Interactive Sports Analytics: An Intelligent Interface for Utilizing Trajectories for Interactive Sports Play Retrieval and Analytics

LONG SHA, Queensland University of Technology
PATRICK LUCEY, STATS
YISONG YUE, Caltech
XINYU WEI, JENNIFER HOBBS, and CHARLIE ROHLF, STATS
SRIDHA SRIDHARAN, Queensland University of Technology

Analytics in professional sports has experienced a dramatic growth in the last decade due to the wide deployment of player and ball tracking systems in team sports, such as basketball and soccer. With the massive amount of fine-grained data being generated, new data-points are being generated, which can shed light on player and team performance. However, due to the complexity of plays in continuous sports, these data-points often lack the specificity and context to enable meaningful retrieval and analytics. In this article, we present an intelligent human–computer interface that utilizes trajectories instead of words, which enables specific play retrieval in sports. Various techniques of alignment, templating, and hashing were utilized by our system and they are tailored to multi-agent scenario so that interactive speeds can be achieved. We conduct a user study to compare our method to the conventional keywords-based system and the results show that our method significantly improves the retrieval quality. We also show how our interface can be utilized for broadcast purposes, where a user can draw and interact with trajectories on a broadcast view using computer vision techniques. Additionally, we show that our method can also be used for interactive analytics of player performance, which enables the users to move players around and see how performance changes as a function of position and proximity to other players.

CCS Concepts: • **Information systems** → *Search interfaces*;

Additional Key Words and Phrases: Multiagent trajectories, alignment, sports analytics, fine-grain retrieval, camera calibration, interactive analytics

ACM Reference format:
Long Sha, Patrick Lucey, Yisong Yue, Xinyu Wei, Jennifer Hobbs, Charlie Rohlf, and Sridha Sridharan. 2018. Interactive Sports Analytics: An Intelligent Interface for Utilizing Trajectories for Interactive Sports Play Retrieval and Analytics. *ACM Trans. Comput.-Hum. Interact.* 25, 2, Article 13 (April 2018), 32 pages.
https://doi.org/10.1145/3185596

1 INTRODUCTION

In terms of sports data, the methods in which humans interact with computers are still quite primitive. In Figure 1(a), we illustrate the conventional method of interaction where a user types a text

Authors' addresses: L. Sha and S. Sridharan, Queensland University of Technology, 2 George St, Brisbane, Queensland, 4108, Australia; emails: sharon81818@gmail.com, s.sridharan@qut.edu.au; P. Lucey, X. Wei, J. Hobbs, and C. Rohlf, STATS LLC, 203 North LaSalle St. Suite 2200, Chicago, Illinois, 60601, USA; emails: {plucky, xwei, jhobbs, crohlf}@stats.com; Y. Yue, Caltech, 1200 East California Blvd, Pasadena, California, 91125, USA; email: yyue@caltech.edu.

https://doi.org/10.1145/3185596

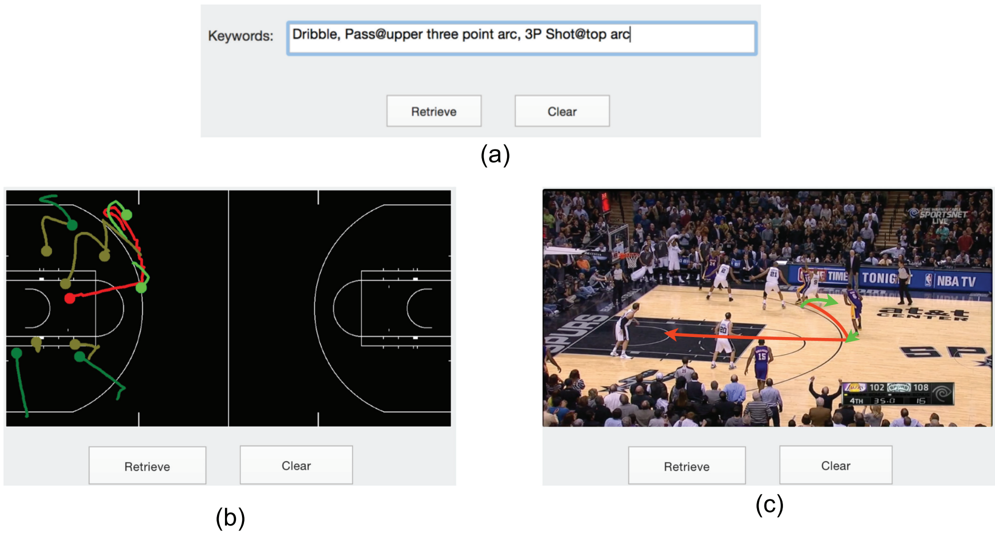


Fig. 1. (a) Conventional methods for sports play access are based on "keywords," which lack specificity and requires the user to browse through a large collection before finding the specific plays of interest. (b) To improve retrieval effectiveness, we instead use trajectories, which is an intuitive and powerful query format that allows the user to specify plays similar to how a coach/analyst draws up plays, and can capture the rich semantics required for effective sports play retrieval. (c) An alternative method of drawing up plays could also be on a broadcast view—we show how this can be done as well.

query into a search bar. In this example, it is a keyword query to retrieve video clips of plays in basketball that have been tagged with those labels. Another method includes that of pivot tables, where a human clicks on certain field(s) to analyze fields of data of a team/player that they are interested in analyzing.

However, with the wide-spread deployment of commercial tracking systems (e.g., STATS Sport VU [66]) in professional sports, which generates the x, y location of players and x, y, z of the ball at high-frame rates, a more granular form of data that can describe team and player behavior has been produced which these methods of interaction are of little use. This is despite the fact the tracking data is "visible" and can be easily understood by a domain expert in terms of trajectories.

For example, if we go back to the play retrieval example in Figure 1(a), describing the subtlety of movement of players via text labels would require an enormous amount of labels (i.e., a picture is worth a thousand words). Instead, it would be more intuitive to use the visual depiction using the raw trajectories of the play as the input query. Not only does this capture all the components of the play—it also invites interaction where the human can select the players of interest, or augment the trajectories to better reflect the query the user is interested in.

In this article, we showcase an intelligent interface for human–computer interaction, which allows for fine-grain search, as well as interactive team and player analytics for sports. To enable such use, we have developed a trajectory-based query language for sports plays. The intuition of such an approach is as follows: instead of searching plays using "keywords" (Figure 1(a)), we use a visual representation of player trajectories—inspired by the "x's and o's" a coach or analyst uses to draw up plays. Figure 1(b) depicts our basic setup: given an example play as the input query, the system retrieves a ranked list of plays ordered by relevance to the query. The query can be issued on a canvas, where a user can use an exemplar play or draw the query by hand. Alternatively, a user could potentially draw on the broadcast video and create a query by drawing the trajectories directly on the broadcast view (Figure 1(c)). This approach can also enable interactive analytics to occur, which relates to giving the user the ability to move players around in a given situation and see how the probabilities of events change based on the movement.

The implications of the work presented in this article are quite profound, especially when one considers the current workflow that exists in sporting organizations. In terms of searching for specific video clips in sports, this endeavor is one that is heavily humanized where experts spend up to two days to find similar video clips. In terms of analytics, there are currently no interactive methods of asking specific questions based on an example. For example, given a 3-point shot taken by a player in a given situation in basketball, the current method of applying analytics is to use an aggregate measure (i.e., his/her average 3-point is 42%, and 46% uncontested and 33% contested). Even though this is a reasonable approximate—it does not answer what is the specific probability in that situation. In this article, we show methods that can (i) search based on a specific exemplar or drawing, and (ii) answer a specific analytics question such as what is the likelihood of a player scoring in this situation. Such methods do not currently exist, and we believe our interactive method enables such specific questions to be answered.

The rest of the article is as follows. In Section 2 of the article, we present an overview of our approach for interface design for intelligent human–computer interaction. In Section 3, a description of the related work is documented and we explain how our work differs from the existing efforts. In Section 4, we describe our retrieval system, which includes how we form our trajectory-based language through trajectory alignment, as well as showing how we can draw on the broadcast video. This section also describes how we measure play similarity as well as hashing to allow for responsive retrieval. In Section 5, we validate our retrieval system using a variety of empirical evaluations, ranging from analyzing the accuracy/efficiency tradeoff of the various system components, to conducting a user study on the end-to-end task of sports play retrieval. This study demonstrates that our trajectory-based approach can achieve orders-of-magnitude improvements in search quality compared to systems based only on categorization, which showcases the efficacy of our approach and we discuss the findings. In Section 6, we showcase our interactive analytics method that allows users to ask specific questions as well as discuss the implications of this work. We then finalize the article with some concluding statements in Section 7.

2 OVERVIEW OF THE PROPOSED INTELLIGENT INTERACTIVE INTERFACE

We view our trajectory-based retrieval approach to being complementary to existing categorization methods. Much like how the early work on information access for the World Wide Web was largely dichotomized into directory/taxonomy vs. query/retrieval paradigms [70], we view our approach as the first instance of the query/retrieval paradigm for sports plays to complement the existing approaches for the directory/taxonomy paradigm. Note that the two types of approaches can be integrated together for improved performance (e.g., one can retrieve for only certain types of plays). Our interface can be categorized into the following two components: (i) *retrieval*, and (ii) *prediction*.

2.1 Retrieval

Below we list the types of fine-grain queries our retrieval interface can deal with (visual depictions are shown in Figure 2):

Exemplar-based query: A user can use a short play sequence from an existing game as the query (Figure 2(a)). The features include: (A1) a visualization window of play containing player and ball trajectories, (A2) a retrieval window that displays the most relevant plays, (A3) a retrieval button,

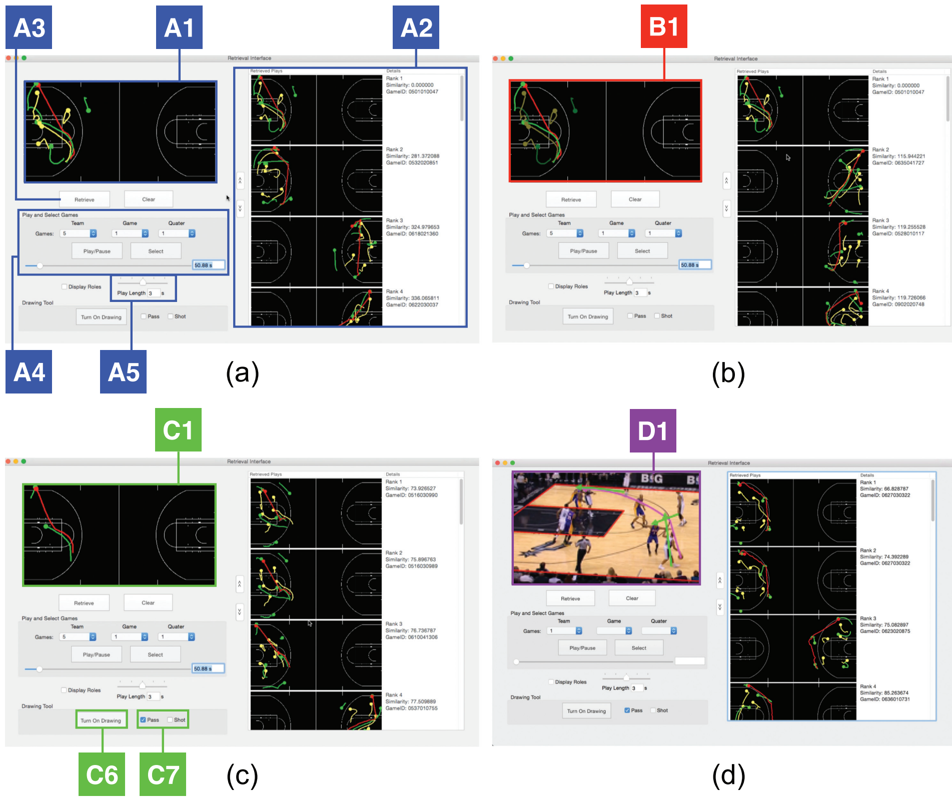


Fig. 2. Our Chalkboarding query language can take three different input types. (A) Exemplar-based where a user can play an existing game in the database and choose a short segment play as query—features include: (A1) Visualization window of play containing player and ball trajectories (red = ball, green = offensive-players, yellow = defensive-players, dots = end-of-trajectory), (A2) retrieval window that displays the most relevant plays in terms of rank, (A3) retrieval button, (A4) game-selection and scroll-bar to scrub through game of interest, and (A5) specify length of trajectories in seconds. (B) Manipulating exemplars where a user can select just a subset of trajectories which they deem relevant—it is the same interface except for (B1) the visualization window where the user can right-click to select the trajectories of interest. (C) Drawing plays where the user draws the play of interest like a coach would on a chalkboard—additional features include (C1) visualization window where the user draws the play of interest, (C6) draw tool selection button, (C7) toggle buttons to include actions in play (note, the user must also select the length of the play). (D) Drawing on a broadcasting video, (D1) is the window where user can play the video and draws the play of interest.

(A4) a game-selection and scroll-bar to scrub through game of interest until the portion of game is found, and (A5) a time-window selection. The last feature is quite important as the current system requires the user to define a time-window to the nearest second, and all searches will be constrained to look for trajectories fitting that time duration.

Manipulating exemplar-based queries: A user can also manipulate an existing play sequence before using it the query (Figure 2(b)). The primary manipulation we consider in this article is erasing irrelevant players (B1). More generally, one could also consider adjusting the trajectories of an existing play as well.

Drawing-based query: A user can draw a play of interest on the chalkboard (Figure 2(c)). Again, the interface is similar to the other two with the only difference being that the visualization

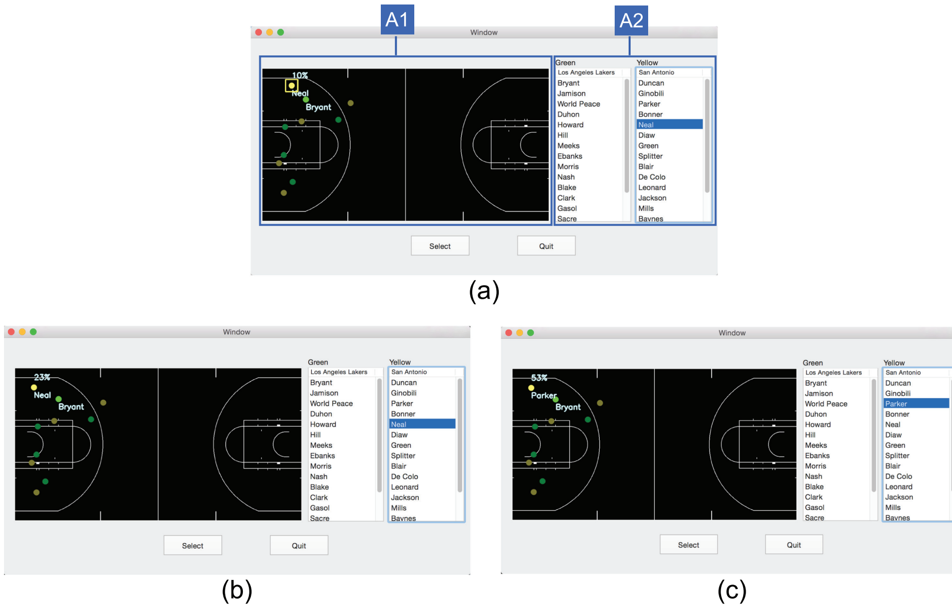


Fig. 3. The functionalities of our analytic module. A1 is the visualization panel that displays the estimated basket probability of offense players. A2 lists all the players in both team, user can swap them with any player on the interface. (a) Estimated point value of player Neil (yellow) in a particular play. The above value is the basket likelihood and the text below is his name. (b) User can move a player to see the probability change. (c) User can swap players to see the probability change.

window is now a canvas in which the user draws their input play (C1). To select the drawing feature, a “draw-tool” selection button needs to be enabled (C6) and to include semantics such as “pass” or “shot” onto the trajectories, we have a toggle button for those actions included (C7).

Drawing-based query on broadcast feed: Additionally, the user can also draw on a broadcast feed of a game (Figure 2(d)). The chalkboard window can also be used to play a broadcasting video and allows user to draw the play of interest on it (D1). To draw the play on the broadcast view may be a more intuitive way of interacting with the data. This requires an additional process of court calibration, which is not trivial as it centers on doing this on a moving camera. We dedicate an entire section describing how we do this (Section 5).

2.2 Interactive Prediction

Leveraging off the building blocks of our retrieval approach, we can also perform situational interactive analytics. Figure 3 showcases how we can achieve this through our interface. In this interface, the visualization panel (A1) shows the position of each player in the last frame of a play. A user can then click and drag any player on the court to see their basket probability (above player) and name (under player). A2 lists all the players in both team, once a player on court is selected, user can swap him with any other player in the list. Figure 3(a) shows the basket probability of player Neal in yellow team with a defender nearby. At this moment, Neal only has 13% likelihood to make a basket. But if we move Neal a bit left on the screen, the probability would increase to 27% (see Figure 3(b)). And if we replace Neal with Parker, the probability would jump to 43% (see Figure 3(c)). In this article, we show how we can calculate such probabilities “on-the-fly” based on the user interaction.

3 RELATED WORK

3.1 Multimedia Retrieval

The study of information retrieval enjoys a long and rich history in the information science and computer science communities [48, 62]. The overwhelming majority of previous work has been based on text-based or other types of tokenized query formats, which are unsuitable for multimedia data such as image, video, or spatiotemporal data in sports.

In order to retrieve multimedia data effectively, various query formats are developed depending on the granularity of search desired. In terms of image retrieval, exemplar-based or sketch-based query has been widely used [41, 60, 73, 74]. In these works, instead of using tags or attributes, a combination of edge-based features and lines are used as the input query. From this initial representation, feature descriptors or shape words are extracted to match the input query to the target image subset. Similarly, audio-based query enables users to accurately retrieve the desired music from a large database [7, 27, 65]. Another category of query uses hybrid formats, which combines content and semantic information. Users can use both textual query and visual query to improve the precision in retrieving multimedia data [19, 52, 59].

Even though the retrieval of certain types of multimedia data has been well investigated, there are very limited works in retrieving spatiotemporal data. The most commonly studied type of retrieval and recommender systems that incorporate spatial or temporal data are those that are "location-aware," i.e., use the location of users as semantic information to retrieve or recommend more relevant information [2, 4, 18, 68, 75]. Previous data access work in the sports domain have largely focused on the directory/taxonomy paradigm [70] via improving categorization of plays [5, 16, 24, 49]. The categorization process is normally expensive and retrieval using those keywords is also inefficient since the play of interest cannot be described explicitly. Some trajectory-based retrieval work lies in surveillance video retrieval [29, 31, 40], where trajectories are used to search for certain events. However, they only focus on single-agent trajectory while we are facing a more complex multi-agent scenario.

Our multi-agent scenario is similar to the setting in [67]. Team-based multi-agent data describes the behaviors of two or more agents involved in the same activity over a period of time, those agents are on the same team and share the same overall goals. Multi-agent trajectories consists of a spatio-temporal trace, which is a sequence of observations of the 2D position of each agent through time. Our Chalkboarding can be viewed as a new variant of content-based retrieval designed for multi-agent spatiotemporal trajectory domains, which is inspired by the sketch-based query format in image retrieval. It can also cooperate with semantic information to offer further benefit.

3.2 Computer-Human Interaction for Sports Data

In terms of human–machine interaction (HCI) for sports data, this has been a recent surge in work in this area thanks to the ease of capturing sports video, personal wearables, in addition to tracking system in professional sports [51, 54]. In terms of sports video, quite a number of works in the HCI community center on interacting with broadcast video. Since generating live broadcasts of sports normally requires a coordinated crew of people with different background, Perry et al. [56] boosted the team collaboration by analyzing video-mediated indexical gestures while Foote et al. [25] presented a unimodal interface concept that allows one person to operate multiple cameras. Apart from production, some works also focus on the broadcasting video augmentation. Yu et al. [76] proposed a new 3D camera calibration algorithm so that the virtual content (VC) can be inserted accurately in tennis match, while Liu et al. [44] presented a more generic VC insertion system that enables inserted VC can be noticed by audience at the right time without interrupting audience's viewing experience. This kind of work on broadcasting video heavily relies on the computer vision techniques and the target users are the video production teams and audience.

Another interesting area of research has focussed on assisting training in sports. The works in [38] and [55] describe methods to capture movement data with wearable devices for skateboarding and climbing. This compares to the work in [57] and [53], where both sets of authors presented novel ways to visualize tracking data so performance could be analyzed in a intuitive way. Altimira et al. [1] created two balancing techniques by using digital augmentation techniques to the sports field. By balancing the skill level of opponent, training can be conducted more effectively.

Although the above works are impressive, there is still an enormous amount of fertile ground for human–computer interaction in the sports domain when one considers all the knowledge discovery and data-mining research that is currently being conducted. Recent efforts have focussed on using machine learning algorithms to solve the problems such as developing advanced metrics to evaluate player/team performance [26] and analyzing team play patterns/styles [50, 78]. These new data points have shown that they can better reconstruct how a match is being played, as well as predict future performance. However, another use of these data-points is to elicit interaction from users to answer fine-grained questions that they may not have been previously able to answer. The goal of this article is to show how we can merge state-of-the-art machine learning on fine-grained into an interactive environment where a domain expert can draw further value from the fine-grain data collected.

Our approach bears affinity to other work on redesigning the interface between the human user and the data repository for various information retrieval and gathering tasks [8, 14, 21, 64]. Oftentimes, tuning system components such as the relevance estimation method results in relatively modest improvements in performance (cf. [77]), whereas designing a new interface to either accept richer inputs or produce richer outputs can lead to orders-or-magnitude improvements in overall system quality.

3.3 Representing and Understanding Multi-Agent Trajectories

From the technical perspective, the primary challenge that we study is how to compare the similarity between two multi-agent trajectories. There have been ample previous work studying how to measure similarity between trajectories and time series [17, 20, 69, 80], but they are largely focused on single trajectories rather than multi-agent ones. We build upon recent work that leverage a "role-based" representation [47, 71] that can compactly and efficiently characterize group behavior and formation in the sports domains.

The role representation is used primarily for alignment purposes and leaves open the question of what similarity measure to use for comparing individual pairs of aligned trajectories. One popular line of research focused on elastic measure that address warping and shifting effects in space and/or time [36, 37, 43, 61], while another line of research focused on finding the most similar or dissimilar points in two trajectories in order to ensure some notion of robustness (e.g., via the Hausdorff distance) [34, 45]. As the latter is designed to measure distance between polygons and does not take the direction of trajectories into consideration [80], in this article, we test some representative algorithms from the former category to find an ideal one for our task.

All modern retrieval systems require fast indexing in order to quickly search through a large repository, and one popular approach (which we also adopt) is to use a hash table [32, 79]. In different domains, the hash function is designed differently for specific applications. In general, the goal of hashing is to minimize time cost. Our approach is built on the concept of locality sensitive hashing (LSH) [28, 33], which is designed to place similar samples into a same address.

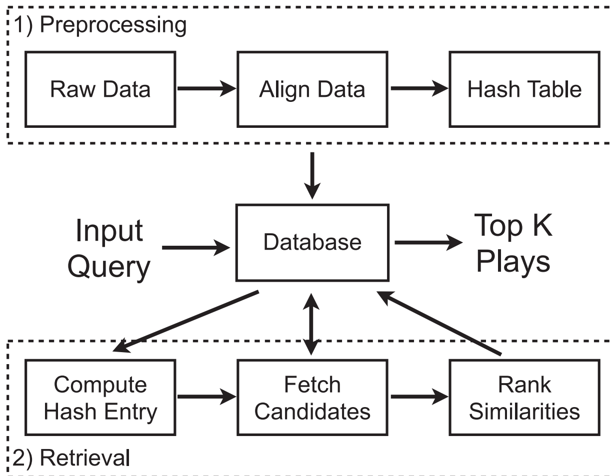


Fig. 4. Our retrieval system is divided into two parts, preprocessing and retrieval. The first part aligns the raw data and builds a hash table, whereas the second part contains a fast retrieval procedure. They are explained separately in this section.

Such an approach has also been applied in other settings where a similarity measure or ranking is required [35, 58].

4 TRAJECTORY-BASED RETRIEVAL SYSTEM

Figure 4 shows an overview of our retrieval system. To enable quick and accurate retrieval, "intelligent preprocessing" of the raw trajectory data is required. This preprocessing takes the form of using a role-based representation to enable fast alignment of trajectories, using a simple similarity measure to estimate relevance, followed by learning a hash-table to find the most relevant plays quickly.

After preprocessing, play retrieval comprises of first computing the hash entry from the query to find the most likely candidates, and then followed by local search to rank the candidate plays. Descriptions of these modules are given in the following subsections. Our full-stack system can retrieve relevant results from a repository of hundreds of thousands of plays in less than second, and can thus support interactive use cases.

4.1 Role Representation for Fast Alignment

To measure play similarity, we employ an agent-to-agent trajectory comparison method. The advantages of using such an approach are as follows: (i) the representation is lossless (i.e., no quantization is required), (ii) only a limited number of additional features are required to be stored (and not an overcomplete set of hand-crafted features that maybe hard to store in memory), (iii) the representation is visual and interpretable, and (iv) it allows for full interactivity with the data (i.e., users can select the precise time-window and agents of interest).

A drawback, however, is the problem of permutations. An example of this problem is shown in Figure 5(a), where we show the raw trajectories and the covariances of each player across a quarter of a match. Here, we can see players tend to move all around the court and not in one distinct area. In terms of matching trajectories, this means we would have to check all permutations, which in the case of basketball is $5! = 120$ (if we compare the trajectories of both teams it is 120^2).

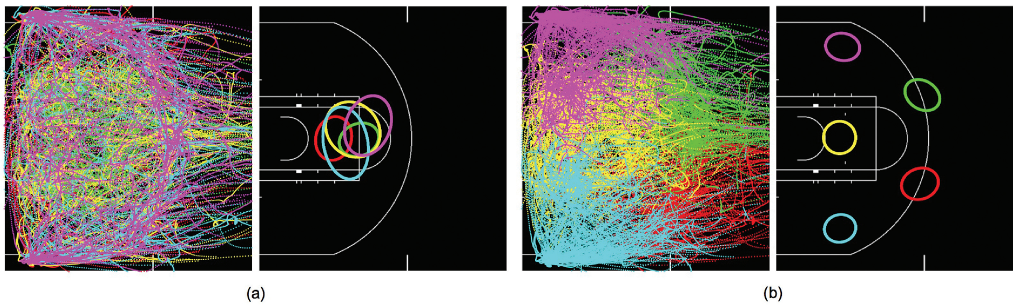


Fig. 5. (a) Shows the raw trajectories (left) and the covariance (right) of each player during a quarter of play, which shows that players can be located in any part of the court. (b) However, if we align the player positions to a formation template at each frame, we show that players are spaced into a formation.

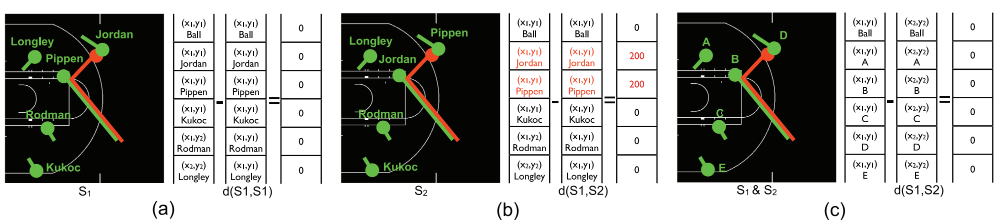


Fig. 6. A toy example of the data alignment problem. (a) S_1 is the play of interest, if we find an exactly identical play, the distance between them $d(S_1, S_1)$ should be zero. (b) However, if there is another same play S_2, but player Jordan and changed their roles, the distance $d(S_1, S_2)$ will no longer be zero due to the misalignment of these two players.

Although exhaustively comparing all trajectories would yield the optimal agent-to-agent alignment, recent research in multi-agent systems have shown that matching the trajectories to a pre-defined formation template can yield near-optimal agent-to-agent alignments [6, 47] in a single pass. We discuss the choice of distance metric in the next section on relevance estimation.

Figure 6 depicts a toy example that illustrates the intuition behind the formation-based approach [6, 47]. We can represent the play in Figure 6(a) as $S_1 = [\mathbf{s}_{\text{ball}}, S_{\text{offTeam}}]$, where each individual trajectory can be described as the vector $\mathbf{s} = [x_1, y_1, \ldots, x_F, y_F]^T$, and F is the number of frames in a play. Team behavior can then be described as the trajectories of the five players: $S_{\text{offTeam}} = [\mathbf{s}_{\text{Jordan}}, \mathbf{s}_{\text{Pippen}}, \mathbf{s}_{\text{Kukoc}}, \mathbf{s}_{\text{Longley}}, \mathbf{s}_{\text{Rodman}}]$. If the team runs exactly the same play at some other point in time, then those two plays will already be pre-aligned (i.e., each player runs the exact same trajectories in both plays), and the distance between them will be zero (for any distance metric).

If we instead compare S_1 with S_2 from Figure 6(b) using the same representation as S_1, then the distance will not be zero as Jordan and Pippen have switched positions. However, if we discard the identity of the player and use their role information at that frame – $S_{\text{offTeam}} = [\mathbf{s}_A, \mathbf{s}_B, \mathbf{s}_C, \mathbf{s}_D, \mathbf{s}_E]$, the distance between S_1 and S_2 both plays are seen as the same as the distance is zero (Figure 6(c)). In addition to normalizing for permutation, the role-representation is also robust to substitutions, and agnostic to team identity.

To learn the template for fast alignment, instead of manually labeling the roles of each player in every frame like [6, 47]. We compute the template from the data directly by using an unsupervised approach. We randomly collected a 10,000-frame subset from the dataset as training set and followed the steps below:

(1) Randomly select one frame from the training set and use the players' permutation in this frame as initial template T.
(2) Pick a frame f in the remaining training set and compute the cost of assigning each player in f to each role in template T based on their location. The cost here means the Euclidean distance between each player–role pair.
(3) Use Hungarian algorithm [39] to select the roles for each player and acquire aligned f^*.
(4) Use aligned f^* to update template T (via averaging).
(5) Repeat steps 2–4 until no frame remains in the training set.

Such approach generates a formation template (Figure 5(b) right), which is very similar to the template learned by manually labeling in [6, 47].

At each frame, we want to know which role within a formation a player is in. At the start of the match, we assign the player with an original role (or ordering in the feature vector). But as the game is dynamic players could have switched positions with their team-mate during a period during the match. So given a frame, and the initial ordering of players, we calculate the pairwise L_2 distance between the player's current position and each of the other positions within our role template. This gives us a square cost matrix, which is of size $N \times N$. Given this cost matrix, we then want to find the role assignment that has the lowest total cost between the position of players according to their initial ordering and the role template. Applying the Hungarian algorithm [39] to this cost matrix gives us the assignment/permutation with the lowest total cost. The assignment results in a single role feature being added to the raw position data of each player, which is basically just an extra column in the database.

We also run a sliding window (i.e., 1–5 seconds) and do a majority vote to determine which role that player was in for different time-windows (i.e., 1–5 seconds), which results in an additional five columns in the database of features, and is still very lightweight. In order to make all the plays comparable, we rotate all the plays that are conducted on the right side of court by 180 degrees so that the coordinates of all the plays are aligned. The reason why we did rotation rather than mirroring is that mirror did not adequately reflect the thoughts and wants of the experts as the handedness is a strong factor analyzing plays (i.e., a drive down the right side of the basket is far more common than the left-hand side as more players are right-handed dominant). We asked a number of users, and they indicated that they much preferred that mirroring was not incorporated as they felt that did not represent plays in the form in which they were run (i.e., if they searched for a play down the right and it returned them on the left they thought the retrieved results were wrong).

To show that our fast alignment method approximates the optimal alignment, we randomly selected 10 queries and chose the top 50 retrieved results. Figure 7 shows the plot of the average distance against the curve of three different alignment methods (see next subsection for metric distance used). The best performing alignment method was that of the exhaustive method. The next best was our fast-alignment using the role-representation, which was a close approximation to the optimal alignment. The worse performing alignment algorithm was that of identity, where the initial representation was fixed for the entire play.

Figure 8 shows the timing comparison, which reports the average time cost of 100 retrievals in an 100,000-sample database, and the time cost of each step is visualized. It shows that using role-based alignment for relevance estimation is 120-times faster than trying all permutation exhaustively in the ranking process for each time window size. This experiment was conducted on a 3.2GHz, 8GB RAM computer. It should be noted that only the attacking team is considered in this test. Had both teams (offense and defense) been involved in the retrieval task, the time cost difference would be

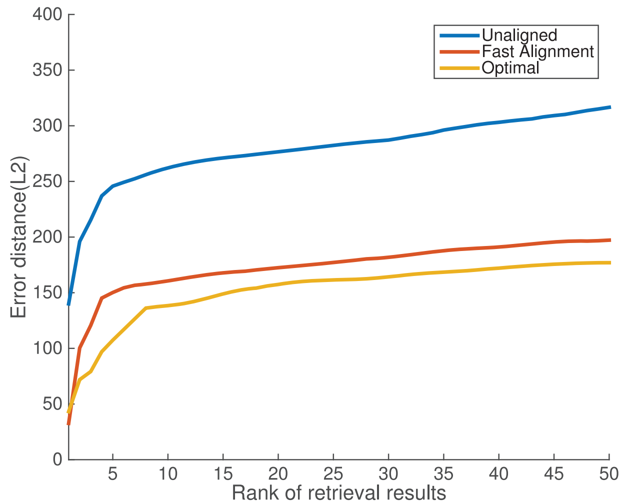


Fig. 7. The plot of similarity distance against retrieved-result ranks. It compares the performance of three alignment methods. Red curve is the optimal result, which is acquired by exhaustively searching the optimal permutation. It shows that our role-based representation (yellow) is much closer to the red curve than identity-based representation (blue), while our method also much faster than exhaustively permuting the data.

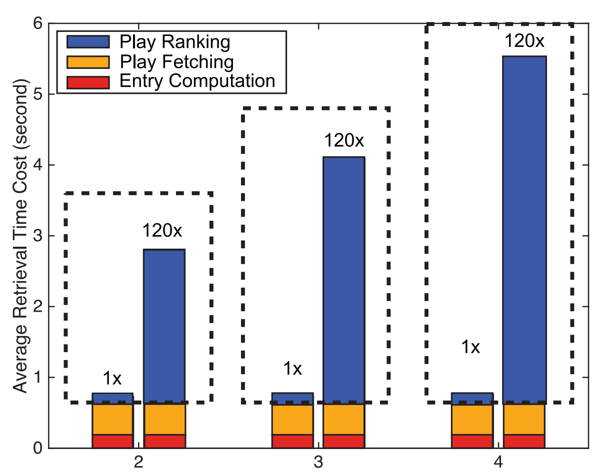


Fig. 8. The time cost of our role-based representation (left bar of each pair) and exhaustive permutation (right bar of each pair). It shows that role-based representation is 120-times faster than trying all permutation exhaustively in ranking for each time window size. The test uses an 100,000-sample database and only the attacking team is considered.

even more significant by another order of magnitude. The overall time complexity of our system will be explained later.

4.2 Relevance Estimation

Relevance estimation is the problem of determining which plays in the data repository is most relevant to an input query, and is typically addressed by producing a ranking of decreasing

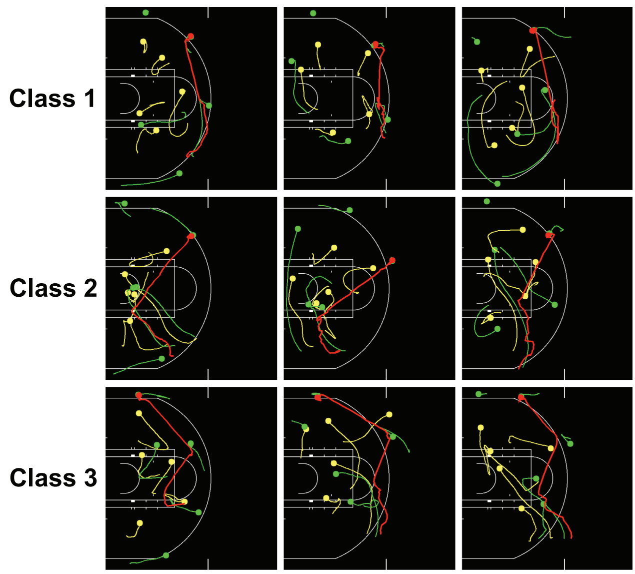


Fig. 9. This figure displays three class examples in our metric comparison experiment. We show three plays for each class. An expert annotated those examples based solely on the ball trajectories.

similarity to the input query [48, 62]. Given that our approach compares agent-to-agent trajectories, we need a suitable similarity metric to capture the differences between plays.

We use a supervised classification experiment to evaluate which distance measure is most suitable for measuring play similarity. We provided a domain expert a dataset that consists 1,600 3-seconds plays, and asked them to group the similar plays into one class based solely on the ball trajectories. Eventually, those 1,600 plays were divided into 38 classes, some examples are shown in Figure 9. We conducted the experiment via 5-fold cross validation and we compared six different metrics/measures on the aligned trajectories: l_2 distance, l_∞ distance, Frechet distance, dynamic time warping (DTW), longest common subsequence (LCSS), and edit distance (Edit). For LCSS and edit distance, we use the SAX [42] to transform the time series into symbolic representation. More precisely, it transforms each subsequence in the time series into a symbolic vector, so the whole time series becomes a concatenation of symbolic vectors, which can be thought as a big matrix with symbols. Since symbolic vector is computed locally from the subsequence, it can be computed incrementally. By contrast to using a single long symbol string to represent the time series, SAX is very robust to outliers and can be computed in a streaming fashion. Coupled with SAX representation, Edit distance and LCSS can be computed since they measure the similarity by counting the similar or dissimilar elements in the symbolic representation. Normally, SAX requires three parameters, length of subsequence, word length, and cardinality. In our case, we set the length of subsequence to half second. Word length and cardinality are empirically set to 18 and 10. We used nearest neighbor as our classifier.

Table 1 shows the classification accuracy. We can see that the first four metrics all worked reasonably well while LCSS and edit distance are much worse. Both LCSS and edit distance methods lose the direction and location information of trajectories, which is shown to be very critical in sport plays. To test if there were any significant differences between the first four metrics, we used the Mann–Whitney U significance test, which we chose as the number of examples per class were imbalanced. After running our significance tests, we found that there was no significant difference

Table 1. The Average Accuracy of Each Metric in 5-fold Cross Validation Experiment

	l_2	l_∞	DTW	Frechet	LCSS	Edit
ACC	0.773	0.755	0.787	0.741	0.430	0.412

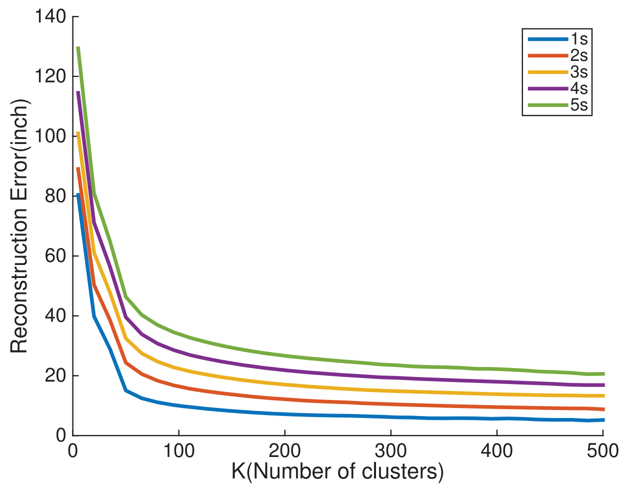


Fig. 10. The reconstruction error against the number of clusters for time window sizes from 1 to 5 seconds.

between these four. As such, we used the l_2 distance as our similarity measure as it is the easiest to deploy.

4.3 Hashing for Fast Indexing

In our dataset, we had over 600 hours worth of tracking data (~30GB), making similarity comparisons between a query and the entire database an expensive task. For example, if we were to break the database into a series of 3-seconds plays, this would equate to 4 million different examples. If each comparison took 30 milliseconds, searching the entire dataset would take 120 seconds – far too long to be usable as an interactive retrieval system.

The standard approach to improve retrieval speed is to use a hash-table or some other kind of indexing [48]. Based on domain experts' opinions as well as our own observations while building the initial version of the system, we used the ball trajectory as our index feature. To learn the dictionary of indexes, we applied k-means clustering using the l_2 distance on all ball trajectories for time-windows of 1, 2, 3, 4, and 5 seconds. The reason why we use 1–5 seconds time window is 2-fold. First, this is a heuristic that we used by talking to many expert basketball coaches. In terms of fine-grain behavior, we found through our interactions that the plays of interest were between the window of 1–5 seconds (pick-and-rolls, isolation plays, etc.). We were also motivated by the book [3], in one of the chapters, coach Stan Van Gundy stated that a play normally lasted 2–3 seconds and the sketches within this playbook were assumed to be of that length. The second issue is that the variance grows exponentially large as we expand the time-window, which means the number of "similar" plays retrieved was relatively small and not that useful for the user. To choose appropriate k for each time window, we inspected the reconstruction error plots (Figure 10), which we chose to be around 60 clusters. Examples of some of the clusters for the 3 seconds trajectories are shown in Figure 11.

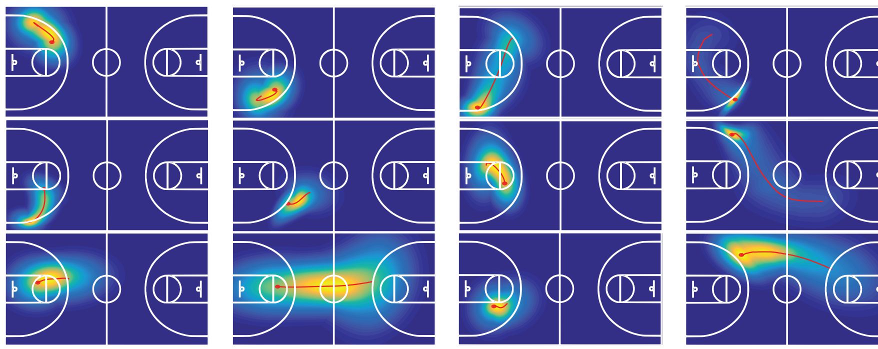


Fig. 11. Some examples of the ball trajectories distribution in each cluster with 3-seconds time window. Red curve is the mean trajectory and the small red circles are the end points.

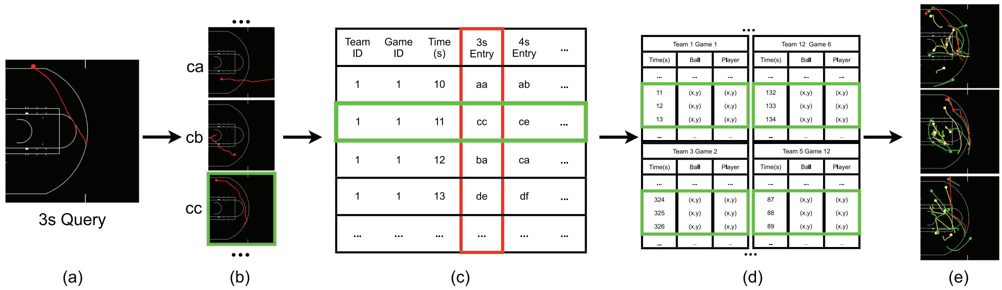


Fig. 12. An example of our retrieval process: (a) 3-seconds query with only ball trajectory, (b) compare to the centroid of each cluster, (c) acquire the plays with entry cc (cluster index) from hash table, only look at 3s entry (red column) in this case, (d) according to the game index, fetch the corresponding plays from raw tracking data, and (e) find the top K nearest neighbor. Green box indicates the selected candidates at each step.

In addition to using the spatial location, ball-actions (e.g., pass, dribble, shots) were incorporated by further splitting the clusters into semantic clusters. To ensure that each cluster contained less than 1,000 plays, we further divided clusters which had more than 1,000 plays into sub-clusters by applying another round of k-means on the specific cluster until this was achieved (k was chosen according to the size of plays within the current cluster—e.g., if a cluster had 2,000 plays, we used $k = 3$).

In terms of preprocessing, we use each frame as the starting point of the ball trajectory and obtain the ball trajectory for various time-windows (1–5 seconds). For each time-window, we then compare that trajectory to the centroid trajectory within the hash-table for the various time-windows. We then assign the index value of the closest centroid trajectory to that frame for each time-window.

4.4 Summary of Retrieval Process

Figure 12 depicts the end-to-end process of computing similarity measures of plays in our repository against an input query. Given an input query (Figure 12(a)), we first compare its ball trajectory against the centroids of every cluster (Figure 12(b)). In this case, cluster "cc" has the best match, and acquire the 3-seconds plays from that cluster using a hash table lookup (Figure 12(c) and (d)). Finally, we perform alignment and similarity matching between the query and the plays in the

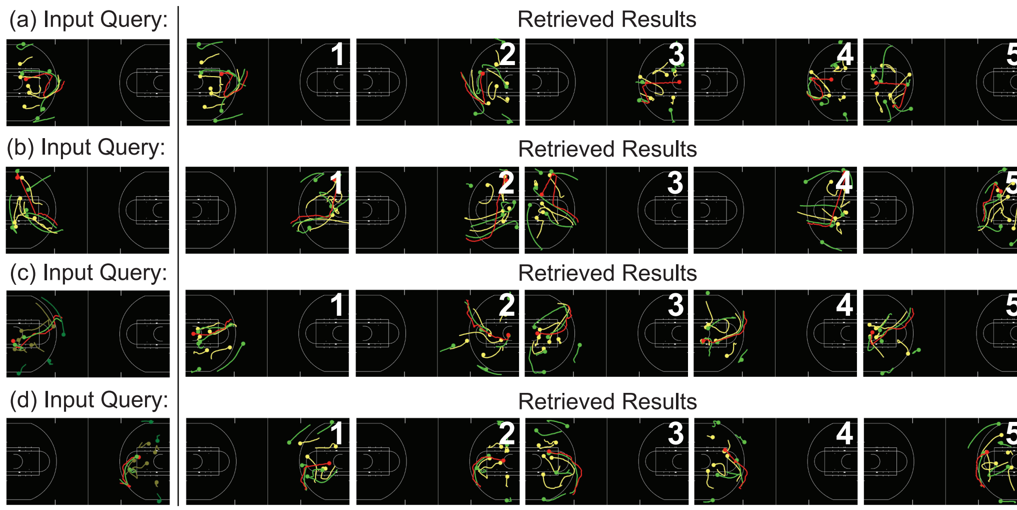


Fig. 13. Different input queries using the whole game-exemplars (Query (a) and (b)) and select only the trajectories of interest (Query (c) and (d)).

cluster, and return a ranking of the most similar plays. In terms of time complexity, we describe it as a function of the length n of query. The complexity of each step is listed below:

(1) The Euclidean distance calculation in the hashing process (Figure 12(b)) is linear to the length of array $O(n)$.
(2) The hash table lookup (Figure 12(c)) takes constant time.
(3) Fetching the candidates from database (Figure 12(d)) is linear to the length $O(n)$. We assume the number of candidates in each hash entry is similar.
(4) Aligning the query play is linear to the length $O(n)$ since aligning each frame takes constant time.
(5) The Euclidean distance calculation in the ranking process (Figure 12(e)) is linear to the length $O(n)$.

As each of these is (at most) linear in n, the whole system has a linear time complexity $O(n)$.

Note that our approach can continuously run in the background and produce increasingly better rankings of retrieved plays over time, e.g., by checking the next best cluster from Figure 12(b) (although the top retrieved plays tend to be from the best cluster and so does not change). Figures 13 and 14 show some retrieval examples of using different inputs.

4.5 Drawing Query on Broadcasting Video

Instead of retrieving plays using trajectories from a top-down viewpoint. Most coaches/analysts however, may prefer to see the actual video of the game instead of the tracking data. As such, it maybe preferable for the user to draw on the broadcast view. As we are still dealing with trajectories, this is achievable. However, the challenge is projecting or transforming the drawn trajectory on the broadcast view to the top-down viewpoint so comparisons can be made to our database. To do this, we have to calibrate the broadcast camera view to the top-down view.

4.5.1 Broadcast Camera-View Calibration. Camera-calibration is a fundamental task in computer vision. For our application, the calibration task can be defined as a homography problem.

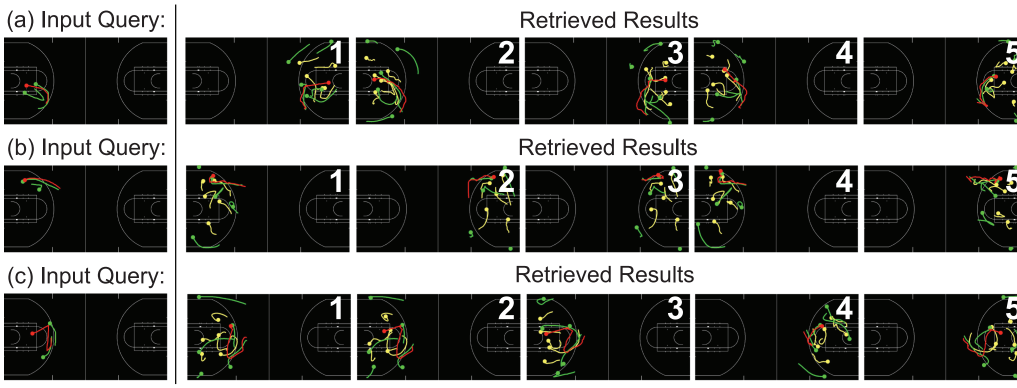


Fig. 14. Different input queries using user drawn plays.

As we can interpret the basketball court as a two-dimensional plane, this means that we need to compute the homography matrix $\mathbf{H}$ or the warping function $\mathcal{W}(p)$ in equation $X = \mathbf{H}x = \mathcal{W}(x, p)$ to project the position x on the floor of the court in the broadcast view to the top-down view X. A popular approach of doing this is via a template-based matching approach [15]. The intuition behind this approach is that we have a pre-defined template of a basketball court with the specific location of all the court markings known. We then map the detected landmarks on the broadcast view to the template. This method works quite well, but is susceptible when there are not enough landmarks in a frame (due to motion blur or occlusion). Instead of using landmark points, another method called *pointless calibration* [10], which uses all the edge gradients of a court to optimize projection parameters. The drawback of this method is that it requires an reasonable initial guess of parameters for optimization, it can be normally obtained from the previous frame, but if the camera moves rapidly, the parameters obtained from the previous frame would no longer be a good initial guess. Our calibration method combines both these two methods to obtain a robust calibration for every frame.

4.5.2 Implementation. The first step of our approach uses the template-based matching which is similar to [15] and [72]. In order to focus on the landmark matching on the court, we first need to acquire a clean view of the court in different perspectives so that the matching process will not affect by the moving objects. In our implementation, we collected 15 frames with different perspectives and manually annotated the landmark points on them. With the landmarks, all those frames can be projected to the real world coordinate, thus, every pixel on the court will have multiple values computed from different frames. Eventually, we acquire an relatively clean overhead view of the court by averaging the value of each pixel on the court. With the overhead view, clean views in original perspectives without players and audience can be generated as templates (see Figure 15). Then, homography matrix H is computed by using scale-invariant feature transform (SIFT) [46] and random sample consensus (RANSAC) [23] to match every frame to one of those templates and find the correspondence H.

However, the first step cannot correctly calibrate every frame in the video because of the occlusion or some edge blur in the generated overhead court image. A relative high threshold is set for the first step, and for those frames that fail to be calibrated, pointless calibration [10] is applied. One challenge of gradient-based image alignment is the narrow convergence of an edge image. In order to overcome the problem, this method computes a gradient at location x by fitting a plane to pixel intensities contained within a window centered at x. Thus, edges can have wider

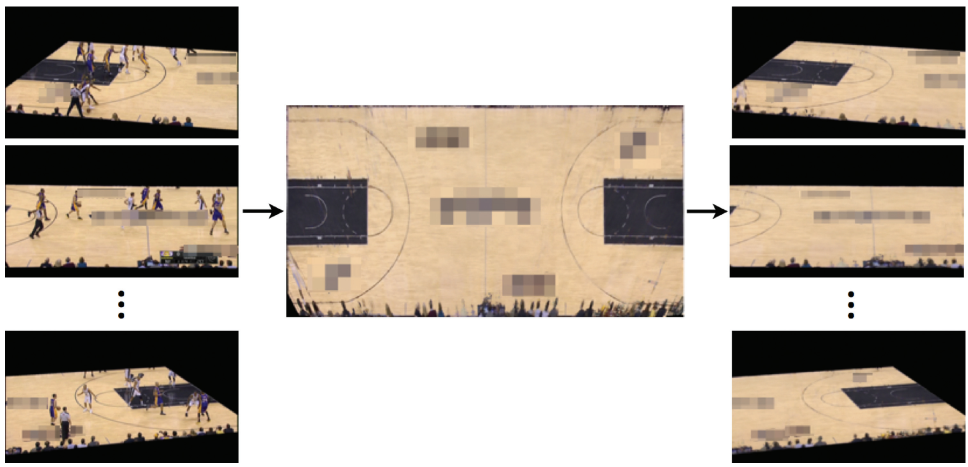


Fig. 15. Synthesize a clean overhead view through multiple frames and reconstruct court view in different perspectives (logos have been blurred out).

convergence, which is called long range gradient. Using the long range gradient, non-zero image difference can be computed between the projected location and the true location, which can be used to compute Δp in $\mathcal{W}$ for parameter optimization. For every frame that does not successfully calibrated by landmark, we use parameter H from the previous frame as the initial guess of current frame.

4.5.3 Broadcast Drawing Examples. For each match, the court calibration process described in the previous subsection needs to be conducted. This is because the broadcast camera view could be in a different position or vantage point due to the different venue or the camera has been moved. Once that is conducted, then the retrieval process described in Section 3 can be performed. In Figure 16, we show some examples of the drawn queries on the broadcast view. If we look at the first example, we show the user drawing on the broadcast view. Then, we show our transformation of the drawn trajectories of the broadcast view to the top-down view. As you can see these look like reasonable transformations. Given these transformed trajectories, we then show the retrieved results. In the next two examples, we show different types of plays with similarly retrieved plays. In practice, the user needs to stop the play at a specific frame, and then draw the trajectories on top of that frame.

5 USER STUDY EXPERIMENTS

Since our trajectory-based retrieval system is designed to help users to quickly find similar plays in a huge database, we validated our system via a user study on the end-to-end task of play retrieval. The goal of the user study was to compare the search quality of rankings generated by our trajectory-based approach vs. conventional keyword-based queries.

5.1 Baseline

The baseline system that we compare against is a keyword-based retrieval system. Because the variance associated with doing key-word search for such fine-grained and complex behaviors is extremely large and subjective, a means of normalizing against such an issue was to get an expert to adequately describe the play. In order to have the best key-word to describe the play, we asked a basketball expert to provide the keywords (action + coarse location) for each retrieval task, so

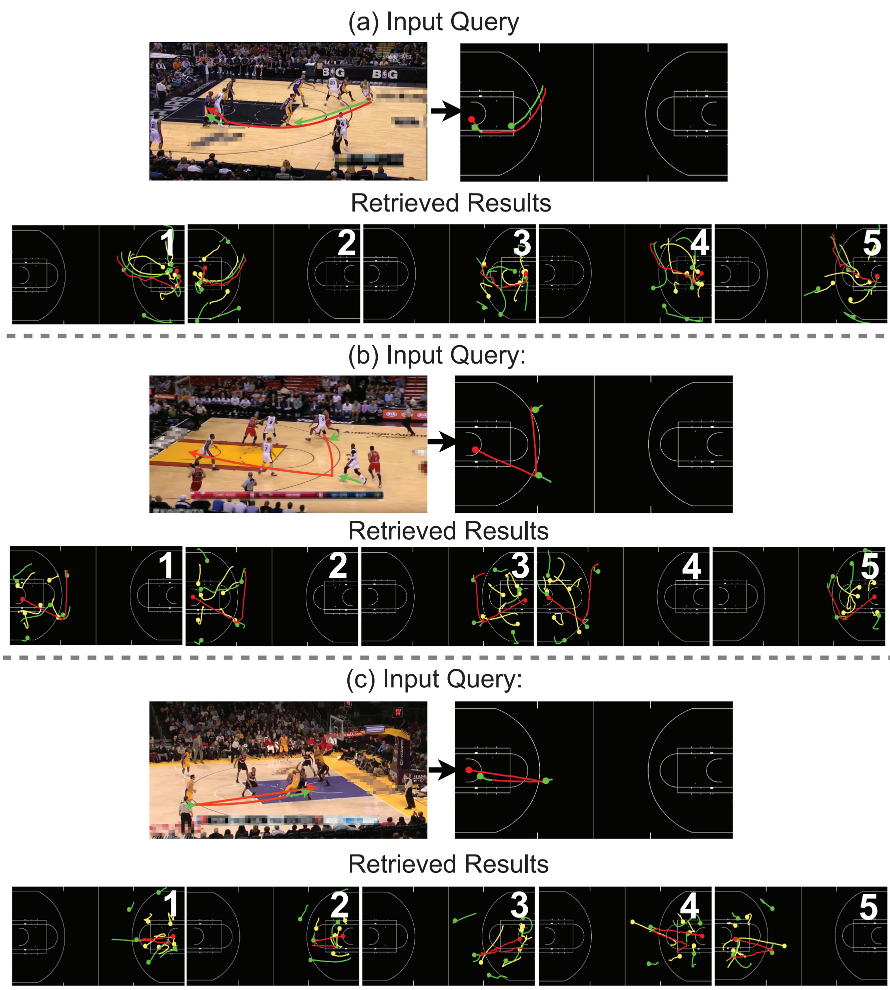


Fig. 16. Different input queries when drawing on broadcasting view. An additional step is required to project the trajectories to the chalkboard.

that users with little prior knowledge do not need to select the keyword themselves. Granted if the performance between key-word and chalkboarding was close, this would have been a big concern in terms of the comparability of two systems. However, as the users clearly much preferred using the chalkboard interface in the later section, we believe that our approach was adequate in comparing the two methods.

5.2 Experiment Design

A number of plays were showed to the expert that annotated the baseline method, and we selected eight retrieval tasks for our user study that are representative of basketball plays that are shown in Figure 17. For the trajectory-based approach, the first four plays are provided as exemplar-based queries which the users can use. For the remaining four queries, we only showed them the exemplar figures with selected trajectories and asked the users to draw the queries on the interface as a drawing-based query.

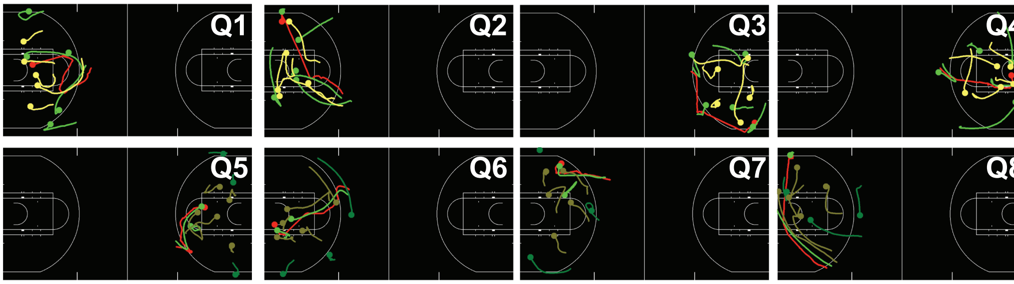


Fig. 17. Depicting the eight test queries for our user study. These queries cover a wide range of plays in competitive basketball.

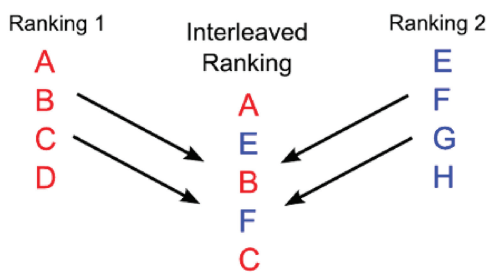


Fig. 18. Depicting an interleaving of two rankings.

We evaluated the retrieval quality via an interleaved evaluation, where the top 30 results from the trajectory-based and baseline systems combined via the Team-Draft Interleaving method [12] into a single ranking (see Figure 18). The combined ranking is then embedded in the retrieval window of our interface (pane B2 in Figure 1(b)), after which the user will scan the results top-down and select the relevant plays.

This setup offers two benefits: (i) it is a blind paired test that can control for user- and task-specific variability, which is often more reliable and sensitive than two-sample or A/B tests [12]; and (ii) by only presenting a single ranking, interleaving allows for evaluation to be conducted in a natural usage context.

5.3 Procedure

We recruited 10 volunteers with a wide range of basketball knowledge to participate in our user study. We first provided every volunteer with a 10 minute introduction to ensure they understood basic basketball concepts. In particular, we used a ninth retrieval task as a demonstration for how to recognize relevant plays. Every participant was allocated half an hour to perform all eight retrieval tasks. After the initial demonstration, all felt comfortable issuing trajectory-based queries.

Participants used our trajectory-based query interface to retrieve each query and the result panel showed interleaved results from both systems. They were asked to scan the results top-down and highlight retrieved plays that they think are relevant to the input query. For most users, it took about 10 minutes to acclimatize to the interface. After this time however, the average time it took to determine play relevance was around 15 seconds.

Table 2. Comparing the Mean Average Precision Aggregated Across All Users for Each Query

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Overall
Trajectory-Based	0.78	0.88	0.73	0.91	0.90	0.71	0.83	0.85	0.83
Keyword	0.06	0.17	0.24	0.09	0.06	0.13	0.03	0.18	0.12
Win / Lose	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	80 / 0

Note: Query 1–4 used exemplar-based queries in our system and 5–8 used drawing-based query. The "Win/Lose" row shows the number of users for whom the trajectory-based approach achieved a higher average precision. We see that our approach achieves orders-of-magnitude better retrieval quality.

5.4 Benchmark Results

Using the relevance judgments of the participants, we performed a benchmark comparison using two standard retrieval evaluation metrics: average precision [48, 62, 77] and expected reciprocal rank of the first result [13].

Let r_j denote the rank of the jth relevant document, then the average precision of a ranking can be defined as

$$\texttt{AvgPrec} = \frac{1}{\#rel} \sum_{j} Prec@r_j, \tag{1}$$

where $Prec@r_j$ is the precision of the top r_j items in the ranking (i.e., fraction of relevant results in the top r_j). The expected reciprocal rank is defined as

$$\mathsf{ERR} = \frac{1}{r_1}, \tag{2}$$

which is simply the inverse of the rank of the first relevant result. Average precision is more recall-focused (i.e., places more emphasis on the rank location of all relevant results), whereas expected reciprocal rank is more sensitive to initial search time to finding the first result. For our setting, we computed both of the evaluation measures on the two virtual ranking functions embedded in the interleaved ranking, and over the pooling of both top-30 results (i.e., pooling based retrieval evaluation [48]).

Table 2 shows the results for average precision. The top two rows of the tables show the mean average precision aggregated across all 10 users for each system. Recall that the first four retrieval tasks performed were using exemplar-based queries and the second four were using drawing-based queries. In both settings, our trajectory-based system substantially outperformed the baseline system. It should be noted that the difference in retrieval quality is very large for all queries.

The "Win/Lose" row in Table 2 shows the breakdown of how many individual users experienced higher average precision using our trajectory-based approach compared to the baseline system. We see that our proposed approach wins for every user on every retrieval task. Table 3 shows the results for expected reciprocal rank, and is structurally analogous to Table 2. We again see that our trajectory-based approach achieves significantly better performance compared to the baseline approach.

5.5 Subjective Evaluation

We also conducted two subjective evaluations. In the first evaluation, we showed participants both the keyword-based interface (i.e., Figure 1(a)) and the Chalkboard interface and asked them which interface they prefer to use for each retrieval task. All participants had previous experience with keyword-based search interfaces and naturally understood how to use our keyword-based interface. Table 4 shows the results. For most retrieval tasks, the participants unanimously preferred

Table 3. Comparing the Expected Reciprocal Rank of the First Relevant Result Across All Users for Each Query

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Overall
Trajectory-Based	0.83	0.85	0.63	1	0.95	0.82	0.95	0.90	0.86
Keyword	0.03	0.06	0.14	0.04	0.03	0.11	0.02	0.14	0.08
Win / Lose	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	10 / 0	80 / 0

Note: Comparing the average expected reciprocal rank aggregated across all users for each query. Query 1–4 used exemplar-based queries in our system and 5–8 used drawing-based query. The "Win/Lose" row shows the number of users for whom the trajectory-based method found a relevant result earlier in the ranking. We see that our proposed approach achieves orders-of-magnitude better retrieval quality.

Table 4. Showing Users' Reported Interface Preference for Each Query

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Overall
Trajectory-Based	10	9	7	10	10	9	10	8	73
Keyword	0	1	3	0	0	1	0	2	7

Note: We see that users overwhelmingly preferred our trajectory-based approach over the baseline system.

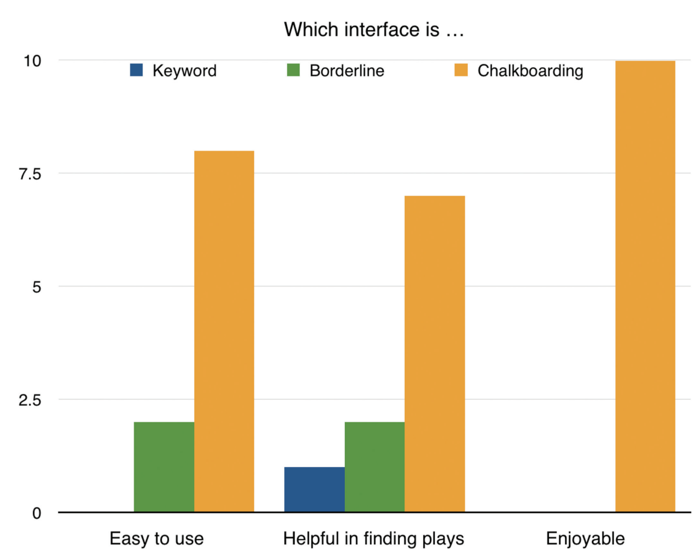


Fig. 19. Showing results of our subjective survey. We see that users largely preferred the trajectory-based system and found it easy and enjoyable to use.

using the trajectory-based retrieval system over the baseline system. Only for queries 2, 3, 6, and 8, some users still like to use the search bar because those plays were easy to describe.

We finally asked the participants to answer a short survey regarding their user experience, and the results are shown in Figure 19. All participants agreed that our system was more enjoyable to use because it had more functions as well as being more intuitive. In terms of effectiveness, two users with rich basketball knowledge thought the two systems were similar because they did not think using terminology to describe a play is more complex than drawing. But for other users, instead of using domain knowledge to describe a play, drawing a play was definitely easier to find what they want. Seven participants thought the trajectory-based system was more helpful for play retrieval, while the remaining three were neutral.

In our open-ended discussions about the system after the user study, the main criticism raised by the three participants who did not find our proposed trajectory-based retrieval method more helpful is that it is sometimes also important to acquire all the plays of one category. One suggestion was to have a hybrid system, where the user could use both the key-word and the trajectory-based approach for their input query. Another criticism was that the user had to assign a role to each player, which some felt was unnecessarily complicated.

5.6 Discussion of System Performance and Limitations

Both the benchmark results and the subjective evaluation show a significant performance gap between baseline and our method. This is an important issue and highlights the inadequacies of query methods available for fine-grain behaviors such as tracking data in sports. As the domain is so complex, and the data is extremely fine-grain, key-words are grossly inadequate for searching such data/behavior. Using a visual yet interpretable query language such as ours presented, gets closer to what the user is actually wanting to retrieve (i.e., a picture tells a thousand words). The current keyword-based search engine for sports play is basically a categorization system, it divides the plays into a number of groups based on their tags. For example, if we used the keyword "3-point shot, left wing," it would only return all the plays that contain a 3-point shot made at left wing, but ignore players' movement before and after the shot. Technically, keyword-based system is not designed for fine-grained play retrieval and that is the reason why it performs much worse than our method. However, some plays can be easily described with text. An good example is the Query 3 in our user study, which can be described explicitly as a swing pass. By using this keyword, the baseline method can also give reasonable result, which somehow reduces the average precision of our method.

In a sense, our approach can be thought of as the simplest approach that can support effective sports play retrieval, and motivates many directions for future work. For instance, we can incorporate more sophisticated categorization techniques in order to further refine our query language (i.e., further combine the query/retrieval and directory/taxonomy paradigms). From an interface design standpoint, one obvious area for improvement is spatial manipulation of existing or predrawn trajectories (i.e., slightly translating or rotating an existing trajectory).

There are other important information access tasks beyond retrieval, most notably including summarization and personalized curation [14, 64]. A typical interface for the summarization setting is to construct clusters of results and describe the relationships between clusters. Chalkboarding can also be used to improve categorization systems by creating exemplars for clustering purposes.

From a system design standpoint, it would be highly beneficial to further improve retrieval speeds. For instance, many applications designed on top of web search require issuing multiple search queries, which is impractical given our current system. It is possible that using a hierarchical hashing/indexing technique and/or coresets [22, 30] can substantially improve retrieval speed.

From a relevance estimation standpoint, our choice of distance measures can be improved. For instance, one of the early breakthroughs in conventional information retrieval is the concept of "inverse document frequency" [62], which essentially states that tokens that appear in many documents in the corpus (e.g., stopwords such as "the") are not as indicative of relevance more rare words. It would be interesting to incorporate this concept into our trajectory distance measures. More generally, one can consider applying machine learning to develop even more accurate measures given appropriate training data [9, 77].

Another interesting direction is to incorporate more spatial regularities into the query processing. For example, understanding spatial equivalence classes [50, 78] can enable the retrieval system to understand that the left corner 3-point shot is (almost) equivalent to the right corner 3-point shot, which can further improve the accuracy of the system.

6 INTERACTIVE ANALYTICS

In the previous two sections, we have showcased an interactive method of using an exemplar play—whether that be an actual play, augmented play or drawn play represented by the raw trajectories—and using as an input query to answer the specific question, *find me all the plays like that?* Given that we can do that using our method, this this opens up a plethora of other potential questions the user can ask. In this section, we show how we can use an exemplar play to answer a specific analytic questions such as, *what is the likelihood a player will score in that situation?*, *what if I switch that player with another player?*, and/or *how successful is that team running that play?* We call this "interactive analytics" and we showcase how we can do that for both player and team analysis.

6.1 Interactive Player Analytics

Apart from play retrieval, estimating the likelihood that a player will make a basket in a specific situation would be useful for experts, this likelihood is called expected point value (ExP) in professional sports analytics. ExP is a value between 0 and 1 that measures the likelihood of making a shot and it is influenced by many factors such as shooter's location, defender's location, and so on. Although there are some methods [11] that can compute ExP for domain experts, the existing solution requires programming skills to use those tools, whereas by using our interface, users can easily view the ExP in a more visual way. In addition, they can specify a situation by simply dragging and clicking on the interface so that the outcome difference between two situations can be observed.

6.1.1 Model Learning. In order to enable shooting analysis, an accurate and efficient model of ExP prediction is required. To do so, we grouped shots into eight classes based on locations, recent actions (e.g., dribble vs. pass vs. offensive rebound), and possession-states (transition vs. half-court set). Grouping shots in this manner enables us to linearize the data, and then fit a model to the shots in each class. Creating classes of shots could be achieved through a variety of unsupervised techniques, but here we elect to group shots into eight classes by leveraging domain-knowledge.

At the highest level, we first divide shots into those taken in-transition (i.e., on a fast break) and those taken in the half-court set. A shot is classified as in-transition if it occurs within the first 6 seconds of possession and either the ball or the fastest offensive player is traveling down-court at greater than 11ft/second. Transition shots can further be classified as at-the-rim (i.e., within 10ft of the basket) or not. Shots taken in the half-court set are broken into 2-point vs. 3-point shots. Three-point shots are classified as either catch-and-shoot or normal shot. A catch-and-shoot shot is defined as one during which the player shot the ball without dribbling and within 2 seconds of receiving a pass. Two-point shots are separated as drives, points-in-the-paint, catch-and-shoot and normal shot. Points-in-the-paint are classified as either put-backs (occurring in the paint after an offensive-rebound) or other.

We treat ExP as a conditional expectation:

$$\mathrm{ExP} = P(\mathrm{points}|\phi(\mathrm{S})), \tag{3}$$

where S is the spatial locations of all players in the given frame and $\phi(\mathrm{S})$ is a set of high-level features that are computed from the input frame. $\phi()$ crafts a myriad of global features for each player:

$$\phi(S) = \left(\ldots, x^m, y^m, v^m, d^m_{hoop}, d^m_{def}, \theta^m_{hoop}, \theta^m_{def}, p^m, \ldots\right), \tag{4}$$

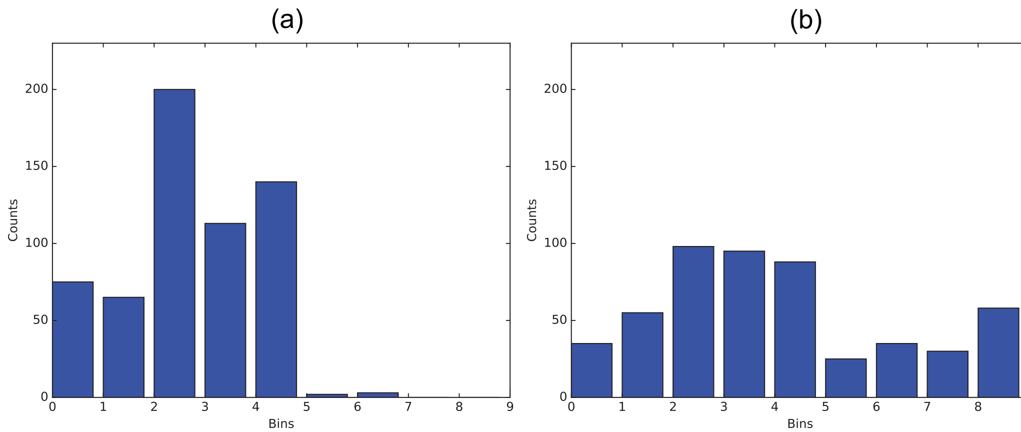


Fig. 20. Personal histograms of two players. Each bin represents a particular play cluster. Roughly speaking, bin 1–5 contain the plays near the basket, whereas 6–9 contains the plays that are far from the basket. The left one is a player who focuses on 2-point shot, whereas the right one is a well-rounded player.

where x^m, y^m are the location player m on the court, v is the velocity, d_{hoop} is the distance to the hoop, d_{def} is the distance to the closest defender, θ_{hoop} is the angle relative to the hoop, θ_{def} is the angle to the closest defender, and p is the overall average basket probability at this location. Every shot example is assigned into its appropriate class as described above. The same set of features are crafted for each shot-class and the ExP ground truth of each example is 0 when it is a miss and 1 if it is a goal. A logistic regressor is fit for each of the eight shot classes.

The model we have trained is a set of global models that predicts what the average player would do in each situation. However, due to the various attributes (both physical and skill/talent wise), there will be a great difference in the execution rate. As we will never have enough data to model all the specific interactions (i.e., player location, other players on the court and their positions and motions), we need to do some type of latent factor modeling/collaborative filtering. In our Chalkboarding, we have aligned the multi-agent trajectory data and learned a hash table to partition the whole database. The hash table can be thought as a playbook that contains all kinds of play styles, which we can used to achieve personalization in ExP prediction.

In our implementation, we use some high-level nodes (around 10) that are only learned from shots in our hash-table as a latent factor H. H contains the normalized frequency count of how often a player participates in each play style and their execution rate. By concatenating $\phi()$ and H, Equation (3) can be re-written as

$$\text{ExP} = P(\text{points}|\phi(\text{S}), H). \tag{5}$$

Figure 20 shows the histogram of two players, where player (a) normally makes 2-point shot (center) while player (b) is a well-rounded shooter (shooting guard). Bins 0–9 represent 10 different clusters of plays, after manually inspecting the plays in each cluster, we found out that bins 1–5 contain the plays near the basket, whereas bins 6–9 contain the plays that are far from the basket. Please note that each bin among 6–9 contains both 2-point shots and 3-point shots, so 2-point shots and 3-point shots are not separated in those bins.

6.1.2 Interaction Procedure. Even though logistic regression is a very efficient learning algorithm, some information still needs to be computed and stored in advance to ensure interactive feedback. For all the shots in our database, their ExPs can be pre-computed and stored at each

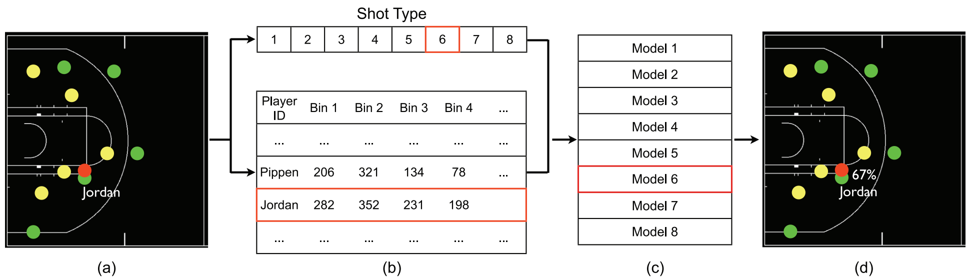


Fig. 21. (a) and (b) are two analytic examples. The first row is the original basket probability of a player in a particular scenario. The second row shows how its probability change when we drag the player to a better position. The third row shows how the probability increase when we move the closest defender away. The last row shows probability change when we swap player's identity.

Table 5. Basket Prediction Evaluation by Using Two Metrics

	RSME	Log loss
Global feature	0.51	0.67
Global feature + personal histogram	0.45	0.61

Note: Using the personalized histogram slightly improves the performance.

frame so that the interface can display the ExP for any existing game without any computation using the models. However, in order to have fast response for those shots that are modified by users, the histograms of all the players need to be pre-computed and stored as well. A new player table is created to store the players' information.

Figure 21 summarizes the procedure of computing the ExP for a situation. In this example, the input situation is that Jordan is making a 2-point shot close to the free-throw line. Given this input, we first find its shot type based on the spatial information. Since the input is a modified version of an exemplar, the velocity and the actions in its original play are also considered in shot type classification. But if the input frame was drawn by the user, we would assume they are normal shots because the velocity and the previous actions are unknown. In Figure 21, this shot is classified to type 6. Meanwhile, we find the personal histogram of Jordan from the player table in our database. After combining the spatial feature $\phi()$ and the personal histogram, it is fed into the model that is associated to shot type 6. At last, the model returns the ExP for this situation. Every time when users drag a player or change the identity of a player. this process will repeat to update the ExP value on the interface.

6.1.3 Prediction Evaluation. In order to compare the ExP prediction accuracy with and without personalization using our hash table, we conducted a quantitative experiment with a shot dataset. There are totally 200,000 shots in our dataset (46% are basket and the rest of them are miss), after assigning each shot to a class, each class has over 20,000 examples. In each class, 80% of examples are used for training and the rest 30% are used for testing. For each sample, our system can output a probability of basket. We compute the root mean square error (RMSE) and log loss averaged across all the classes to evaluate the prediction. The prediction performance is shown in Table 5.

It can be found that personal histogram can somehow improve the result but not very significant in this case. This is caused by the simplified hash table we learned for personalization. Due to the

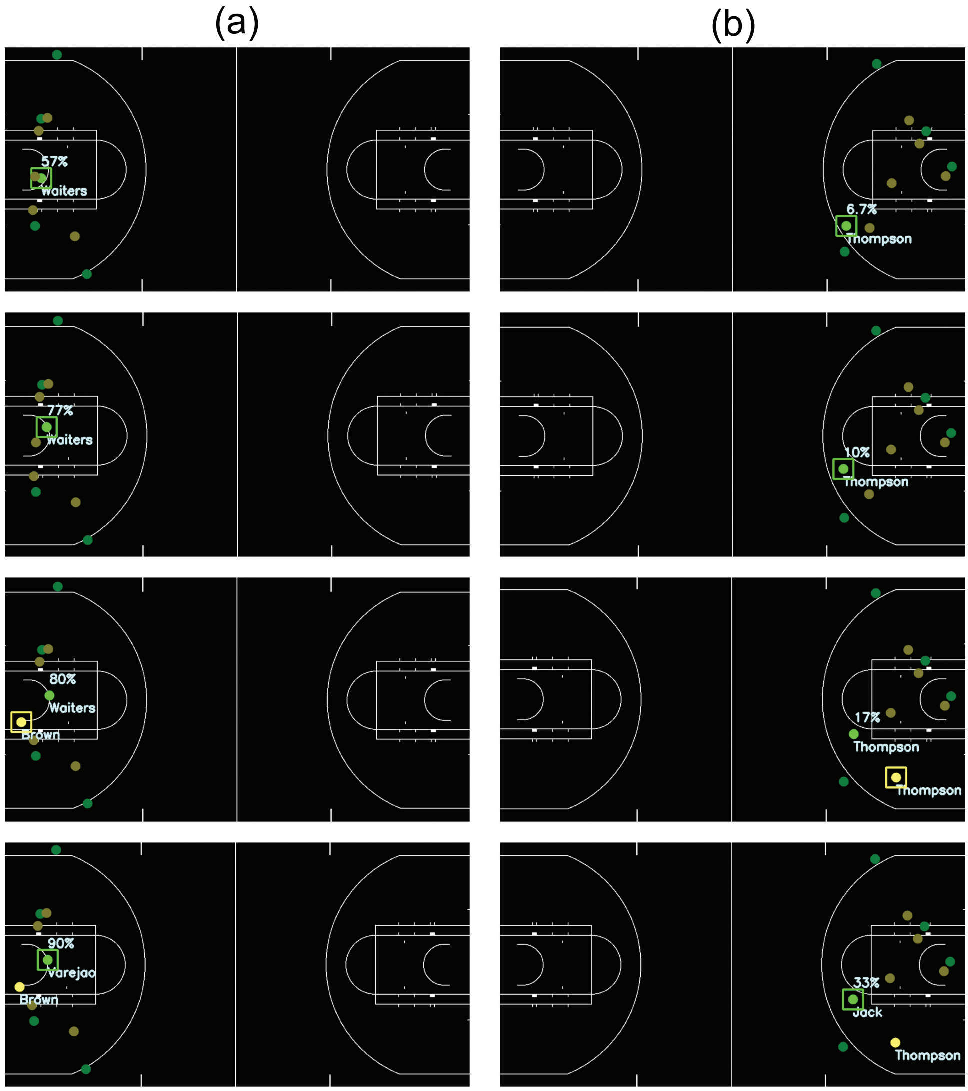
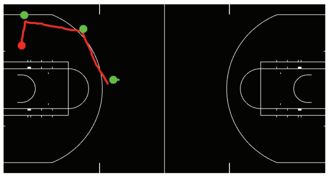


Fig. 22. The summary of player analytics procedure. Given a shooting situation (a), we first classify the shot type by using the spatial information, at meanwhile, we search the personal histogram of the target player in our database (b), after spatial features are computed, the corresponding model is used to predict the ExP (c). At last, the value is displayed on the interface (d).

limited data, this small hash table is very coarse, which cannot provide enough information for sufficient semantics, but we can still see the evident improve after adding the personalized factors.

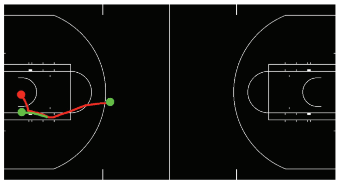
Figure 22 shows some examples on how it can be utilized. In Figure 22(a-top row), we show a specific example where Waiters is closely guarded near the basket. Based on historical data, he is likely to make 57% of the shots in that situation. However, if we move him a slightly better position, his probability increase up to 77% (Figure 22(a-2nd row)). If we move the defender further, his probability again increases to 80% (Figure 22(a-3rd row)). Now if we were to replace Waiters with

(a)



Team Name	#Total occurrence	Average per game	Average in quarter 1	Average in quarter 2	Average in quarter 3	Average in quarter 4	Success rate
New York	61	1.36	0.29	0.22	0.36	0.49	34.00%
Cleveland	31	0.84	0.24	0.24	0.14	0.26	48.00%

(b)



Team Name	#Total occurrence	Average per game	Average in quarter 1	Average in quarter 2	Average in quarter 3	Average in quarter 4	Success rate
Milwaukee	649	15.10	3.28	3.95	4.19	3.67	42.00%
Dallas	564	14.10	3.45	3.60	3.58	3.48	45.00%

Fig. 23. The team analytics based on our retrieval results. (a) and (b) shows two different queries. (a) is a "swing pass" ends up with a shot around 3-point-line and (b) is a common inside play. The tables below show how many times they were performed by two teams, the frequency they occurred in each game and each quarter and the overall success rate.

Varejeo, we can see that Varejeo is 90% likely to make that shot in that situation—10% more than Waiters (Figure 22(a-4th row)). We show an example in Figure 22(b) with a similar functionality.

6.2 Interactive Team Analytics

In addition to player analytics, team-based analytics also becomes achievable with the retrieval system we have built. The current interactive framework allows users to ask the question "how often does this team perform this play and is it effective?" Figure 23 shows examples given two different input queries. In Figure 23(a), a drawn play is shown that depicts a "swing pass," which ends up with a jump-shot in the right-hand wing. By using the searching system, instead of returning the most similar play, we can use all the candidates in the database to summarize the frequency and the effectiveness of the input play for a specific team. The table underneath the input query shows that New York had a higher frequency of performing such play (61 times in the season), while Cleveland utilized the play less, they had a higher success rate (48% vs. 34%). For the New York, they were more likely to perform the play in the fourth quarter.

Figure 23(b) shows another drawn play where a ball is quickly passed to a player cutting toward the left-hand-side of the rim. This seems to be a very frequent play, and we show that both Milwaukee and Dallas utilize it around 15 times per game with similar success rate. Even though we only showed two plays here, this type of analytics can be applied to any input play (drawn or exemplar). Additionally, it would be useful to see which players are effective in these types of situations. To do this, we have to personalize the analysis that we show in the next section.

6.3 Discussion of Limitations of Interactive Analytics

The player and team analytics discussed above shows some new interactions that we can build on top of our interface. Since visualized trajectories have replaced the keywords or tags in the interaction procedure, users are able to ask specific questions and receive interactive feedback. Apart from new interaction methods, the technical contribution of alignment and hashing can also be used to build other functionalities.

However, there is still some limitations in the player interactive analytics. The current interface for player analytics (see Figure 3) only allows user to manipulate the spatial locations of players at the shooting frame, but not the actions or velocity that are very important in modeling the ExP. Since we have grouped the shots into eight classes for better performance, some complementary input formats should be added so that users can make more specific query with detailed information that leads to this shot. From the technical perspective, the ExP prediction accuracy can be further improved if more sophisticated learning method and better latent vector are used. As we are focusing on the human–machine interaction in this work, those issues beyond the scope of this article and will not be discussed here.

On the other hand, the current interaction is still passive as users can only get feedback by providing certain situations or the player of interest. Some tasks like player recommendation and style recommendation should be able to perform automatically, which could further simplified the interaction process and boost the efficiency. For example, given a shooting situation, instead of letting users to find an ideal formation or player by manually changing the identity and location of the shooter, the system should be able to recommend the best spatial formation for the current shooter, or the best shooter for the current formation. Further, both team analytics and player analytics can be combined to improve the interaction experience. After a team is selected, its most frequent used strategies and the best player deployment for each strategies can be showed simultaneously. In order to achieve such simplified interaction, we need to build a much more complex backend framework that is left for our future work.

7 CONCLUSION

In this article, we have presented an interactive retrieval and analytics interface that enables a user to ask specific questions and received specific answers. The key component of this work is the formulation of a new query format that can naturally capture complex semantics of multi-agent trajectories for sports play retrieval. We have also presented a retrieval system tailored toward efficient and accurate sports play retrieval. Our full-stack system can achieve accurate retrieval at interactive speeds, and we demonstrate its effectiveness in a retrieval user study where our approach achieves orders-of-magnitude improvement in retrieval effectiveness over baseline methods. Beyond sports domains, query formats such as the one presented here could be applied to a wide range of spatiotemporal retrieval settings. Perhaps the most natural data domain is other types of behavioral tracking data, such as animal behavior.

ACKNOWLEDGMENTS

The authors would like to thank Dr. P. Carr of Disney Research Pittsburgh for his guidance in the computer vision part of this work.

REFERENCES

[1] David Altimira, Florian Floyd Mueller, Jenny Clarke, Gun Lee, Mark Billinghurst, and Christoph Bartneck. 2016. Digitally augmenting sports: An opportunity for exploring and understanding novel balancing techniques. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1681–1691.

[2] Shahriyar Amini, A. J. Brush, John Krumm, Jaime Teevan, and Amy Karlson. 2012. Trajectory-aware mobile search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2561–2564.

[3] National Basketball Coaches Association and Giorgio Gandolfi. 2009. *NBA Coaches Playbook: Techniques, Tactics, and Teaching Points*. Human Kinetics.

[4] Paul N. Bennett, Filip Radlinski, Ryen W. White, and Emine Yilmaz. 2011. Inferring and using location metadata to personalize web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 135–144.

[5] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, and Iain Matthews. 2014. "Win at home and draw away": Automatic formation analysis highlighting the differences in home and away team behaviors. In *Proceedings of MIT Sloan Sports Analytics Conference (SSAC'14)*.

[6] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, and I. Matthews. 2014. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *Proceedings of ICDM*.

[7] Nastaran Borjian, Ehsanollah Kabir, Sanaz Seyedin, and Ellips Masehian. 2017. A query-by-example music retrieval system using feature and decision fusion. *Multimedia Tools and Applications* 77, 5 (2017), 6165–6189.

[8] Christina Brandt, Thorsten Joachims, Yisong Yue, and Jacob Bank. 2011. Dynamic ranked retrieval. In *ACM International Conference on Web Search and Data Mining (WSDM'11)*. ACM, 247–256.

[9] Christopher J. C. Burges. 2010. *From Ranknet to Lambdarank to Lambdamart: An Overview.* Technical Report MSR-TR-2010-82. Microsoft Research.

[10] Peter Carr, Yaser Sheikh, and Iain Matthews. 2012. Point-less calibration: Camera parameters from gradient-based alignment to edge images. In *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 377–384.

[11] Dan Cervone, Alexander D'Amour, Luke Bornn, and Kirk Goldsberry. 2014. POINTWISE: Predicting points and valuing decisions in real time with NBA optical tracking data. In *Proceedings of the 8th MIT Sloan Sports Analytics Conference*, Vol. 28.

[12] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems* 30, 1 (2012), 6:1–6:41.

[13] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. 2009. Expected reciprocal rank for graded relevance. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, 621–630.

[14] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. 2011. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 167–176.

[15] Jianhui Chen and Peter Carr. 2015. Mimicking human camera operators. In *Proceedings of 2015 IEEE Winter Conference on Applications of Computer Vision (WACV'15)*. IEEE, 215–222.

[16] Sheng Chen, Zhongyuan Feng, Qingkai Lu, Behrooz Mahasseni, T. S. Fiez, Alan Fern, and Sinisa Todorovic. 2014. Play type recognition in real-world football video. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV'14)*.

[17] Yueguo Chen, M. A. Nascimento, Beng Chin Ooi, and A. K. H. Tung. 2007. SpADe: On shape-based pattern detection in streaming time series. In *Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE'07)*. 786–795. DOI: http://dx.doi.org/10.1109/ICDE.2007.367924

[18] Zhiyong Cheng and Jialie Shen. 2014. Just-for-me: An adaptive personalization system for location-aware social music recommendation. In *Proceedings of International Conference on Multimedia Retrieval*. ACM, 185.

[19] Stéphane Clinchant, Julien Ah-Pine, and Gabriela Csurka. 2011. Semantic combination of textual and visual information in multimedia retrieval. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. ACM, 44.

[20] Philipp Eichmann and Emanuel Zgraggen. 2015. Evaluating subjective accuracy in time series pattern-matching using human-annotated rankings. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 28–37.

[21] Khalid El-Arini and Carlos Guestrin. 2011. Beyond keyword search: Discovering relevant scientific literature. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. ACM, 439–447.

[22] Dan Feldman, Matthew Faulkner, and Andreas Krause. 2011. Scalable training of mixture models via coresets. In *Proceedings of Neural Information Processing Systems (NIPS'11)*. 2142–2150.

[23] Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6 (1981), 381–395.

[24] Michael Fleischman and Deb Roy. 2007. Unsupervised content-based indexing of sports video. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval*. ACM, 87–94.

[25] Eric Foote, Peter Carr, Patrick Lucey, Yaser Sheikh, and Iain Matthews. 2013. One-man-band: A touch screen interface for producing live multi-camera sports broadcasts. In *Proceedings of the 21st ACM International Conference on Multimedia*. ACM, 163–172.

[26] Alexander Franks, Andrew Miller, Luke Bornn, and Kirk Goldsberry. 2015. Counterpoints: Advanced defensive metrics for NBA basketball. In *Proceedings of MIT Sloan Sports Analytics Conference (SSAC'15)*.

[27] Zhouyu Fu. 2015. Improving feature aggregation for semantic music retrieval. In *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM, 1019–1022.

[28] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity search in high dimensions via hashing. In *Proceedings of Conference on Very Large Databases (VLDB'99)*, Vol. 99. 518–529.

[29] Andreas Girgensohn, Don Kimber, Jim Vaughan, Tao Yang, Frank Shipman, Thea Turner, Eleanor Rieffel, Lynn Wilcox, Francine Chen, and Tony Dunnigan. 2007. DOTS: Support for effective video surveillance. In *Proceedings of the 15th ACM International Conference on Multimedia*. ACM, 423–432.

[30] Sariel Har-Peled and Soham Mazumdar. 2004. On coresets for k-means and k-median clustering. In *Proceedings of ACM Symposium on Theory of Computing (STOC'04)*. ACM, 291–300.

[31] Weiming Hu, Dan Xie, Zhouyu Fu, Wenrong Zeng, and Steve Maybank. 2007. Semantic-based surveillance video retrieval. *IEEE Transactions on Image Processing* 16, 4 (2007), 1168–1181.

[32] Samuel Huston, J. Shane Culpepper, and W. Bruce Croft. 2014. Indexing word sequences for ranked retrieval. *ACM Transactions on Information Systems* 32, 1 (2014), 3.

[33] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of ACM Symposium on Theory of Computing (STOC'98)*. ACM, 604–613.

[34] Imran N. Junejo, Omar Javed, and Mubarak Shah. 2004. Multi feature path modeling for video surveillance. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 2. 716–719.

[35] Mehran Kafai, Kave Eshghi, and Bir Bhanu. 2014. Discrete cosine transform locality-sensitive hashes for face retrieval. *IEEE Transactions on Multimedia* 16, 4 (2014), 1090–1103.

[36] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7, 3 (2005), 358–386.

[37] Eamonn J. Keogh and Michael J. Pazzani. 2000. Scaling up dynamic time warping for datamining applications. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 285–289.

[38] Felix Kosmalla, Florian Daiber, and Antonio Krüger. 2015. ClimbSense: Automatic climbing route recognition using wrist-worn inertia measurement units. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2033–2042.

[39] H. W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1–2 (1955), 83–97.

[40] Thi-Lan Le, Alain Boucher, Monique Thonnat, and François Brémond. 2008. A framework for surveillance video indexing and retrieval. In *Proceedings of International Workshop on Content-Based Multimedia Indexing (CBMI'08)*. IEEE, 338–345.

[41] Bo Li, Yijuan Lu, Azeem Ghumman, Bradley Strylowski, Mario Gutierrez, Safiyah Sadiq, Scott Forster, Natacha Feola, and Travis Bugerin. 2015. 3D sketch-based 3D model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 555–558.

[42] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. 2003. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. ACM, 2–11.

[43] Jennifer Listgarten, Radford M. Neal, Sam T. Roweis, and Andrew Emili. 2004. Multiple alignment of continuous time series. In *Proceedings of Neural Information Processing Systems (NIPS'04)*. 817–824.

[44] Huiying Liu, Shuqiang Jiang, Qingming Huang, and Changsheng Xu. 2008. A generic virtual content insertion system based on visual attention analysis. In *Proceedings of the 16th ACM International Conference on Multimedia*. ACM, 379–388.

[45] Jiangung Lou, Qifeng Liu, Tieniu Tan, and Weirning Hu. 2002. Semantic interpretation of object activities in a surveillance system. In *Proceedings of the 16th International Conference on Pattern Recognition*, Vol. 3. IEEE, 777–780.

[46] David G. Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, 1150–1157.

[47] Patrick Lucey, Alina Bialkowski, Peter Carr, Stuart Morgan, Iain Matthews, and Yaser Sheikh. 2013. Representing and discovering adversarial team behaviors using player roles. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*.

[48] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Vol. 1. Cambridge University Press, Cambridge.

[49] Armand McQueen, Jenna Wiens, and John Guttag. 2014. Automatically recognizing on-ball screens. In *Proceedings of MIT Sloan Sports Analytics Conference (SSAC'14)*.

[50] Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. 2014. Factorized point process intensities: A spatial analysis of professional basketball. In *Proceedings of International Conference on Machine Learning (ICML'14)*.
[51] Florian Mueller, Rohit A. Khot, Alan D. Chatham, Sebastiaan Pijnappel, Cagdas Chad Toprak, and Joe Marshall. 2013. HCI with sports. In *Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI'13)*. ACM, 2509–2512.
[52] Apostol Paul Natsev, Alexander Haubold, Jelena Tešić, Lexing Xie, and Rong Yan. 2007. Semantic concept-based query expansion and re-ranking for multimedia retrieval. In *Proceedings of the 15th ACM International Conference on Multimedia*. ACM, 991–1000.
[53] Leyla Norooz, Matthew Louis Mauriello, Anita Jorgensen, Brenna McNally, and Jon E. Froehlich. 2015. BodyVis: A new approach to body learning through wearable sensing and visualization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1025–1034.
[54] Stina Nylander, Jakob Tholander, Florian Mueller, and Joe Marshall. 2014. HCI and sports. In *Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI'14)*. ACM, 115–118.
[55] Hyung Kun Park, HyeonBeom Yi, and Woohun Lee. 2017. Recording and sharing non-visible information on body movement while skateboarding. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2488–2492.
[56] Mark Perry, Oskar Juhlin, Mattias Esbjörnsson, and Arvid Engström. 2009. Lean collaboration through video gestures: Co-ordinating the production of live televised sport. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2279–2288.
[57] Tom Polk, Jing Yang, Yueqi Hu, and Ye Zhao. 2014. Tennivis: Visualization for tennis match analysis. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2339–2348.
[58] Jie Qin, Li Liu, Mengyang Yu, Yunhong Wang, and Ling Shao. 2015. Fast action retrieval from videos via feature disaggregation. In *Proceedings of British Machine Vision Conference (BMVC'15)*.
[59] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R. G. Lanckriet, Roger Levy, and Nuno Vasconcelos. 2010. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*. ACM, 251–260.
[60] Shuai Ren, Cheng Jin, Chang Sun, and Yuejie Zhang. 2014. Sketch-based image retrieval via adaptive weighting. In *Proceedings of International Conference on Multimedia Retrieval*. ACM, 427.
[61] Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. 2005. FTW: Fast similarity search under the time warping distance. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, 326–337.
[62] Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc.
[63] Long Sha, Patrick Lucey, Yisong Yue, Peter Carr, Charlie Rohlf, and Iain Matthews. 2016. Chalkboarding: A new spatiotemporal query paradigm for sports play retrieval. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 336–347.
[64] Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Metro maps of science. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, 1122–1130.
[65] Jialie Shen, Tao Mei, Dacheng Tao, Xuelong Li, and Yong Rui. 2015. EMIF: Towards a scalable and effective indexing framework for large scale music retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 543–546.
[66] L STATS. 2010. SportVU. Retrieved from https://www.stats.com/sportvu-basketball/.
[67] Gita Sukthankar and Katia Sycara. 2011. Activity recognition for dynamic multi-agent teams. *ACM Transactions on Intelligent Systems and Technology* 3, 1 (2011), Article 18.
[68] Martin Svensson, Kristina Höök, and Rickard Cöster. 2005. Designing and evaluating kalas: A social navigation system for food recipes. *ACM Transactions on Computer-Human Interaction* 12, 3 (2005), 374–400.
[69] Kevin Toohey and Matt Duckham. 2015. Trajectory similarity measures. *SIGSPATIAL Special* 7, 1 (2015), 43–50.
[70] Aaron Wall. 2015. History of Search Engines: From 1945 to Google Today. Retrieved from http://www.searchenginehistory.com.
[71] Xinyu Wei, Long Sha, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. 2013. Large-scale analysis of formations in soccer. In *Proceedings of 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA'13)*. IEEE, 1–8.
[72] Pei-Chih Wen, Wei-Chih Cheng, Yu-Shuen Wang, Hung-Kuo Chu, Nick C. Tang, and Hong-Yuan Mark Liao. 2016. Court reconstruction for camera calibration in broadcast basketball videos. *IEEE Transactions on Visualization and Computer Graphics* 22, 5 (2016), 1517–1526. DOI: http://dx.doi.org/10.1109/TVCG.2015.2440236
[73] Changcheng Xiao, Changhu Wang, Liqing Zhang, and Lei Zhang. 2015. Sketch-based image retrieval via shape words. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 571–574.

[74] Dan Xu, Xavier Alameda-Pineda, Jingkuan Song, Elisa Ricci, and Nicu Sebe. 2016. Academic coupled dictionary learning for sketch-based image retrieval. In *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 1326–1335.

[75] Alexandre Yahi, Antoine Chassang, Louis Raynaud, Hugo Duthil, and Duen Horng Polo Chau. 2015. Aurigo: An interactive tour planner for personalized itineraries. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM, 275–285.

[76] Xinguo Yu, Xin Yan, Tran Thi Phuong Chi, and Loong Fah Cheong. 2006. Inserting 3D projected virtual content into broadcast tennis video. In *Proceedings of the 14th ACM international conference on Multimedia*. ACM, 619–622.

[77] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. A support vector method for optimizing average precision. In *Proceedings of ACM Conference on Information Retrieval (SIGIR'07)*. ACM, 271–278.

[78] Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain Matthews. 2014. Learning fine grained spatial models for dynamic sports play prediction. In *Proceedings of IEEE International Conference on Data Mining (ICDM'14)*.

[79] Shaoting Zhang, Ming Yang, Timothee Cour, Kai Yu, and Dimitris N. Metaxas. 2015. Query specific rank fusion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 4 (2015), 803–815.

[80] Zhang Zhang, Kaiqi Huang, and Tieniu Tan. 2006. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proceedings of 18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 3. IEEE, 1135–1138.

Received January 2017; revised September 2017; accepted January 2018