Int J Software Informatics, Volume 10, Issue 3 (2016), pp. 000–000
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661048

DOI: 10.21655/ijsi.1673-7288.00231

Performance Histogram Curve: Abstractive Visualization and Analysis of NBA Games

Biao Zhu and Wei Chen

(VINCI '16 Dallas, Texas, USA)

Abstract NBA game is one of the world's most exciting sports. Multiple datasets for a NBA game are available, e.g., play-by-play data, shot-position data, twitter and videos. Existing methods for visualizing game data commonly focus on the composition of multi-variate information of a game process. In this paper, we introduce a new parametric modeling approach, Performance Histogram Curve (PHC), that locally and adaptively encodes the game progression with game-related features derived from the play-by-play data. By transforming a PHC into the two-dimensional space with a two-phase projection technique, we create a unique 2D line representation. The 2D representation and auxiliary views abstracts the progress of a game and the performance of each team along the timeline. Our integrated system favor browsing a single play, analyzing game performance, and comparing multiple games. We conducted two case studies to demonstrate the effectiveness of our approach.
Key words: projection; temporal; multi-dimension; NBA

Zhu B, Chen W. Performance histogram curve: Abstractive visualization and analysis of NBA games. *Int J Software Informatics*, **Vol.10, No.3 (2016): 000–000. http://www.ijsi.org/1673-7288/10/231.htm**

1 Introduction

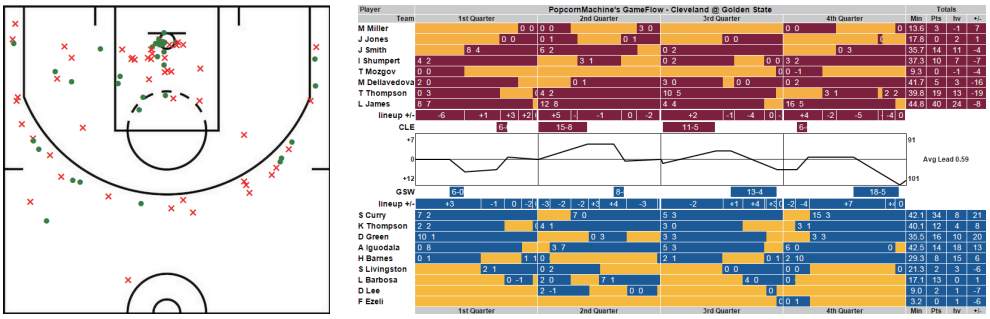
Basketball is one of the most popular sports all over the world. And NBA (National Basketball Association) is no doubt the premier men's professional basketball league. At the age of big data, people from all walks of life collect different kinds of data, as is in the basketball industry. Many companies and associations have been constantly collecting all kinds of NBA game data for many years.

Besides the common data, such as images, videos and box score data, that can be found on the NBA official site, there is more detailed NBA game data available to public. For instance, Basketball-reference[1] provides data that records every event within a game, and shot position data that records positions where players make field goals. Sportradar[2] provides API for convenient access to all kinds of NBA game data including Play-By-Play data, player profile and so on.

Corresponding author: Biao Zhu, Email: zhubiao@zju.edu.cn
Received 2016-08-15; Revised 2016-10-01; Accepted 2016-10-28.
[1]http://www.basketball-reference.com
[2]http://developer.sportradar.us

Such game data is quite helpful. For NBA teams or coaches, they use it to evaluate players' performance in each game, also to analyze their advantages and shortcomings, so as to play better in the following games. For basketball fans, they knows how their idol players performs against other players, or whether a already finished game deserves watching.

Visualization techniques have already been applied to better convey such game data. Basketball-reference plots shot positions with markers (Fig. 1(a)), two different kinds of markers represent the two against teams. It clearly shows how many and what kind of field goals both sides made, as well as the exact positions those field goals are made at. PopcornMachine[3] designs a game flow chart (Fig. 1(b)) to summarize a overview of a game. From it, users can gain much useful information, such as who wins the game, how a specific player contributes to the game and whether the game is intense (score difference curve seesaws frequently may indicate that the game is intense) at a quick glance.



(a) short charts (b) game flow

Figure 1. NBA game visualization examples.

Despite so much effort made to collect, and visualize NBA game data, to the best of our knowledge, there is not yet any visualization approach to effectively analyze NBA games so far. For instance, users can easily learn how score difference changes over time from the game flow chart (Fig. 1(b)), but it can never be easy for them to understand what leads to those changes. It might be a tactic made by the coach, or simply a good athletic state of several core players. From the view of the data, it is time-varying as well as multi-dimensional, which makes it a significant challenge to effectively analyze it.

In this paper, we use data as input. The data is firstly discretized with a sliding window approach, then smoothed with a gaussian kernel. After that, we employ LMDS to consistently project the multi-dimension data onto 2D. Together with elaborate interactions, our approach enables users to perform

– thorough analysis of a single NBA game,

– comparison among different NBA games and

– analysis of different NBA game patterns.

[3]http://popcornmachine.net

The paper is organized as follows. First, related work is discussed in Section 2. The visual analytics approach is presented in Section 3. In Section 4 we apply our approach to artificial and real-world dynamic networks. The approach is discussed in Section 5 and finally, conclusions and directions for future work are given in Section 6.

2 Related Work

Due to the popularity of basketball, many efforts have been made to collect and analyze NBA game data, in both academia and industry. Jin et al.[8] leverage treemap to visualize NBA players, the size and color of cells are used to convey the attributes of players. To facilitate the task of basketball coaches, Therón et al.[12] visualize GPS (Global Positioning System) position of players, thus assisting dynamic distance and area analysis. Kowshik et al.[11] visualize annotated optical tracking data with heat map, support analysis players respect to events like shooting, passing and dribbling. PopcornMachine[1] designed GameFlow diagram to summarize a single NBA game with points difference curve and technical statistics. Above mentioned approach can partially support the analysis of players' performance. However, none of them is able to effectively analyze the performance of a team, they can not answer questions like what tactics a team used to win a game, or how the tactics of a team different from others. In this paper, we tried to address such issues.

Dimensionality reduction is a well studied approach to analyze high-dimensional data. SOM (self-organizing Maps)[10] is an unsupervised non-linear approach to cluster similar data, Hu et al.[6] use SOM together with LLE to layout human motion data. PCA (Principal Component Analysis)[9] and MDS (Multidimensional Scaling)[4] are widely used dimensionality reduction approaches, both of them can project a N-dimensional data into a user defined M-dimensional space ($M < N$). Chen et al.[3] visualize sequential document with a 2D layout produced by MDS. Van et al.[13] and Bach et al.[2] both analyze temporal by reducing high-dimensional data to a points with MDS and PCA. Dominik et al.[7] leverage 1D MDS to analyze temporal multivariate data. The common thing among all these works is that they all project high-dimensional data into 2D layout and link projected points together to express some kind of sequential order (e.g. temporal order). In this paper, we handle Play-By-Play data in a similar way.

3 Method

3.1 Play-by-play data

The Play-By-Play data used in this paper is crawled from basketball-reference. Just as we can tell from its literal meaning, it records every "play" within games in ascending temporal order. Figure 2 is an example segment of such data. Each line is a "play", which consists of time, detailed action and points gained.

The data covers almost all the actions taken within a NBA game, including miss/make shot, offensive/deffensive rebound, turnover, timeout, substitution, free throw, foul and so on. In other words, it contains enough information to fully reflects the performance of players and the whole team.

9:43.0	Defensive rebound by I. Shumpert	
9:22.0	Turnover by T. Mozgov (bad pass; steal by S. Curry)	
9:16.0	K. Thompson makes 2-pt shot from 2 ft (assist by S. Curry)	+2
8:50.0	T. Mozgov misses 2-pt shot from 20 ft	
8:50.0	Offensive rebound by Team	
8:50.0	Loose ball foul by H. Barnes (drawn by T. Mozgov)	
8:39.0	M. Dellavedova makes 2-pt shot from 10 ft	+2
8:22.0	A. Iguodala misses 3-pt shot from 26 ft	
8:21.0	Defensive rebound by T. Thompson	
8:13.0	I. Shumpert misses 3-pt shot from 23 ft	
8:12.0	Defensive rebound by A. Iguodala	
8:07.0	S. Curry makes 2-pt shot from 3 ft (assist by A. Iguodala)	+2
7:46.0	Turnover by M. Dellavedova (bad pass; steal by A. Iguodala)	
7:41.0	D. Green makes 2-pt shot at rim (assist by A. Iguodala)	+2
7:20.0	L. James misses 2-pt shot from 19 ft	

Figure 2. An example segment of Play-By-Play data.

3.2 Discretization and smooth

The Play-By-Play data is much like the activity log mentioned in Ref. [13], i.e., a game G can be defined as:

$$G = (P_1, P_2, \ldots, P_N), \tag{1}$$

where $P_i (i \in [1, N])$ is a play. Unlike the activity mentioned in Ref. [13], a play does not necessary involve two instances, i.e., it does not contains an edge to form a dynamic network.

So, we process the data in a different way. While we also employ a sliding window to discrete the data, instead of extracting a network, we statistically calculate predefined dimensions for each window and separately for each team. In total, we predefined 23 dimensions, $D_i (i \in [1, 23])$, as shown in Table 1. Some of them are directly generated by count how many times corresponding action occurs within a window, others are proportional dimensions calculated from count dimensions. Users can select/unselect dimensions to meet their analyzing target. As a result, we get a collection of multi dimensional vectors reflecting behavior patterns for each teams, the number of vectors for each team is equal to the number of the window. We name these vectors as feature vectors.

Table 1 Dimensions calculated from Play-By-Play data.

dimension type	dimension name
count	points / field goals attempt / field goals made / three points attempt / three points made / two points attempt / two points made / free throw attempt / free throw made / assists / turnovers / blocks / rebounds / offensive rebounds / defensive rebounds / steals / personal fouls / technical fouls
proportional	field goals percentage / three points percentage / two points percentage / free throw percentage / assists turnover ratio

Play-By-Play data is quite sparse, and those dimensions generated by count are indeed category data with finite states. As a result, even with a sliding window, the

result data is not that smooth. As we can see in the first row in Fig. 3, even when the overlap ratio becomes 359/360, the project result is still not smooth enough for analysis. Figure 3(g) shows the result of comparison view (Section 3.5.2) corresponding to Fig. 3(c), it reveals some of the reason. As we can see in Fig. 3(g), most variables have no more than 20 different values, some even below 10. It means that those feature vectors only have limited number of different status, in another word, they are more like discrete rather than continuous, thus not projected to form a smooth curve.

In order to make such data available for visual analysis, we use a gaussian smooth approach to further smooth the feature vectors. Here, we have two parameters, the smooth radius and the delta within the gaussian kernel, to control the level of smooth. A default delta is empirically set to 1/3 of the smooth radius. Figure 3 shows the effect of gaussian smooth under different radius and delta settings. The larger the radius, the more the smooth. Figures 3(e) and (f) are smooth enough to perform further analysis.

To enable analysis at different level of details, we provide interface for users to interactively adjust all the discretization and smooth parameters, including window size, window offset, smooth radius and gaussian delta.

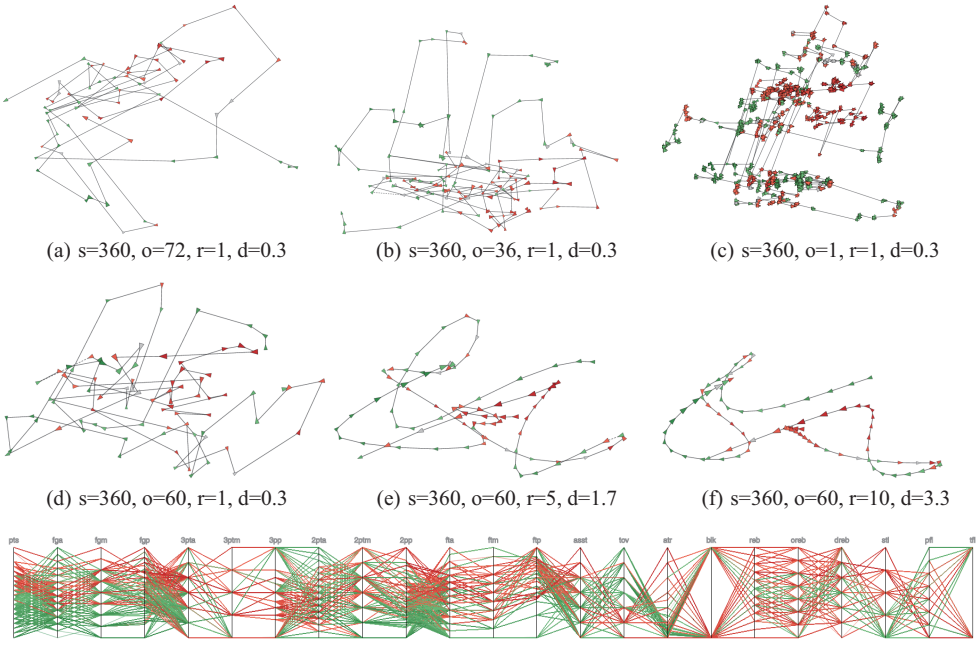


Figure 3. First row: effect of different window offset in the discretization process. Second row: effect of different smooth radius and gaussian delta in the smooth process. Here, 's' stands for window size; 'o' stands for window offset; 'r' stands for smooth radius; 'd' stands for gaussian delta.

3.3 Normalization

Those dimensions get by counting are measured in the same way, but they are

likely to result in diverse ranges, e.g., dimension "points" tends to have larger value than dimension "field goals made". The values of those proportional dimensions are all within range $[0, 1]$, thus is supposed to be much smaller than those counting dimensions. Without normalization, dimensions with large value will tend to dominate the final projection results. So, before applying projection methods, normalization must be performed first. In this paper, we use a min-max normalization that scales the data into a fixed range $[0, 1]$.

After a min-max normalization, all dimensions are with the same scale. However, depend on the analysis target, sometimes users may want to put particular emphasis on several most relevant dimensions. To meet such a demand, we allow users to interactively adjust the weight $w_i \in [0, 1]$ ($i \in [1, 23]$) of each dimension, D is the number of dimensions. In this way, this approach becomes more flexible and can support analysis from different perspectives.

3.4 Dimensionality reduction

After the above steps, for each game, we get a time-varying multidimensional data for both teams. We can soon come up with many mature methods to visualize such a data. For instance, we can leverage parallel coordinates or radar chart to address the multidimensional aspect, and use sequential colors to encode time. In such a way, however, we can not even clearly see the data due to visual clutter, let alone analysis.

MDS is a classical isometry-invariant multidimensional scaling method. It can reduce multidimensional vectors into 2D points, while preserve their relative distance in high dimensional space. By visualizing with points, visual clutter problem is dramatically relieved, while geometric features are preserved to the most. Then, we can start search and analyze features in 2D space, which is more feasible and efficient.

When comparing among different games, if MDS process is performed separately each time we make a choice of games, then the projected points of the same game will be different in different game combinations. An ideal way to perform MDS would be processing all the data at the same time. However, a major concern of MDS is that it does not scales well. We have the Play-By-Play data of thousands of NBA games, it could take minutes to perform a MDS process over all the data, which does not meet our interactive analysis requirement.

In this paper, we employ LMDS[5], instead of the classical MDS, to address this issue. LMDS is performed in two phases:

1. M vectors are randomly chosen from the whole dataset as landmarks and then projected to 2D with classical MDS.

2. Each vector in the chosen data is projected with an affine linear transformation, taking its squared distances to the landmarks as input.

It is much faster than classical MDS, especially when the input data grows large. Games can be projected separately, while the projected points are stable Because they are projected within the same embedding computed from the landmarks. Thus it allows reasonable comparison among different games.

3.5 Visualization and interaction

To enable interactive analysis of the Play-By-Play data, besides the conventional parameter setting, game and dimension selection views, we also use three other views (as shown in Fig. 4): a projection view showing the distribution of the projected points; a comparison view use parallel coordinates to display and compare selected dimensions; and a auxiliary score view to display basic game information such as team name and score difference curve.

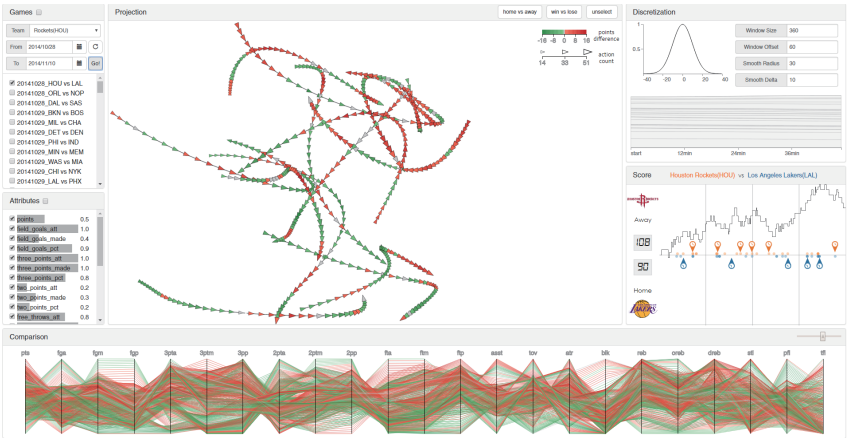


Figure 4. User interface of the prototype.

3.5.1 Projection view

For the design of the projection view, our target is to show the projection result clearly, while encoding as much information as possible to assist efficient analysis. Figure 5(a) is part of the result from a single game projection, each triangle stands for a projected point. Triangles are linked in a temporal ascending order so that users can better perceive these triangles as a sequence, dash lines encode intervals between adjacent quarters. To emphasize the temporal ascending order, we make each triangle orients to its temporal subsequent neighbor triangle. To enable comparison of both teams involved in a game, we visualize each team separately, i.e., there are two linked strings of triangles for each game.

The color of a triangle encodes the points difference against the other team during the corresponding window. Red sequential colors encode positive points difference and Green sequential colors are used to encode negative points difference, when two teams make a tie, we fill the triangle with gray. Thus, from the color of the triangles, we can know when a team is in good condition and gains points, and vise versa. The size of a triangle is used to encode how many actions the corresponding team takes within the window. More actions taken may indicates a more intensive play.

Our approach are designed to support visualization and comparisons among multi NBA games. However, with a projection based approach, visual clutter will become increasingly severe with the increase in number of entity. As we can see in Fig. 5(c), we can hardly distinguish the projection result shown in Fig. 5(a) from others. To address this issue, we highlight the whole game when users hover over any node of

the game, and the node hovered gets extra emphasize on its size. Also, we will show detailed information associated with that node.

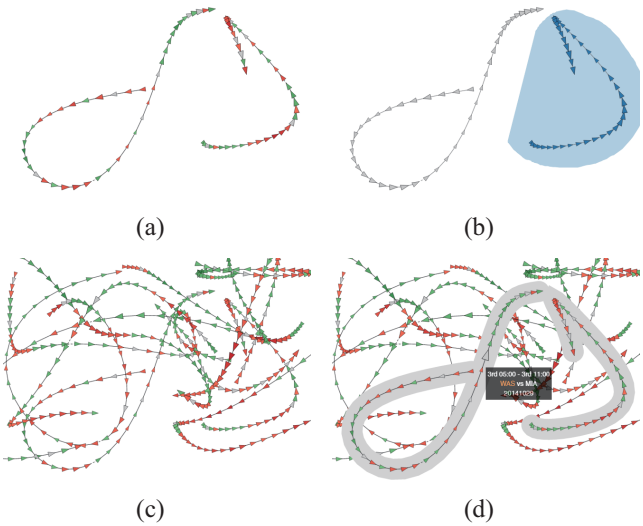


Figure 5. Design of Projection view. (a) Projection result of a single game. The two lines represents two teams. (b) Perform selection on an interested area. (c) Projection result of multiple games. (d) Hovered node and game are highlighted.

Projection view is taken as a main view for analysis, so it is placed in the center as we can see in Fig. 4. Users are supposed to find interested patterns in this view. So, we provide an interaction that users can freely drag a closed path to choose interested nodes (as shown in Fig. 5(b)). After the selection, users can see the distribution in each dimension of the selected nodes in the comparison view (Section 3.5.2) and perform further analysis. Besides, we provide buttons for quick access to several predefined selections for specific analysis target. For instance, press the "home vs away" button will perform two selections, select the games the selected team as a home team and as a away team respectively, enabling analysis of how a team plays differently at home court and at away court.

3.5.2 Comparison view

To compare the nodes that users selected in the projection view, we use parallel coordinates (Fig. 6). Parallel coordinates is widely used to display multi dimensional data, it encodes each multi dimensional vector as a polyline. As a result, however, it may encounter severe visual clutter when there is plenty of vectors to show.

In our approach, users are supposed to select clusters in the projection view. Thanks to the isometry-invariant characteristic of LMDS, visually perceived clusters in the projection view usually indicate that the corresponding data is similar in the original multi dimensional space. Which means, when data within a visually perceived cluster plotted with parallel coordinates, their corresponding polylines have a high possibility being bundled together. Thus visual clutter issue of parallel coordinates is reduced. As we can see in Fig. 6, it is easy to compare among the three selections of data, we can clearly see how they differ respect to different dimensions.

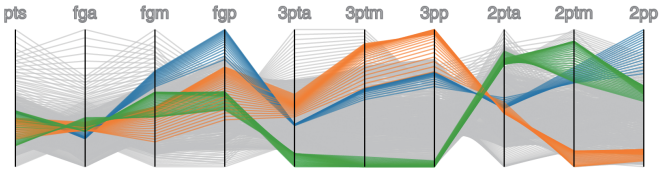


Figure 6. Compare selected data with parallel coordinate.

3.5.3 Score difference view

In NBA games, score is a significant metric. Score difference shows the leading team and finally decides which team wins. From how score difference changes, we can even infer whether a game is intensive and worth watching. So, inspired by popcornMachine, we designed a score difference view, as shown in Fig. 7.

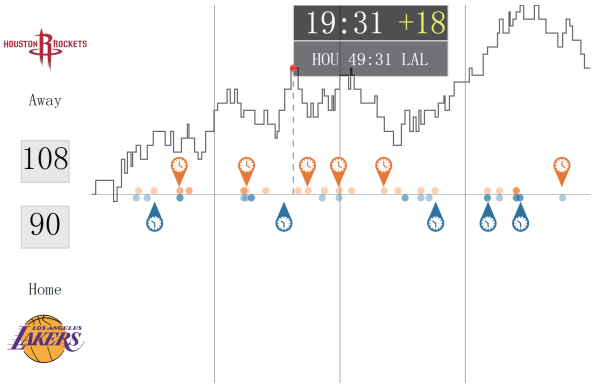


Figure 7. Score difference view.

Team logos, their role (home/away) and the final score are displayed on the left side. On the right side is the score difference curve, time information is encoded in x axis, it lapse from left to right, i.e., the origin stands for 00:00 and the right most position stands for 48:00, we use three vertical lines to indicate quarter intervals. We ignore overtime for simplicity. The y axis encodes score difference: if the score difference curve is above the x axis, it means that the away team is leading the game; if it is on the x axis, it means a draw; if it is below the x axis, it means that the home team is leading the game.

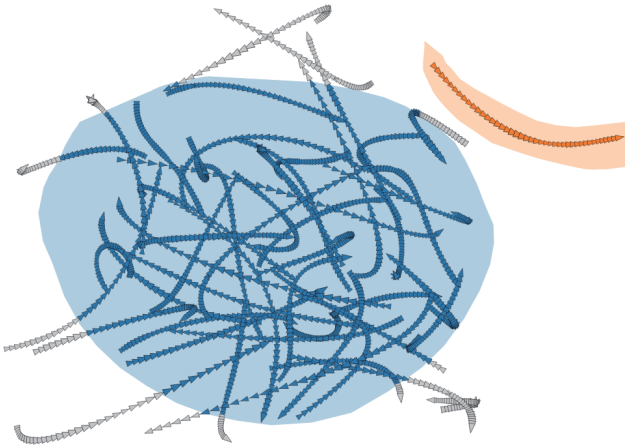
Substitution events and timeout events are encoded along the x axis, with a transparent dot and a pause icon respectively. We add a transparency to the substitution dot because several substitution events could happen at the same time. On mouse hover, we use a dash line and a red dot to indicate the current hovered position, along with a message box to show the exact time, score and score difference at that time point.

4 Case Study

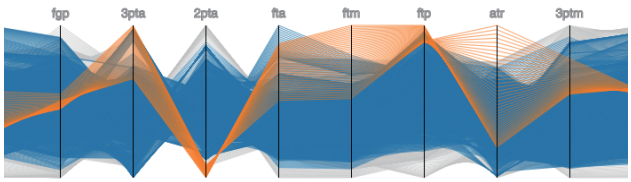
4.1 Outlier detection

According to the isometry-invariant characteristic of LMDS, it preserves the relative distance of the original high dimensional data while reducing to a lower dimensional space, i.e., feature vectors with a high difference tend to have a long distance on the 2D plane. As a result, outliers should easily be perceived visually.

As we can see in Fig. 8(a), there is an obvious outlier at the top right corner. In order to analyze the reason why it is so different. We perform a selection of it and another selection of the rest, then the dimensional details of the two selection is shown in the comparison view, as we can see in Fig. 8(b). We can see that the outlier team attempts more three points goals and attempts much less two points goals, compared to other teams. It means that this team is good at attack from outside. Also, this team made more free throws, has a higher free throw hit rate and assist turnover ratio, which means this team behaves stable and calm in this game.



(a) Outliers can be visually detected in the projection view



(b) Further analysis of the outliers can be done in the comparison view

Figure 8. Outlier detection case.

5 Conclusion

NBA game has been one of the world's most exciting sports for many years. However, through all these years, there is not yet any visualization approach to effectively analyze NBA games. We have developed a new approach to support visual analysis of NBA games. With our approach, users are able to perform

in-depth analysis of individual game (team), as well as comparison among different games (teams). We have demonstrated the visual analysis process and the effectiveness of our approach with case study on real NBA game data.

References

[1] Popcornmachine. `http://popcornmachine.net/`, 2003.

[2] Bach B, Shi C, Heulot N, Madhyastha T, Grabowski T, Dragicevic P. Time curves: Folding time to visualize patterns of temporal evolution in data. IEEE Trans. on Visualization and Computer Graphics, 2016, 22(1): 559–568.

[3] Chen H, Wang G, Peng D, Zuo W, Chen W. Sequential document visualization based on hierarchical parametric histogram curves. Tsinghua Science and Technology, 2012, 17(4): 409–418.

[4] Cox TF, Cox MA. Multidimensional Scaling. CRC Press, 2000.

[5] De Silva V, Tenenbaum JB. Sparse multidimensional scaling using landmark points. [Technical Report], Stanford University, 2004.

[6] Hu Y, Wu S, Xia S, Fu J, Chen W. Motion track: Visualizing variations of human motion data. 2010 IEEE Pacific Visualization Symposium (PacificVis). IEEE. 2010. 153–160.

[7] Jäckle D, Fischer F, Schreck T, Keim DA. Temporal mds plots for analysis of multivariate data. IEEE Trans. on Visualization and Computer Graphics, 2016, 22(1): 141–150.

[8] Jin L, Banks D. Hierarchical visualization with treemaps: Making sense of pro basketball data. IEEE Computer Graphics and Applications, 1997, 17(4): 63–65.

[9] Jolliffe I. Principal Component Analysis. Wiley Online Library, 2002.

[10] Kohonen T. Self-organized formation of topologically correct feature maps. Biological Cybernetics, 1982, 43(1): 59–69.

[11] Kowshik G, Chang YH, Maheswaran R. Visualization of event-based motion-tracking sports data. [Technical Report], University of Southern California, 2012.

[12] Therón R, Casares L. Visual analysis of time-motion in basketball games. Smart Graphics. Springer. 2010. 196–207.

[13] van den Elzen S, Holten D, Blaas J, van Wijk JJ. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. IEEE Trans. on Visualization and Computer Graphics, 2016, 22(1): 1–10.