Baseball4D: A Tool for Baseball Game Reconstruction & Visualization

Carlos Dietrich, David Koop, Huy T. Vo, and Cláudio T. Silva, *IEEE Fellow*

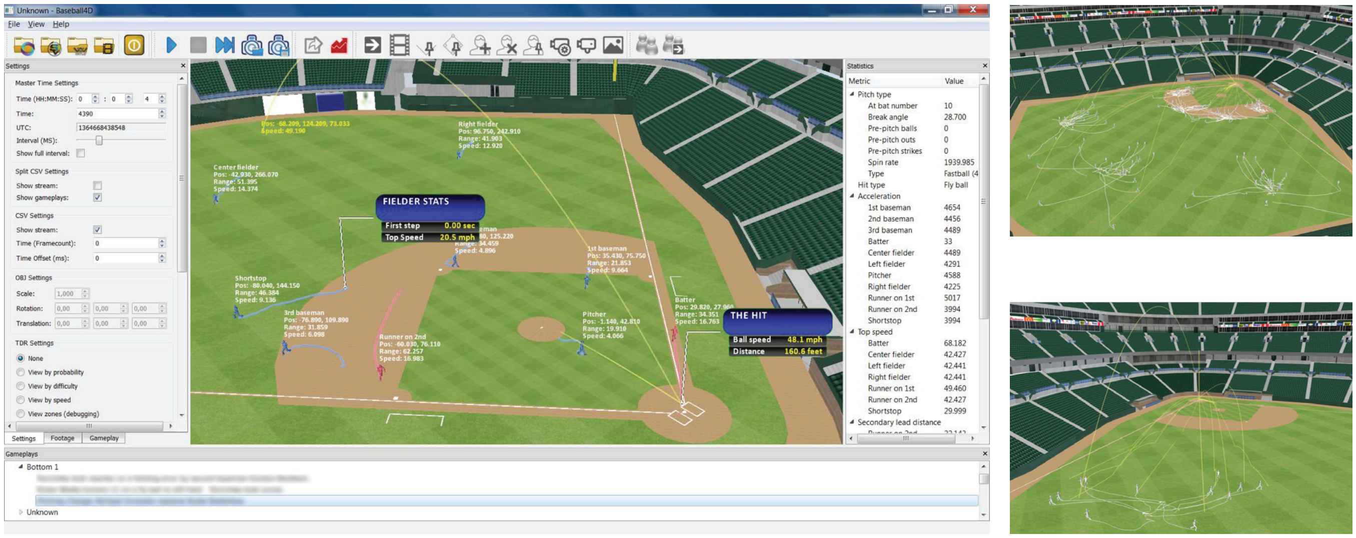


Fig. 1: Baseball4D. A user may explore a single play (left) by scrubbing through the play's timeline, enabling various visual components, and examining static and dynamic statistics. Users may also analyze multiple plays at once using filtering widgets. For example, a user might examine fielder positions in a game for a specific batter (top-right) or a particular fielder's movements during a game (bottom-right).

Abstract— While many sports use statistics and video to analyze and improve game play, baseball has led the charge throughout its history. With the advent of new technologies that allow all players and the ball to be tracked across the entire field, it is now possible to bring this understanding to another level. From discrete positions across time, we present techniques to reconstruct entire baseball games and visually explore each play. This provides opportunities to not only derive new metrics for the game, but also allow us to investigate existing measures with targeted visualizations. In addition, our techniques allow users to filter on demand so specific situations can be analyzed both in general and according to those situations. We show that gameplay can be accurately reconstructed from the raw position data and discuss how visualization and statistical methods can combine to better inform baseball analyses.

Index Terms—sports visualization, sports analytics, baseball, game reconstruction, baseball metrics, event data

1 Introduction

While there has always been interest in analyzing sports, this area has received significantly more attention in recent years due to both the recognition of the importance of objective statistics and the proliferation of available data. New technology promises not only to automate the capture of some of the statistics but also track the positions of all players and capture game events. Instead of being limited to a fixed set of measures, analysts will be able to explore entire games and construct new statistics that can be computed on-the-fly. Such extensions require interactive tools that show entire gameplays, enable comparisons, and display informative raw statistics.

Baseball, a team sport with seasons comprised of large numbers of games and individual battles, has led the movement, offering archives of statistics from decades of data. Enough data exists so that past matchups between a single pitcher and batter can offer reasonable predictions for a given at bat. Indeed, pitching and batting statistics like earned run average (ERA) and on-base percentage (OBP) usually accurately reflect the performance of individual players. However, despite the wealth of statistics in these areas, fielding (what happens after a batter contacts the ball) is often only understood by whether the play results in an error or out. One baseball analyst estimated that while statistics reflect 85-90% of a player's batting or pitching performance, current fielding statistics only give 5% of the picture [5].

Analyzing baseball player performance has long captured the interest of fans, but its importance has greatly increased as team owners, managers, and players have recognized how objective statistics can be used to manage teams. When the 2002 Oakland Athletics made an improbable playoff run with a team whose total salary ranked near the bottom of all teams in Major League Baseball, word spread that their success was partially due to the use of objective statistics to find overlooked players that help provide wins [16]. Now, many teams use similar strategies to better evaluate prospective and current players. Such interest has led to a thirst for more raw data and more targeted statistics.

With new technology, fans and analysts alike look forward to more

- *Carlos Dietrich is an independent consultant. E-mail: cadietrich@gmail.com*
- *Cláudio T. Silva, Huy T. Vo, and David Koop are with NYU. E-mail: {csilva, huy.vo, dakoop}@nyu.edu.*

IEEE Symposium on Visual Analytics Science and Technology 2014
November 9-14, Paris, France
978-1-4799-6227-3/14/$31.00 ©2014 IEEE

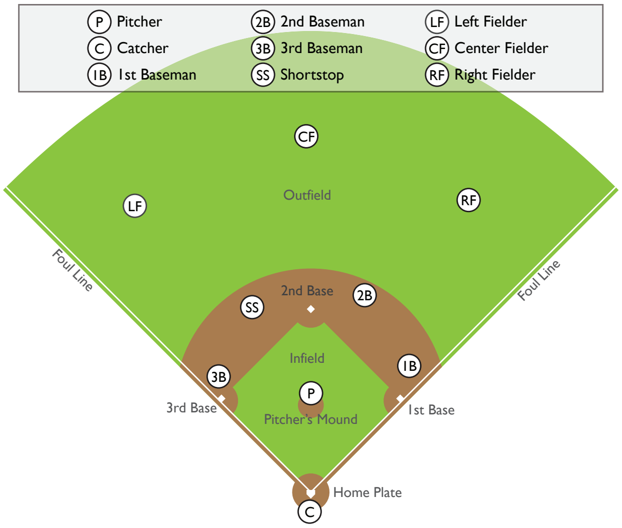


Fig. 2: A diagram showing the major features of a baseball field as well as the common positions for fielding players.

data enriching the understanding of the game. However, the raw data is more often automatically captured and processed which can often introduce noise and/or errors. In addition, this data is often produced by different tools or algorithms and must be reconciled. Finally, while the positions of players and the ball can serve as a visual reconstruction of a game, they do not necessarily capture the game semantics (*e.g.,* when a run was scored or if a base stealer was out). This requires infrastructure to reconstruct gameplay.

We present Baseball4D, a tool that starts to addresses these concerns and allows a new kind of exploration of baseball games. From ball tracking, player tracking, and pitch tracking, we produce full gameplay visualizations. In addition, we show how these visualizations can be interactively explored and how new metrics for fielding and baserunning can be derived from this information. We also allow visualizations to be filtered and customized to show an individual fielder's performance or a set of similar plays. We have used our technique on over 1,100 games comprised of over 52,000 play segments.

Our work includes the following specific contributions:

- We present Baseball4D, the first tool designed to reconstruct baseball games from time-varying 3D ball and player positioning. Baseball4D is a visual analytics tool that allows games to be studied in unprecedented detail. It breaks games into gameplay segments, which can be studied by themselves, but it also allows a collection of gameplays to be explored together, thus allowing game behaviors that were difficult to analyze to now be exactly quantified.

- This paper describes the first use of new in-ballpark infrastructure designed to capture high-resolution gameplay. We describe techniques for using the raw data streams to produce self-contained gameplays that can be used for baseball analysis.

- We present three use cases for Baseball4D. First, we focus on single-play visualization culminating with the validation of the captured input by showing superimposed real footage on top of reconstructed games. Second, we present visualizations that can be computed from a (possibly filtered) collection of gameplays, generating previously unseen renderings. Finally, we provide the first implementation of Rybarczyk's True Defensive Range (TDR) metric [27] using captured data. We discuss how these results can be used for quantifying the difficulty of catches.

2 Related Work

There has been significant work in computer vision from video to extract trajectories of people and objects in sports. In tennis, Pingali et al. introduced LucentVision to obtain motion trajectories of players and the ball [23]. Bebie and Bieri introduced SoccerMan which uses synchronous video to allow games to be virtually replayed from new viewpoints [4]. Saito et al. introduced methods for tracking players in soccer games using multiple-camera video [28]. Höferlin et al. explored the use of video visualization in snooker skill training and discussed how video can be translated to spatiotemporal attributes [14]. Yu and Farin provide a survey and classification of many of the techniques and systems involved in sports video processing [35].

Further work has used this positional data of players to try to understand and classify types of plays or movements. Magee and Perš edited a special issue on computer vision-based analysis in sports [17]. In that issue, Perše et al. looked at complex multi-player behavior during basketball games by analyzing trajectories [21]. Hervieu et al. used hierarchical parallel semi-Markov models to characterize eight activity phases from player trajectories and the referee's whistle [13].

In addition, there has significant interest in commercial systems that automatically capture player locations, game events, and other information throughout the game. Usually, these systems use some type of video processing with multiple cameras, sometimes combined with manual guidance or annotations. In baseball, Sportvision developed PITCHf/X to capture the full path of the ball from the pitcher's hand to the plate [24], and FIELDf/X [10] and PlayItOver [25] promise to capture players and the ball throughout the entire game [5]. STATS has developed the SportVU player tracking technology for soccer, basketball, and American football [32]. Even among sports that do not currently employ automated capture (*e.g.,* hockey), analysts often manually track information about shots [22].

In the area of visualization and visual analytics, there has also been work targeting sports. In 2013, a workshop on sports data visualization was held during IEEE VisWeek and included work on baseball pitch analysis [19]. Pileggi et al. examined the role of visualization in sports analytics and provided a survey of existing work in a range of different sports [22]. Pingali et al. introduced a number of techniques for tennis including virtual replays of serves and coverage maps [23]. Cox and Stasko examined baseball specifically with baseline bar displays and player maps [9]. Wongsuphasawat and Gotz aggregated outcomes of soccer games via clusters and pathways based on events and statistics [34], and Perin et al. showed how soccer data can be used to create connected visualizations that tell stories about play progressions [20]. Albinsson and Andersson note the importance of team-sport event analysis and show how multiple attributes can be explored across histograms and two-dimensional maps via linked views [1]. Legg et al. showed that glyph-based visualization can be used for real-time analysis of rugby during matches [15], and Chung et al. presented a knowledge-assisted sorting system to help users explore game events and video footage [7].

Chloropleth maps (also known as heat maps) and their derivatives have seen significant adoption in sports visualization and analysis. Pitching heat maps have been very popular among fans, with web sites allowing users to customize the display for each pitcher (*e.g.,* [2]). Pileggi et al. used both radial and traditional heat maps to analyze hockey shot data [22], and Goldsberry used a scaled-glyph heat map to track both the density (via scale) and projected point value (via color) of shots in basketball [11].

In addition, both sports fans and sports entertainment industry have long been interested in both accurate capture, replay, and visualization of games and players' traits. Baseball has often led the way in this area, incubating the field of *sabermetrics*, coined from the acronym for the Society of American Baseball Research (SABR), which historically drew interest primarily from fans of the game. Also, video game developers employ motion capture in order to generate more realistic players for its games [6]. Television networks like ESPN commissioned methods like "K Zone" [12] to augment their broadcasts and help viewers better understand the strike zone in baseball, and continue to employ PITCHf/X data to show pitch trajectories. The New York Times Magazine featured a video that visualizes the reconstructed pitches of one famous relief pitcher [26]. Also, the interest in fantasy sports, where fans draft teams comprised of players from dif-

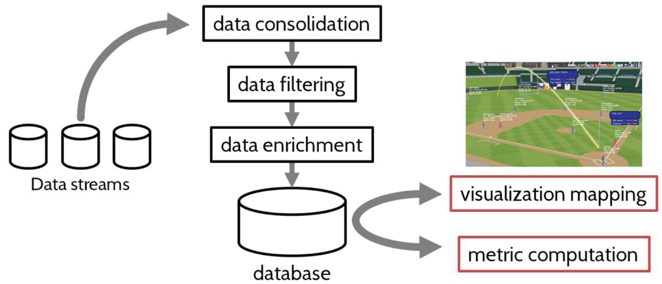


Fig. 3: The Baseball4D architecture. The three data streams (player positions, ball position, and game events) are first consolidated into gameplays and then filtered and enriched with data computed from player movements. The final set of gameplays is then stored for visualization and statistical analysis tasks.

ferent teams and earn points based on each player's individual statistics, has led to even greater interest in sports statistics and analysis.

Furthermore, those involved in the business side of sports, including team owners, managers, scouts, and players, have also recognized the utility of analytics and visualization. While sabermetrics originally drew little attention from this group, this changed with the 2002 Oakland Athletics who used statistics to evaluate player performance in order to better draft and manage their roster [16]. Now, most teams in baseball and many other sports use statistics in evaluating performance, and there are conferences, including the MIT Sloan Sports Analytics Conference, devoted to sports analysis. Interestingly, exclusively using statistics can also cause analysts to ignore important information so a hybrid approach that allows people to explore data is preferred [30].

3 Baseball: Definitions & Overview

While the concepts and visualizations we present have analogs in other sports, we will focus on baseball in this paper. Baseball is played on a (mostly) level field shaped like a circular quadrant. The arc of the quadrant is often deformed due to stadium configurations while the radii define the boundary between *fair* and *foul* territories. The field has a set of four small, square bases placed on ninety-foot square (often called the diamond) that is aligned with the radii and center of enclosing circle of the quadrant. The bases are labeled in counter-clockwise order starting at this center as *home*, *first*, *second*, and *third*. In the middle of the diamond is the raised *pitcher's mound* which is oriented toward home base. The *infield* is the area immediately surrounding the diamond and its interior while the *outfield* comprises the rest of the field. See Figure 2 for a diagram. Baseball is distinct from some other games (notably soccer) in that it does not have a fixed-size playing area, only the infield has a standard size. At home base (also called home plate), there are designated batting areas surrounding the plate. The usual required equipment is a *ball*, usually white and around 3in in diameter, one bat (around 3–3.5 feet in length), and one glove for each fielder to help them catch the ball.

At any time, there are nine active players for each team.[1] There are also a varying number of *umpires* who serve to officiate the game (usually at home plate, 1st base, 2nd base, 3rd base, left field foul inline, and right field four line), and various coaches on the field.

A baseball *game* is divided into *innings*, and each inning has a *top* half where one team bats and the other fields, and a *bottom* half where the roles reverse. The fielding team has a pitcher who stands on the pitcher's mound, a catcher who squats behind home plate, four infielders (first baseman, second baseman, third baseman, and shortstop), and three outfielders (left, center, and right fielders). When a team *bats*, they go through a designated order, maintained across innings, where one player at a time becomes the current batter who is then *at bat*. The batter stands in the batting area on the left or right of home plate and the opposing team's *pitcher* throws (delivers) the ball toward home plate. The batter may swing and try to contact the ball or *take* the pitch and let the catcher catch it. If he swings and does not contact the ball, it is a *strike*. If he takes the pitch, the ball is ruled by an umpire as a *ball* or a *strike* depending on whether it goes through the *strike zone*—the region over home plate between the hitter's shoulders and knees. If he does make contact and the ball drops in foul territory, it is also a strike unless he already has two strikes. If a batter receives three strikes, the batter is *out*, but if he receives four balls, he receives a *walk* and moves to first base, advancing any runners that would end up on the same base to the next base. If the ball, after being contacted by the batter, is caught, the batter is also out. Otherwise, the batter runs to first base, and possibly on to the next consecutive base(s).

If he makes it to first base before the ball makes to a fielder standing on first base, he is *safe*, stays on base, and becomes a *runner*. During the time the ball is in play, any other runners on base can advance to the next consecutive bases. However, if the ball is caught, they must return to their base after the ball is caught before trying to advance. In addition, if any runner is tagged by a fielder with the ball, he is also out. If a runner makes it to home plate after touching all other bases, the batting team scores a *run*. If the ball goes past the outside arc of the field (usually designated by a fence or wall) but between the extended radii of the quadrant, this is a *home run*, and all players including the batter run around the bases to home plate to score a run. Once there are three outs, the half of the inning is over. The team with most runs at the end of a set number of innings wins; if there is a tie, additional innings are played until the tie is resolved. For a complete description of the game and its rules, please see [18].

4 Requirements

Our goal is a system that allows interactive visual analysis of baseball gameplay. To that end, we have two important considerations: the analysis framework and the gameplay data.

4.1 Analysis Framework

Baseball analysts have defined many statistics that help characterize strengths and weaknesses of particular players based on the outcome of at bats or balls in play. Importantly, these statistics allow aggregation over a variety of possible filters. For example, one might examine a hitter's batting average in day versus night games, at a particular stadium, or against a specific pitcher. We wish to allow interactive filtering across gameplay data by allowing users to select events based on a variety of criteria. Such selection might be done directly by selecting specific plays or players, or via user-defined queries, *e.g.,* all plays where the third baseman is batting right-handed.

In addition, our goal is to go beyond outcome-based statistics so that the entirety of the play, including player and ball movement, can be taken into account. Thus, time is important here, and one should be able to examine the state of the players and ball at the sampled timestamps. Then, instead of comparing whether two fielders caught a line drive, an analyst might, for a similarly hit ball, compare their reaction time, speed, and directness of movement. In some cases, such comparisons may lend themselves to the discovery of new statistics such as those which measure fielding performance, but there will also be situations where direct visual comparison is more appropriate. Also note that such visual comparison should go beyond that which can be captured by a sequence of video clips by, for example, allowing direct overlays of multiple plays. With the entire gameplay being taken into consideration, we can also embed real-time statistics (*e.g.,* velocity, acceleration) into a visual display.

In summary, we require an analysis framework that allows visual exploration of multiple timestamped gameplay sequences with flexible user-configurable aggregation and filtering. Our overall design follows Shneiderman's Visual Information Seeking Mantra: "overview first, zoom and filter, then details-on-demand" [29]. Thus, a user should be able to quickly visualize one or more plays, filter those plays based on a variety of criteria, and examine statistics or other metrics on-demand.

[1]When playing with a designated hitter, there are ten, but there are still at most nine fielders and nine batters at any time.

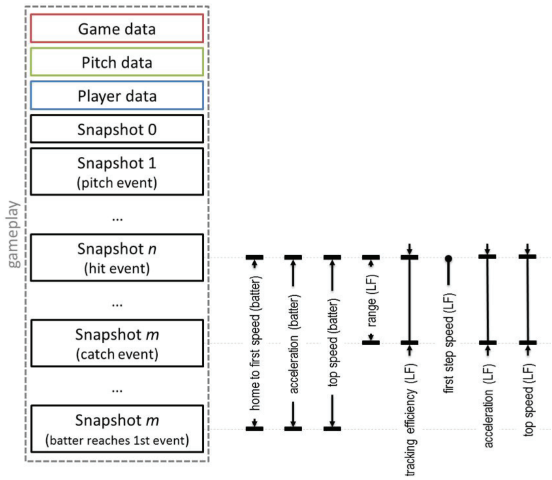


Fig. 4: Common player metrics are computed from snapshots. Metrics are usually defined between events, or when a specific event happens, meaning that only a single loop over the series of snapshots is usually required to compute metrics.

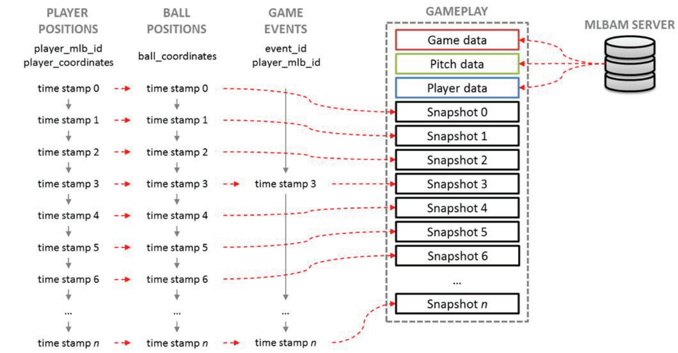


Fig. 5: Records from different data sources are integrated to reconstruct a gameplay. Gameplays are defined between game events and combine all the records available in that time range. The first three sections of the gameplay are populated from external sources (for our examples, via MLBAM Gameday server). The data streams, after filtering and enrichment, populate the gameplay *snapshots*, which encode the dynamic data of each play.

4.2 Gameplay Data

To feed such a framework, we require data that include the player positions, ball location, and semantically-meaningful game events. For our purposes, we consider a baseball game as a continuous stream of data and events, including pitches, catches, throws, and player movements. This game action can be segmented into plays. In general, each play starts when the pitcher goes into his windup and finishes when the ball returns to the pitcher's glove or goes out of play (*e.g.,* a home run or foul ball). Then, plays can be divided into three parts: the pitch (*i.e.,* actions from the windup to the moment the ball is in the front of home plate), the hit (*i.e.,* actions from the moment the ball is hit to the moment it is fielded), and the field (*i.e.,* actions from the moment the ball is fielded to the end of the play). Understanding the state of the ball is critical to determining what part of the play we might be in. At the same time, tracking the ball is challenging, since its state changes considerably, ranging from a reasonably controlled environment (*e.g.,* the pitch) to unpredictable and potentially fast movements as it goes from one player to another.

There are a number of systems that provide information about the game. For example, Sportvision's PITCHf/x system [24] has been installed on most major league ballparks since the end of the 2007 season. MLB Advanced Media provides a textual description of the game as part of their Gameday Data API. Other data providers, including Sportvision [31], Trackman [33], and ChyronHego [8] provide various hit and field data. We do note that play tracking is a new and expanding technology, and it will continue to improve over the years. Our system tries to be as agnostic as possible to vendor-specific data streams. Below we describe a set of minimum assumptions for our system.

We assume three streams of data:

Player positions This stream contains timestamped positions of the people on the field. Obviously, the players are the most relevant, but umpires and coaches may also be tracked. Each position is defined by a pair of coordinates (x,y) in the *diamond domain*[2] and a player identifier. Positions are measured by units in fractional feet (*i.e.,* they are not metric). Time is captured as UTC timestamps in milliseconds. We assume that player positions are captured at least 25 times per second (at least once every 40 milliseconds).

Ball position This stream contains the timestamped position of the ball. The position is defined as a triplet of coordinates (x,y,z) in the diamond domain. Again, the coordinates are measured in fractional feet. Because the ball moves an average of 6.37 times faster than players, the frequency of capture for the ball needs to be higher, at least 150 times per second.

Game events This stream contains timestamped game events. Game events are the high-level events associated with the ball as it is in play—the moments when a specific player hits, obtains possession, or releases possession of the ball. The data stream is composed of tuples that contain a timestamp, game event, and player id. A minimum set of important game events include: "ball was pitched", "ball was caught", "ball was released", "ball was hit", "end of play", "pick off released", and "ball was deflected". More details of the game can be captured by adding game events.

Because this information is often captured by different hardware and/or methods, combining this data is necessary to accurately recreate gameplay. For example, because run-of-the-mill game statistics normally capture hits, runs, and errors, neither the player nor ball position data contains that information. However, using the positions of baserunners as well as the outcome of a particular play (*e.g.,* single, groundout, etc.), we might determine these outcomes.

Timestamps play a critical role in aligning the different sources of information. Some timestamps are synchronized across different types of data while others, like those for pitch tracking, use a separate clock. Often, an offset can be computed between the two so the data can be synchronized. More problematic is the fact that the start and end timestamps for plays often do not align with the pitches. Sometimes this can be corrected by allowing some leeway between timestamps. Another important issue that is beyond the scope of this paper is uncertainty associated with the data, in particular the timestamps.

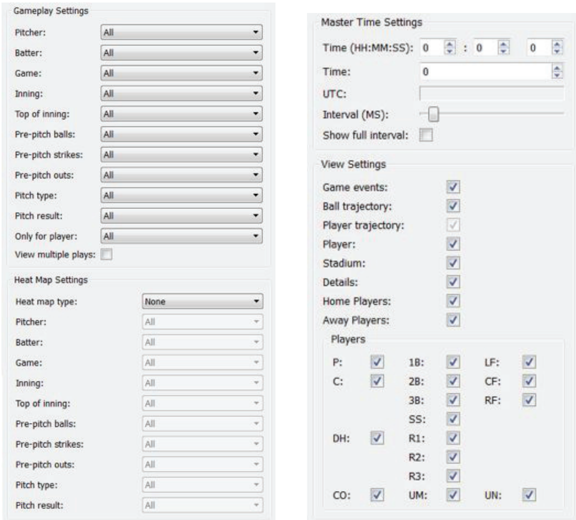
5 The Baseball4D System

With these requirements in mind, we have built Baseball4D, a visual analytics tool for exploring baseball plays. The key components of the system include a representation for gameplay data, infrastructure for aligning, enriching, and storing game data streams, a visual interface for displaying gameplays, and a set of widgets that allow interactive exploration.

5.1 Gameplay Representation

Most baseball statistics are defined based on the outcomes of pitches, whether they are strikes, balls, or put into play. With data that captures the positions of players and the ball in addition to game events, we define a new way to structure baseball data in a self-contained and non-redundant representation that respects the play boundary. A *gameplay* combines pitching, hitting, running, and fielding information, putting spatial and temporal data in a unified format. Such gameplays allow

[2] The *diamond domain* is defined by the y-axis connecting home plate to 2nd base and the x-axis perpendicular to it, and it is oriented according to the well-known "right-hand rule" so the z-axis points up from the field.



(a) (b)

Fig. 6: Some of the Baseball4D widgets that may be used to filter the data (a) and visualization settings (b), allowing an analyst to pinpoint specific players and types of plays.

the computation of existing statistics, visual representations, and new explorations and statistics.

There are four components of the gameplay structure: game data, pitch data, player data, and a set of snapshots with position and event data derived from data streams (see Figure 5). The first three components are obtained from the Major League Baseball Advanced Media (MLBAM) Gameday server. This data is important in allowing users to filter plays based on players, pitches, or other information, because it ties the position information to the players. This allows us to compute player-specific statistics. While some of this information could be normalized (*e.g.,* the first baseman for a team may not change the entire game), we wish to allow cross-cutting analyses where gameplays might be selected from different games or even seasons so our gameplay instances are *independent*. For visualizations that draw information from different games, the locality of the game, pitch, and player data saves time without adding much storage overhead.

Play dynamics are captured by a series of snapshots. A *snapshot* is a representation of the entire field at a given time. It stores players' positions, the ball position, and all game events that occurred at that moment. A series of snapshots is like a series of frames in a video, encoding all tracking information in a given period of time. Snapshots are built directly from the records of the data streams—every time we have a timestamped record as input, no matter the player, ball, or event-type, a new snapshot instance is created in the gameplay. The record is added to the newly created snapshot, as well as any other records with the same timestamp. The result is a series of snapshots identified by timestamps, where all records with the same timestamp belong to the same snapshot (see Figure 5).

5.2 Data Processing

An important consideration in Baseball4D is processing the data streams described in Section 4.2. In order to analyze and visualize the data, we need to consolidate, filter, enrich, and store the raw data streams (see Figure 3). First, because these streams are often obtained from different hardware, they must be consolidated. The records are aligned based on their timestamps, and the streams are divided into gameplays and integrated into a sequence of snapshots for each gameplay.

After consolidation, we filter the gameplays to check for inconsistencies and potential errors in the optical motion tracking. There are four types of filtering that occur:

1. Positional: Check that players' trajectories are consistent so their positions do not jump around the field.

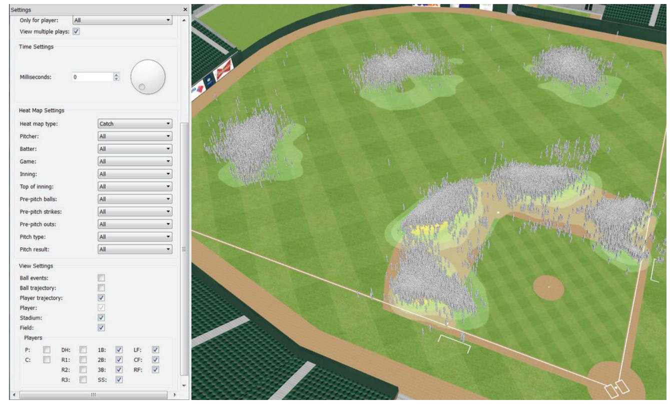


Fig. 7: A heat map showing the locations of all hit balls (no selection applied) and player positioning at pitch release. Even simple visualization mappings, such this one, may aid the analysis of gameplay data. The heat map peaks indicate that the 3rd baseman and the shortstop field most of the balls.

2. Timing: Check that the positions are evenly sampled.

3. Event: Check that the game events sequentially satisfy the rules of the game using a state machine.

4. Semantic: Check that the snapshots match the player, pitch, and game information.

Event filtering checks that the gameplays record three outs per inning, account for runs, etc. Semantic filtering checks that game events match the actual positions of players and the ball. For example, a catch cannot occur unless the player and the ball are in close proximity.

Once gameplays are built and filtered, they go through an enrichment step. This step adds data computed from gameplay data itself and from external sources, *e.g.,* MLBAM Gameday server[3]. Some metrics require special game events, like the moment a runner starts to run, reaches a base, or is tagged. Those events that (1) are not part of the input data and (2) may be computed automatically, are added in this step like any other event required for metric computation. Gameplays can also be enriched with external data, including, for example, information about games, pitches, and play statistics available on the MLBAM Gameday server. This data is linked to gameplays through timestamps which are stored at millisecond resolution. The enriched gameplays are stored in a database where they can be used for visualization, metrics, and interactive analyses.

5.3 Interface

The main Baseball4D interface combines a core 3D visualization canvas that allows interactive views of one or more plays with a set of panels containing widgets that both present related statistics and metrics and allow user selection and filtering (see Figure 1). The main canvas shows the entire field by default, along with the positions (and past trajectories) of the players and ball for the current selected gameplay(s). Importantly, it can not only show a single gameplay as a type of "replay", but it can also visually represent multiple gameplays. In addition, statistics can be displayed for players (*e.g.,* current speed, acceleration) in their current position. Users can change camera position or zoom to examine specific locations. Finally, we can present visualizations like heatmaps that provide other views of the data in separate windows.

Some of the surrounding widgets provide information about the players involved in the gameplay and statistics about them, both static (*e.g.,* their batting averages) and dynamic (*e.g.,* their current running speeds). The right side of Figure 1 shows examples of such metrics. The metrics are computed from gameplays on demand. Metrics are usually defined over game events, or when a specific event happens,

[3] `http://gd2.mlb.com/components/game/mlb/`

Fig. 8: Here, we present real field footage augmented with data captured by the tracking system. We overlay several visual elements that make it easy to see relevant statistics for players and also add visual cues for players and ball, making it much easier to study the game as it happens. All the visual elements can be toggled in real-time on the Baseball4D interface.

meaning that a loop through all snapshots and their events shall suffice to gather the data required for each metric computation. An example of common offensive and defensive metrics computation is illustrated in Figure 4.

Other widgets allow users to select specific gameplays, filter the visualization, or hide graphical elements. Gameplays can be manually selected in the bottom panel, and it is possible to select more than one to be visualized. All other selections and filtering are controlled through the settings panel on the left. These interactions can be quite powerful; it is possible to select all gameplays that fit certain criteria (see Figure 6). For example, by selecting a particular pitcher, all his gameplays can be retrieved by the system, and their statistics can be studied. By adding a batter and his team, fairly complex data products can be studied, and it is possible to allow analysts to see how teams or specific players are positioned during a game (see Figures 1 and 7). In addition, the system makes use of linked views in non-trivial ways; for example, clicking on a textual event during a gameplay will bring you to that particular point in time. The system also has a comprehensive querying API that is not user accessible at this time.

6 Use Cases

Because baseball is played so often, it has drawn much attention from analysts and statisticians. Not only do teams play at least 162 games each season, but because hitting is mostly an individual competition between the pitcher and batter, there are fewer complex team interactions to analyze. The unique rules of the game, coupled with copious amounts of data, has allowed a great deal of analysis through objective evidence. In the following use cases, we describe three different ways Baseball4D can be used that demonstrate the functionality and usefulness of the system with respect to baseball analysis. These studies focus on gameplay as and after the ball is hit, allowing analysts to better understand the dynamics of ball trajectories and fielding players.

The first case focuses on the use of Baseball4D for data from a single gameplay, showing how visual elements augment the exploration of a reconstructed play. The second use case utilizes Baseball4D's query capabilities over information from multiple gameplays to generate visualizations that provide insight into specific game situations (*e.g.,* positioning of fielders for different pitch types). The final use case analyzes the True Defensive Range [27] (TDR) metric, which to our knowledge has not been computed and analyzed before.

Our work started in 2011, and we have data from over 1,000 games, which are a subset of games played in the 2011, 2012, and 2013 seasons. We have a fairly large collection from which it is possible to generate statistically significant results, relevant visualizations, and metric computations. We note that MLBAM has recently announced that complete coverage of all games with the new tracking technology is expected by the 2015 season, roughly a year away. We believe our work will serve as a foundation for analyzing the massive data stream once all of the data becomes available.

6.1 Visualizing Single Gameplays

We will not attempt to be comprehensive on the description of the functionality of the system. Instead, we will focus on key features that have shown to be particularly useful for baseball analysis. As we described before, Baseball4D merges multiple data streams into gameplays, which contain player and ball movements, and timestamped events of everything that happened during the play. This data can be used in a variety of ways. A simple and effective visual representation is to compute a non-photorealistic depiction of the game state, and allow the user to toggle important metrics and visual objects. An example is shown in Figure 1. A few relevant features that are worth highlighting include visual widgets for displaying important information, *e.g.,* maximum ball speed; as well as visual elements to highlight player and ball tracks. Baseball4D also supports an "augmented reality" mode shown in Figure 8. This is similar to what is used in broadcast graphics, but in our system, we do not need for an operator to assist the system. Borrowing ideas from augmented reality and computer vision, we automatically calibrate real video footage with the reconstructed game, and automatically superimpose all the visual elements which can be toggled at will.

These features enhance analysis because they integrate objective measurements into gameplay reconstruction without saddling baseball analysts with the time-consuming tasks of integrating stopwatch times or retracing trajectories from video captures. Like real video footage, Baseball4D can also pause plays at any timestep, and it adds statistics and trajectories to the replay. While many analysts can visually determine whether a fielder got a good jump on the ball, the added speed and time statistics in Baseball4D allow more consistent judgment of reaction times or throwing speed. For the example shown in Figure 8, the ball trajectory makes the path of the two-bounce hit clearer, and combined with fielder trajectories, it allows analysts to understand how well a fielder judged a particular hit. That example also shows both the fielders' and baserunners' speeds integrated with the visualization, allowing a better understanding of speed at any given timestep.

Although we emphasize the use of the system for analysis of plays assuming that the input data is accurate, one of the reasons that we developed Baseball4D was to understand the accuracy and reliability of the input data itself. This was particularly important in the early days as the tracking systems were being tested and validated. For example, when two players run close to each other, the tracking system needs to determine which player is which as they move apart; reconstruction with Baseball4D allows developers to revisit the system's decisions to understand why a particular trajectory may have been confused.

6.2 Exploring Multiple Gameplays: Spray Charts and Heatmaps

The advent of pitch tracking technology, like PITCHf/X, has allowed analysts to not only aggregate pitch data and visualize it with heat maps but also, and perhaps more importantly, to filter that data based

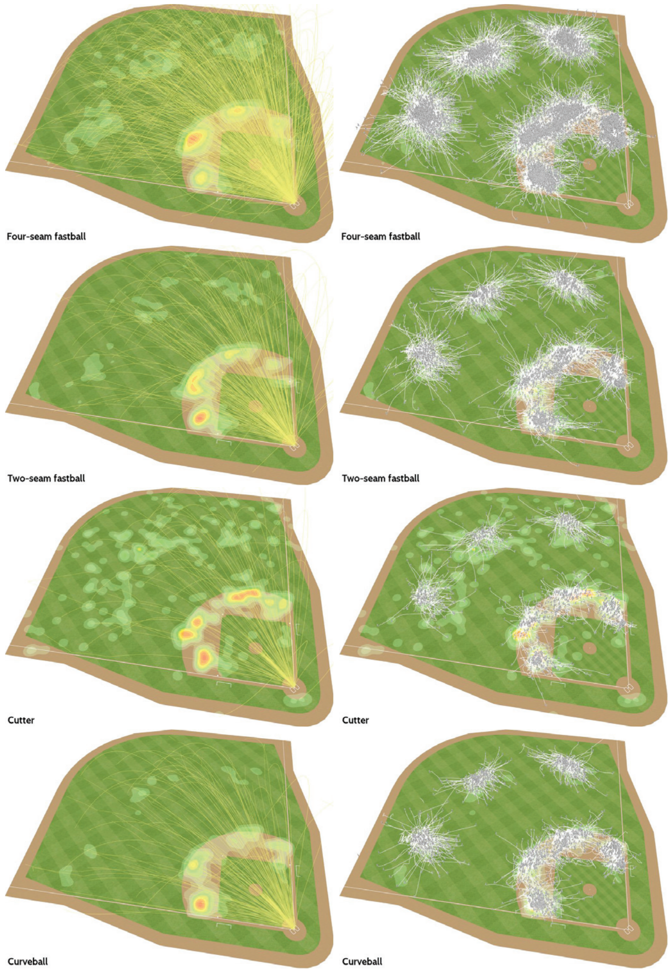


Fig. 9: Ball and fielder trajectories vary for hits that arise from different pitch types.

on hitter characteristics, pitch types, and weather conditions. With Baseball4D, we show that full-field tracking presents opportunities to continue both aggregation and filtering for other parts of the game, including hit trajectories, fielding, and baserunning. In addition, this filtering can be extended to utilize pitch information as well as the other game characteristics.

One feature that Baseball4D provides is the opportunity to selectively view and analyze multiple plays at once. Coaches and players often watch video to improve their own play or scout their opponents. For example, an analyst might view a set of ground balls hit to the shortstop and notice that when he moves to his left, his throws to first base tend to more accurate than when he moves to his right. We could also time-shift plays so that we could see balls as they are caught to evaluate the different trajectories and speeds. Figure 1 (right) shows how Baseball4D can present both multiple hits by the same batter and multiple plays by the same fielder.

As with pitch data, we can aggregate all plays to generate heatmaps of hit ball trajectories. The heat map shown in Figure 7, for example, shows all hits and suggests that the 3rd baseman and the shortstop field most of the hits, perhaps suggesting a larger number of right-handed at bats. However, further investigations might combine this aggregation with filtering to confirm differences between left- and right-handed batters or differences between pitchers. Such selection allows more fine-grained analysis and presents the opportunity to find new correlations between aspects of the game.

One interesting direction that uses filtering is to combine pitch characteristics with hit ball trajectories and fielder trajectories. If a manager wants to induce a double-play, for example, he may signal in a particular type of pitch. Analysis of the hit trajectories of that type of pitch might enhance the understanding of how effective such strategies are. Note that a successful double-play depends on the quality of the pitch, the hitter's decision, the ball trajectory, the fielders' initial positions, the baserunners' movements, and the execution of all of the fielders' throws. Thus, understanding hit ball trajectories and fielder movements for different pitch types (see Figure 9) presents a step toward improving our understanding of such complicated plays. In the figures, heat maps show the outcomes of different pitch types, and spray charts show the trajectories of the balls and fielders. Both visualizations may be also filtered by other gameplay attributes including the pitcher name, the batter name, the game, inning, number of outs, pitch result, and pre-pitch balls and strikes.

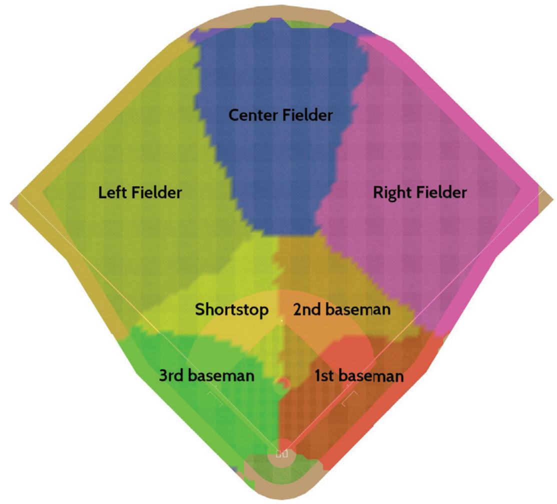


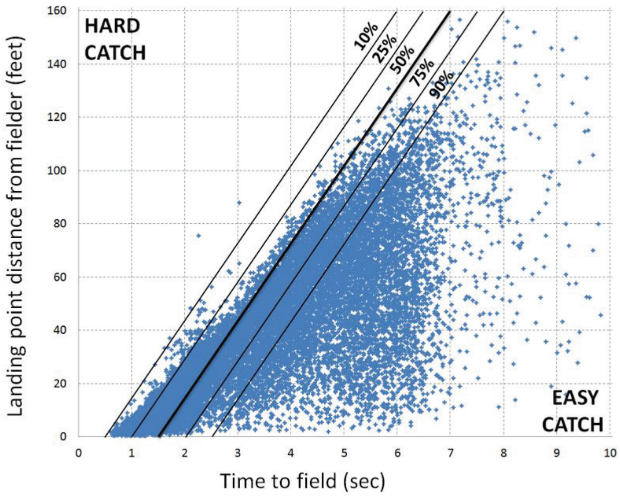
Fig. 10: A data-based partition of which regions a particular fielder is responsible for. Each field point is assigned to the player that made the most catches in that location during the 2013 season. The availability of actual fielding data may help to determine the regions each player is responsible for.

6.3 True Defensive Range: Expanding Defensive Metrics

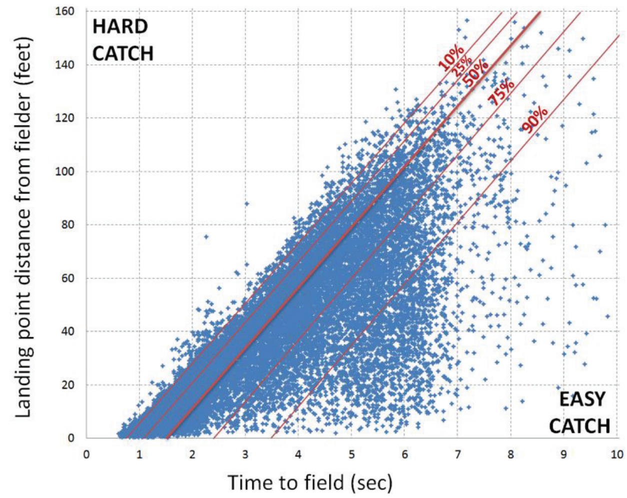
Despite the large number of statistics that allow in-depth analysis of pitching and hitting, it has been much more difficult to evaluate performance once the ball is in play. Often, the number of errors is the only measure available to compare fielding performance. Perhaps some evaluations can be made based on number of outs versus times the ball was hit to a particular region the player was responsible for, but these statistics often lack information about where the ball landed or if the player was positioned according to a particular strategy (*e.g.,* to take away potential extra-base hits). With the reconstruction of entire gameplays, we can compute defensive metrics that evaluate a player's fielding performance. Not only can we see where the ball is hit to but also the speed with which a fielder reacts.

The measurement of defensive skills has been significantly improved in recent years. Defensive metrics can be derived from both defensive events (putout, assists, errors, total chances) and fielding information. The latter has proven to be more a reliable indicator of the fielding ability, but suffers from the lack of accurate data on batted balls (this data is only available since 1989) and players' positions. The early tracking approaches (such as the zone rating of STATS Inc. and Baseball Info Solutions) were based on a discretization of the field into zones. The zones helped the reporters (zone rating operators) to visually determine where the ball landed and what player was supposed to field it (zones were assigned to specific players). The assignment of zones to specific fielders was put to discussion later, however, and the new data shows that players' zones may overlap significantly on the field (see, *e.g.,* Figure 10).

With new tracking abilities, we can better track fielding performance, and this has already led to more discussion and new metrics about defensive skills. Greg Rybarczyk has proposed a measure of



(a)



(b)

Fig. 11: Plots of True Defensive Range (TDR) for all plays in the 2013 season data. The lines show the difficulty levels (probabilities that a hit will be fielded) based on (a) Rybarczyk's estimates and (b) the 2013 fielding data.

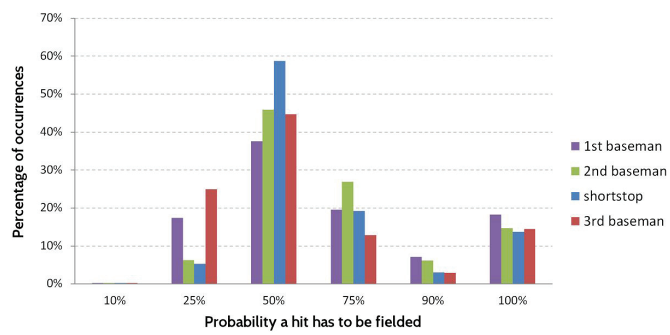
fielding performance termed "True Defensive Range" (TDR) [27]. Instead of tracking when a ball was hit to a given region of the field and if the player was able to catch it for an out, he proposes using the time the ball was in the air and the distance from the fielder's initial position to better characterize each ball hit toward that fielder. From all balls hit to the outfield, for example, we can, given these two measures, determine the probability of a given catch. Then, for each ball hit to a particular fielder, we adjust his TDR based on how easy a catch was and whether he made it. Other proposed metrics include baserunning analysis or evaluating how long it takes an infielder to get to a given ground ball.

Using gameplay snapshots, we can compute fielding metrics like TDR [27] in Baseball4D. From the data, we measure the difference in time between pitch and catch events and the distance from the fielder's initial location based on snapshots. Then, we examine the outcome of the event—whether an out was made. The *difficulty* of a particular time/distance can be set based on the percentage of balls caught for those numbers (2013 data is shown in Figure 11 both based on Rybarzyk's suggestions and using the 2013 data). Then, for each fielder, we can compute his TDR by giving proportional credit for each ball hit to his area (note that we may need to determine these from data as well—see Figure 10). A fielder's score is raised for successful outcomes for difficult outs and lowered for unsuccessful outcomes for easy outs.

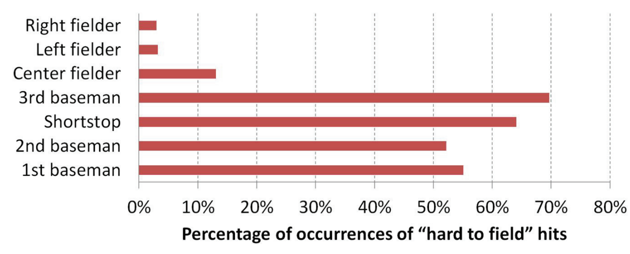
Because we have spatial data about each play, we can also use Baseball4D to visualize TDR, search for patterns, and filter based on particular positions or players. For example, we might evaluate where on the field the difficult plays are made (Figure 13), or how difficult a particular position is (Figure 12). Furthermore, there may also be interesting conclusions that can be drawn about "easy catches" that fall close to a fielder's normal position, or "hard catches" that are actually line drives directly at the fielder. Thus, the spatial component may still play a role in this metric.

The availability of accurate fielder data may be used to review some aspects of the TDR and suggest alternative metrics. Specifically, the relationship between the ball's flight time and the distance the player needs to cover to catch it is actually an indication of the player's speed. The fielder's speed alone may be enough to indicate the difficulty of a hit—if the fielder needs to reach a high speed to field a hit, it probably means that that hit is hard to be fielded. Figure 14 shows that this relationship may be verified spatially.

As noted before [3], the real strength of defensive metrics lies in the combination of box-score statistics and field data. The combination helps to answer questions like how often a play is made for a



(a) Difficulty of plays by infield position



(b) Percentage of difficult plays by infield position

Fig. 12: Measuring the difficulty of balls handled by infielders. By these measures, the 3rd baseman has a slightly more difficult job than other infielders.

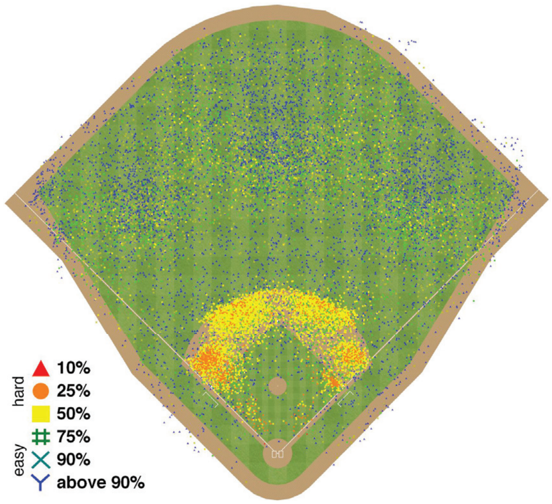


Fig. 13: Plot of TDR difficulty according to the position where the ball landed. Note that the difficulty levels are not linked to specific field positions.

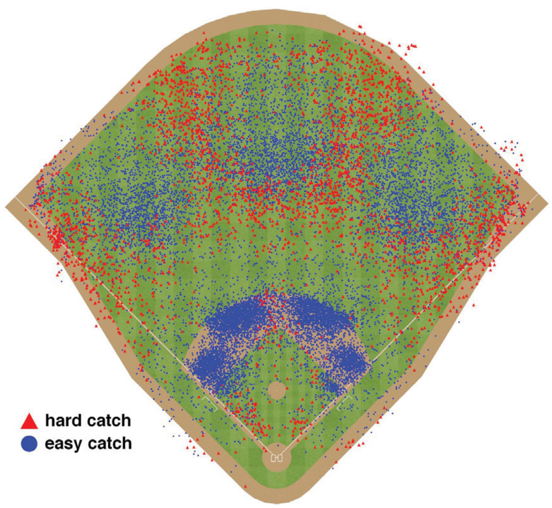


Fig. 14: An alternative characterization of play difficulty might involve the fielder's running speed. Here we plot plays where the fielder ran at least 15 feet/sec (10.2 mph) as hard.

batted ball with a given trajectory and hit location (a contribution of the plus/minus metric). With the advent of fielder tracking technology, the defensive metrics can be computed with large amounts of consistent and accurate field data, and their significance can then be reviewed.

7 Conclusion

In this paper, we present Baseball4D, a new visual analytics tool that has been designed to enable the analysis of high-resolution, time-varying player- and ball-tracking data streams that are becoming available as 3D tracking systems are deployed throughout the world. Baseball4D takes several data streams and consolidates, filters, and enriches them so that gameplay can be more easily studied. In this paper, we show that the consolidated gameplays can be used to generate non-trivial statistics and visualizations that were not possible to be compute before (or at least very hard to do). We have also shown that Baseball4D's intuitive user interface allows users to "slice and dice" games in many ways that were previously impossible.

There are several avenues for future work. We would like to explore adding more metrics to the system, and also develop ways to depict them in an intuitive way. We are also looking forward to having complete coverage of a season. As the amount of data will increase substantially, we will need to revisit some of the components of the system. In particular, Baseball4D currently incrementally loads data as needed, caching it in memory. This approach is not scalable moving forward, and we will need to use an alternative approach. Querying will also need to be improved, and we plan to explore how to couple our system with an efficient database backend. As the data size gets larger, another area that will need to be explored is how to maintain interactive rendering performance. We plan to explore automatic level-of-detail techniques. Also, we would like to add a scripting API to the system that allows the users to implement new analysis techniques without having to use the low-level C++ code that composes most of the system.

Throughout the development of the techniques and tools embedded in Baseball4D, we have been collaborating closely with baseball experts. On purpose, we have refrained from making any mention of specific plays, players, or teams. We do believe that teams and coaches recognize the value of objective data like accurate release points and movement trajectories, and we believe that our work is a step in the right direction toward building tools that will allow these experts to improve baseball statistics and unveil useful information about players and teams.

Acknowledgments

We would like to thank Dirk Van Dall, Cory Schwartz, Andrew Pinter, and Joe Inzerillo and the rest of the talented engineering and stats team at MLB Advanced Media. Without their help and support, this work would not have been possible. We would also like to acknowledge the anonymous reviewers for many helpful suggestions.

References

[1] P.-A. Albinsson and D. Andersson. Extending the attribute explorer to support professional team-sport analysis. *Information Visualization*, 7(2):163–169, 2008.

[2] D. Appelman. Customizable heat maps, Jan 31, 2011. Retrieved July 30, 2014 from `http://www.fangraphs.com/blogs/customizable-heat-maps/`.

[3] D. Basco and J. Zimmerman. *The Baseball Research Journal*, volume 39, chapter Measuring Defense: Entering the Zones of Fielding Statistics. Society for American Baseball Research, 2010.

[4] T. Bebie and H. Bieri. A video-based 3d-reconstruction of soccer games. *Computer Graphics Forum*, 19(3):391–400, 2000.

[5] I. Boudway. Baseball: Running the new numbers. *Bloomberg BusinessWeek*, March 31, 2011. Retrieved July 30, 2014 from `http://www.businessweek.com/magazine/content/11_15/b4223072802462.htm`.

[6] C. Bregler. Motion capture technology for entertainment [in the spotlight]. *Signal Processing Magazine, IEEE*, 24(6):160–158, Nov. 2007.

[7] D. H. S. Chung, P. A. Legg, M. L. Parry, I. W. Griffiths, R. Brown, R. S. Laramee, and M. Chen. Visual analytics for multivariate sorting of sport event data. In *Workshop on Sports Data Visualization*, 2013.

[8] Chyronhego. `http://chyronhego.com/`.

[9] A. Cox and J. Stasko. Sportsvis: Discovering meaning in sports statistics through information visualization. In *Proceedings of Symposium on Information Visualization*, pages 114–115. Citeseer, 2006.

[10] FIELDf/x, Sportvision. `http://www.sportvision.com/baseball/fieldfx`.

[11] K. Goldsberry. Courtvision: New visual and spatial analytics for the nba. In *MIT Sloan Sports Analytics Conference*. MIT Sloan Sports Analytics Conference, 2012.

[12] A. Guéziec. Tracking pitches for broadcast television. *Computer*, 35(3):38–43, Mar. 2002.

[13] A. Hervieu, P. Bouthemy, and J.-P. L. Cadre. Trajectory-based handball video understanding. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '09, pages 43:1–43:8, New York, NY, USA, 2009. ACM.

[14] M. Hferlin, E. Grundy, R. Borgo, D. Weiskopf, M. Chen, I. W. Griffiths, and W. Griffiths. Video visualization for snooker skill training. *Computer Graphics Forum*, 29(3):1053–1062, 2010.

[15] P. A. Legg, D. H. S. Chung, M. L. Parry, M. W. Jones, R. Long, I. W. Griffiths, and M. Chen. Matchpad: Interactive glyph-based visualization for real-time sports performance analysis. *Computer Graphics Forum*, 31(3pt4):1255–1264, 2012.

[16] M. Lewis. *Moneyball: The art of winning an unfair game*. WW Norton, 2004.

[17] D. Magee and J. Perš. Computer vision based analysis in sport environments. *Computer Vision and Image Understanding*, 113(5):589, 2009.

[18] P. E. Meltzer. *So you think you know about baseball?* Norton, 2013.

[19] B. Moon and R. Brath. Bloomberg sports visualization for pitch analysis. In *Workshop on Sports Data Visualization*, 2013.

[20] C. Perin, R. Vuillemot, and J.-D. Fekete. SoccerStories: A kick-off for visual soccer analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2506–2515, 2013.

[21] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš. A trajectory-based analysis of coordinated team activity in a basketball game. *Comput. Vis. Image Underst.*, 113(5):612–621, May 2009.

[22] H. Pileggi, C. D. Stolper, J. M. Boyle, and J. T. Stasko. Snapshot: Visualization to propel ice hockey analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2819–2828, 2012.

[23] G. Pingali, A. Opalach, Y. Jean, and I. Carlbom. Visualization of sports using motion trajectories: providing insights into performance, style, and strategy. In *Proceedings of the conference on Visualization '01*, VIS '01, pages 75–82, Washington, DC, USA, 2001. IEEE Computer Society.

[24] PITCHf/x, Sportvision. `http://www.sportvision.com/baseball/pitchfx`.

[25] PlayItOver. `http://playitover.com/`.

[26] G. Roberts, S. Carter, and J. Ward. How Mariano Rivera dominates hitters. *New York Times Magazine*, June 29, 2010. Retrieved July 30, 2014 from `http://www.nytimes.com/interactive/2010/06/29/magazine/rivera-pitches.html`.

[27] G. Rybarczyk. A FIELDf/X application: True defensive range. In J. Distelheim, B. Taso, J. Oshan, and C. B. Hale, editors, *The Hardball Times Baseball Annual 2011*, pages 115–118. ACTA Sports, 2010.

[28] H. Saito, N. Inamoto, and S. Iwase. Sports scene analysis and visualization from multiple-view video. In *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, volume 2, pages 1395–1398 Vol.2, June.

[29] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[30] N. Silver. *The Signal and the Noise*. Penguin Press, New York, 2012.

[31] Sportvision inc. `http://www.sportvision.com/`.

[32] SportVU, STATS. `http://www.sportvu.com`.

[33] Trackman. `http://trackmangolf.com/company/overview`.

[34] K. Wongsuphasawat and D. Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2659–2668, 2012.

[35] X. Yu and D. Farin. Current and emerging topics in sports video processing. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 526–529, July.