. Valdiviezo and S. Van Aelst. Tree-based prediction on incomplete data using imputation or surrogate decisions. *Information Sciences*, 311:163–181, 2015.
- [39] J. Wang, C. Xu, E. Chng, H. Lu, and Q. Tian. Automatic composition of broadcast sports video. *Multimedia Systems*, 14(4):179–193, 2008.
- [40] X. Wang, K. Hara, Y. Enokibori, T. Hirayama, and K. Mase. Personal multi-view viewpoint recommendation based on trajectory distribution of the viewing target. In *ACM Multimedia Conference (ACMMM)*, 2016.
- [41] X. Wang, Y. Muramatu, T. Hirayama, and K. Mase. Context-dependent viewpoint sequence recommendation system for multi-view video. In *IEEE International Symposium on Multimedia (ISM)*, 2014.
- [42] X. Wei, P. Lucey, S. Vidas, S. Morgan, and S. Sridharan. Forecasting events using an augmented hidden conditional random field. In *Asian Conference on Computer Vision (ACCV)*, 2014.
- [43] R. Yus, E. Mena, S. Ilarri, A. Illarramendi, and J. Bernad. MultiCAMBA: a system for selecting camera views in live broadcasting of sport events using a dynamic 3D model. *Multimedia Tools and Applications*, 74(11):4059–4090, 2015.
- [44] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.