StatCast Dashboard: Exploration of Spatiotemporal Baseball Data

Marcos Lage[1], Jorge Piazentin Ono[2], Daniel Cervone[2], Justin Chiang[3], Carlos Dietrich[2], and Cláudio T. Silva[2]

[1]Universidade Federal Fluminense
[2]New York University
[3]Collegiate School

Abstract

Major League Baseball (MLB) has a long history of providing detailed, high-quality data, leading to a tremendous surge in sports analytics research in recent years. In 2015, MLB.com released StatCast, a novel spatiotemporal data tracking system that has been used in around 2,500 games since its inception to capture player and ball locations, as well as semantically meaningful game events. In this paper, we present visualization and analytics infrastructure to help query and facilitate the analysis of this new tracking data. Our goal is to go beyond descriptive statistics of individual plays, allowing analysts to study diverse collections of games and game events. Our system enables the exploration of the data through a simple yet powerful querying interface and a set of flexible interactive visualization tools.

1 Introduction

Although many sports now use statistics and videos to analyze and improve game play, baseball has led the charge throughout the history of sports analytics. With the advent of new technologies that can track every player and the ball across the entire field, it is now possible to bring the understanding of this game to another level. This past year, MLB.com released StatCast, a novel system that uses player and ball location, and semantically meaningful game events to capture games with unprecedented detail. The StatCast Metrics Engine is one of the key components of the system, which uses discrete locations across time to reconstruct entire baseball games, enabling the computation of new player statistics, such as "route efficiency" or "lead distance", which allow for more detailed and accurate analyses of player and team performance. StatCast has been fully operational across all MLB ballparks since the beginning of 2015, and it has been used to capture around 2,500 games since its inception.

The StatCast infrastructure is a complex system that involves new ballpark sensors, sophisticated software, and a completely new workflow to capture baseball game at a high level of spatiotemporal detail. Games are stored as collections of actions from the beginning to end of the play that are called a *gameplay*; other ancillary data, including videos and metrics generated during each play, are also stored. This wealth of new data is just starting to be used for analysis.

In this paper, we present our work on building visualization and analytics infrastructure to help to study baseball tracking data. Our goal is to go beyond single-play statistics, and to allow for studying a collection of games and analyzing trends. Our system allows the exploration of the data through a simple yet powerful querying interface and a set of flexible interactive visualization tools.

2 Related Work

The area of sports analytics has exploded in the last few years. Now, most teams in baseball and many other sports use statistics to evaluate performance, and there are conferences, including the MIT Sloan Sports Analytics Conference, devoted to sports analysis. Below we give a short introduction to related work, mostly taken from [8], which describes a preliminary version of the StatCast Metrics Engine.

There is 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, and sometimes these cameras work together with manual guidance or annotations. In baseball, Sportvision developed PITCH f/X to capture the full path of the ball from the pitcher's hand to the plate [18]. Meanwhile, FIELD f/X [9] and PlayItOver [19] promise to capture players and the ball throughout the entire game [4]. STATS has developed the SportVU player tracking technology for soccer, basketball, and American football [22].

There has also been work targeting sports in the area of visualization and visual analytics. In 2013, a workshop on sports data visualization was held during IEEE VisWeek and included work on baseball pitch analysis [14]. Pileggi et al. [16] examined the role of visualization in sports analytics and provided a survey of existing work in a range of different sports. Pingali et al. [17] introduced a number of techniques for tennis including virtual replays of serves and coverage maps. Cox and Stasko [6] examined baseball using baseline bar displays and player maps. Wongsuphasawat and Gotz [23] aggregated outcomes of soccer games through clusters and pathways based on events and statistics, and Perin et al. [15] showed how soccer data can be used to create connected visualizations that tell stories about play progressions. Albinsson and Andersson [1] noted the importance of team-sport event analysis and showed how multiple attributes can be explored across histograms and two-dimensional maps via linked views. Legg et al. [12] showed that glyph-based visualization can be used for real-time analysis of rugby matches, and Chung et al. [5] presented a knowledge-assisted sorting system to help users explore game events and video footage.

Heat maps and their derivatives have seen significant adoption in sports visualization and analysis (*e.g.* [7]). 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 [16], and Goldsberry used a scaled-glyph heat map to track

both the density through scales and projected point value of shots through colors in basketball [10].

In addition, both sports fans and the 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. Television networks such as ESPN commissioned methods like "K Zone" [11] 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. As an other example, the *New York Times Magazine* featured a video that visualizes the reconstructed pitches of one famous relief pitcher [20]. Also, the interest in fantasy sports, where fans draft teams earn points based on each player's individual statistics, has led to even greater interest in sports statistics and analytics.

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 players in order to better draft and manage their roster [13].

3 StatCast Overview

The StatCast project is an effort to capture all actions performed in the play field and process them to generate interesting content for the public. The project comprises both the hardware necessary to capture the players and ball behavior as they move over the field (cameras, radars, etc.) and the software layer required to gather and process this information. This section we will focus on the characteristics of the tracking data (the data captured from the players and the ball) and on the processing of that data for the computation of metrics inform the public about the performances players and teams.

3.1 StatCast Player and Ball Tracking

For each play, there are detailed measurements of each players' location, of the ball's position and of the ball-related events (pitch, catches, releases, etc.). These measurements are processed to generate player- and team-level metrics, like, for example, all metrics related to the pitch: pitch release time, speed, back-spin, side-spin, extension, and so on. The center and right images of Figure 1 shows an example play and a list with a subset of its associated metrics.

StatCast works by optically-tracking player location in the field as (x, y, z) measurements at high frame rate (nearly 30 Hz), which gives a detailed description on what the players do on the field. For each player, there is a sequence of positions recorded for the duration of the play. In the StatCast coordinate system, $(0, 0, 0)$ is at home plate where the batter is, the y-axis points to pitcher's mound, and the x-axis points to the right, going parallel to a line between the 3rd and the 1st bases (see the left image on Figure 1 for an illustration).

The player tracking system used is ChyronHego Tracab. This sophisticated system uses three pairs of stereo cameras to track in real-time all the players in the field. The system "compresses" all the information about a player to a two-dimensional point. The process of taking complex

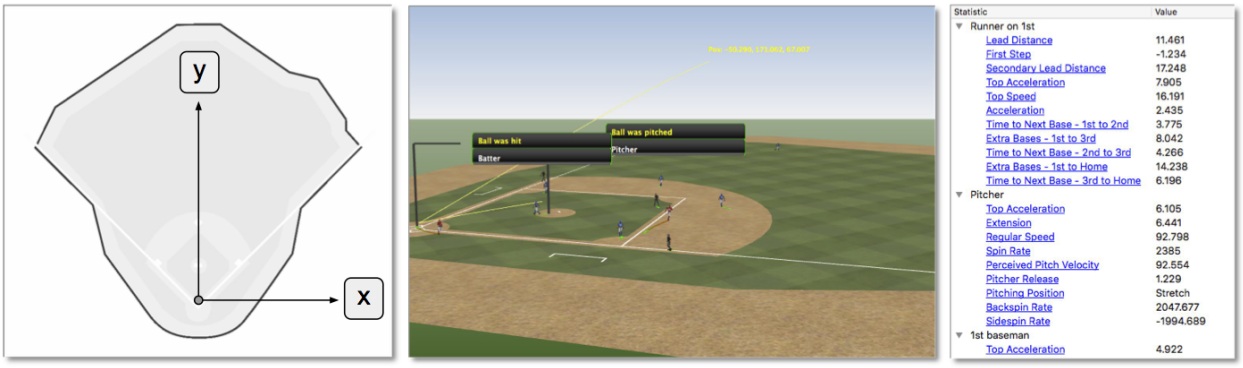

Figure 1: StatCast Overview. Coordinate system (left); Player and ball tracks and game events (center); partial list of metrics computed by the system (right).

human motion in three dimensions and turning into a single two-dimensional point eliminates a good amount of fine details, which obviously limits the kind of movement analysis that can be performed on the output. For instance, consider a runner trying to reach a base by sliding. What (x,y) position should be reported by the system during this movement? Will the player's location ever truly overlap with the geometric position of the base? Answering these questions depends on the exact algorithm used for such a conversion.

One way to achieve this is described by Borg [3] in the context of tracking soccer players. Borg's algorithm uses the pairs of stereo images to compute a set of three-dimensional voxels (*i.e.* a volume element) that belong to each player as they move. This computation involves finding corresponding pixels on the different images and using stereo disparity to recover the depth information of the voxels. These three-dimensional data points are then projected in the field to compute a two-dimensional center of mass of the voxels, which are used as an approximation for the center of mass of the player. Note that player movement affects their "location" even if they are not actually moving at all; imagine a player swinging his arms — the voxels detected by the system will change over time, and their projected location will change as well. One way to think of the tracking algorithm is that it is *integrating* a complex motion into a simpler signal. A detailed analysis of the accuracy and the limits of this system is beyond the scope of this paper, but the knowledge of the measurement process characteristics is crucial to the design of the metrics computation and analysis.

The ball moves much faster than the players, so it requires a higher sampling rate. The system uses Trackman radar technology to track the ball. Using a radar solution has a number of advantages over optical tracking, most notably the higher temporal resolution. This enables the computation of nearly exact timing of significant game events, for instance, when the ball leaves the pitcher's hand, or when it is caught by a fielder. The ball is also optically tracked, and in cases where the radar measurements are unavailable (*e.g.* a rolling ball on the field is hard to track with the radar), the optically tracked positions are used.

3.2 StatCast Metrics Engine

The StatCast Metrics Engine uses discrete positions across time to reconstruct entire baseball games, enabling the computation of new player metrics. The system that is currently deployed is a highly refined version of the one described in [8]. The StatCast Metrics Engine works on top of data about player and ball location, and semantically-meaningful game events. The baseball game is taken as a continuous stream of data and events, including pitches, catches, throws, and player movements. 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. Understanding the state of the ball is critical to determining what part of the play we might be in.

Besides the positioning data, a stream of "game events" is key to being able to reconstruct the gameplay. Game events are the high-level events associated with the ball while it is in play—the moments when a specific player hits, obtains possession, or relinquishes possession of the ball. The data stream is composed of tuples that contain a time stamp, 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".

At a high level, the StatCast Metrics Engine works by filtering the location data merging the game events, and eventually reconstructing a state machine that represents the game. It computes a wide set of metrics for players grouped in different categories, for instance, there are baserunning metrics (*e.g.* player acceleration, speed, and lead distance), fielding metrics (*e.g.* arm strength, pop time, route efficiency), and pitching metrics (*e.g.* extension, release time, spin rate, perceived velocity). We note that the StatCast Metrics Engine is a surprisingly sophisticated piece of software. One of the most complex parts of the system are the data filtering operations. The system performs a substantial amount of error checking, which is used in other parts of StatCast, including informing human operators when parts of the system might be malfunctioning or need user input.

3.3 Building The Analysis Database

In order to build a unified database that enables interactive visualization and exploration of the gameplay tracking data, we merge the StatCast data with the MLB.com *Game Metadata Directory*. The Game Metadata Directory provides metadata information about the baseball games such as the date and time they were played, their scoreboards, players appearances, among others. The data is publicly accessible through a web API.

The Dashboard Database is a document-oriented database that enables fast querying over the derived gameplay statistics and the tracking dataset. Also, in order to facilitate the querying process, we developed a simple keyword-based query system. The Dashboard database is composed of five different document collections, described below.

In the *games* collection, each game is represented by one document and in order to allow for fast queries, we use different game properties as indexes. More specifically, we chose to have the game unique identifier, the team names, and the game date as indexes since we were interested in

filtering games by dates and involved teams or analyzing the data produced during a specific match. We also have corresponding data collections for *players* and *lineups*. Since baseball analysis is often focused on pitcher-batter match ups and we are interested in studying optimized indexes for pitchers, batters and fielders.

The last and largest part of our analysis database is based on the output of the StatCast Metrics Engine on each gameplay. The StatCast Metrics Engine outputs a clean and filtered dataset that contains all of the positioning data for the players, the ball, and related game events and metrics in a normalized format. We saved this set of *tracking* information and related *metrics* in our database as two separate document collections.

The StatCast tracking data is a large dataset. Each of MLB's thirty teams plays 162 games during the regular season and a typical game has between three hundred and four hundred plays, so the total number of plays recorded during the season is in the order of 700,000 plays that occupy 1.5 terabytes on disk.

4 StatCast Dashboard

The purpose of the StatCast Dashboard is to make it easier to explore and analyze the StatCast data. The complexity and richness of the data will certainly require different visual user interfaces depending on the target audience and applications, but at this point, our goal was to build a system that enables initial studies of the data. More precisely, we decided to design a web-based system that allows for an easy way to query, filter, and explore a large collection of gameplays, the related metrics computed by the StatCast Metrics Engine, and the tracking data. Here we list the specific tasks we wanted to enable a user to do:

T1: The user should be able to query gameplays based on a flexible selection mechanism, without the need to knowledge of programming or database technology. For example, the system should allow the user to select gameplays played by Michael Pineda from August 2015 to October 2015 or gameplays from games between Mets and Yankees that Juan Lagares played as a fielder.

T2: The user should be able to spatially visualize individual gameplays or groups of gameplays. The visualization of individual gameplays is important since it gives the user the ability to study the behavior of players and the ball. The visualization of groups of gameplays provides a global understanding of the patterns and can help the study of game strategies.

T3: The user should be able to interactively filter the queried gameplays based on the metrics computed by the StatCast Metrics Engine. That is especially useful when the user is interested in performing comparisons of the type: which right fielders had the highest *Top Speed* values or which pitchers had an *Extension* in a range of interest.

T4: The user should be able to draw over the field to interactively define regions that in turn directly filter the queried gameplays. That is, the system should enable direct interaction with the tracking data. For example, the user can find gameplays where the center fielder ran to their left to catch the ball.

T5: Finally, the user should be able to export data, that is, the metrics and the tracking information of the gameplays that were selected. This functionality allows the user to continue the analysis of the identified gameplays using other tools like R or Python.

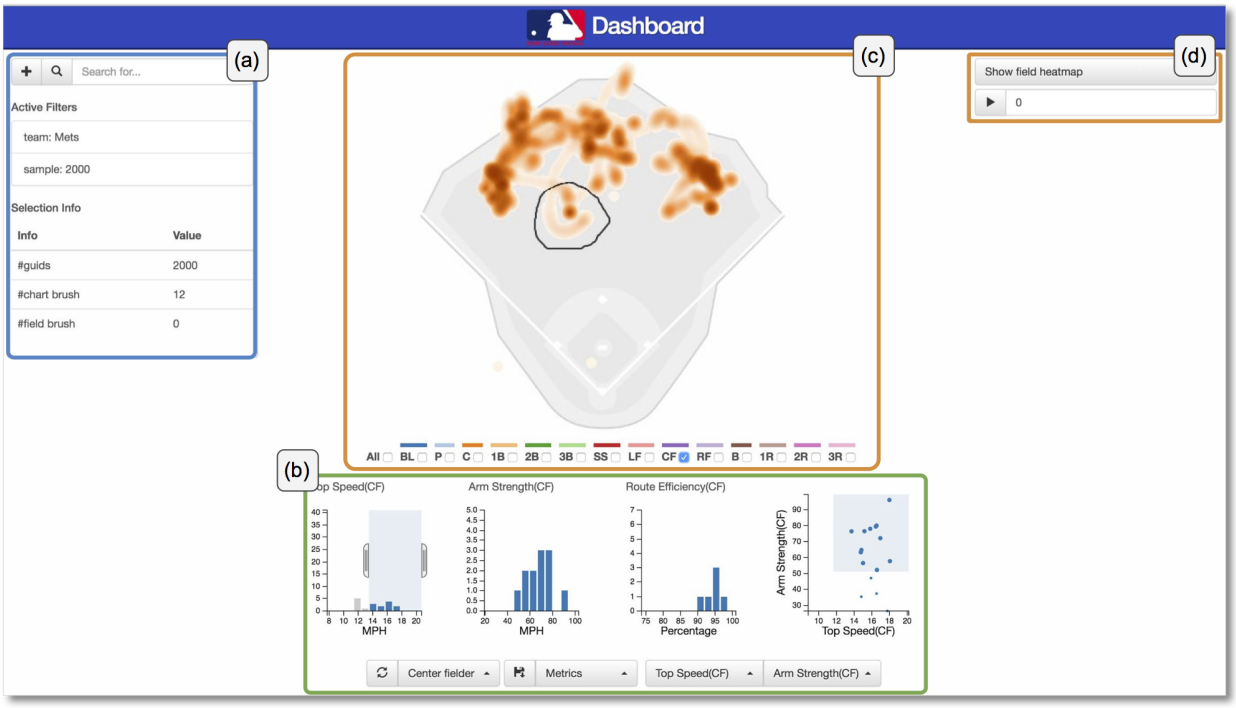

Figure 2: The StatCast Dashboard visual interface is composed of three core components, the Keywords Manager **(a)**, the Statistics Viewer **(b)** and the Gameplay Viewer **(c-d)**, that can be used by the user to query, filter, explore and export gameplays from the StatCast dataset. The **VIDEO** submitted with our paper shows the system in action.

To address the previous tasks, we designed a visual user interface for the StatCast Dashboard composed of three core components: the Keywords Manager, the Statistics Viewer and the Gameplay Viewer. The current version of the visual interface of the StatCast Dashboard is shown in Figure 2. In the following we will give a description of the main system interactions available through this interface, the strategies used to efficiently implement the available interactions and the interface components associated with each interaction.

4.1 Querying Gameplays

To facilitate the access to the database and address task T1, we developed a keyword based querying strategy, that allows for users without programming knowledge to use the Dashboard and select gameplays from StatCast. The keyword query system is based on a set of available keywords and a list of active keywords instances. To create a new keyword instance, the user needs to write using the syntax `keyword:value1[,value2,...,valueN]`. Currently, the Dashboard supports ten different keywords: `team`, `vs`, `date`, `pitcher`, `batter`, `fielder`, `game_id`, `play_id`, `limit`, and `sample`.

The `team` keyword can be used to query gameplay data that involves a given team. For example, if one wants to load the gameplays played by the Boston Red Sox (both batting and fielding)

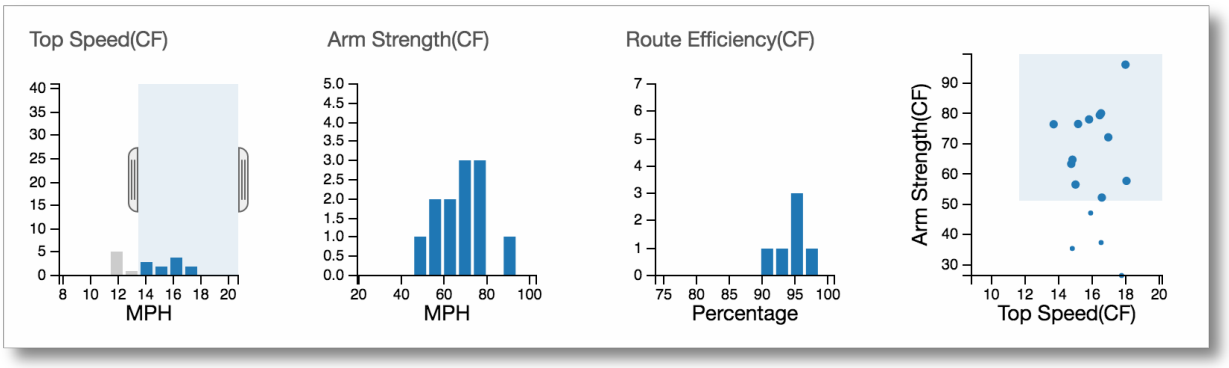

Figure 3: The Metrics Viewer is composed of a set of histograms and a scatter plot. The histograms shows the distribution of the available metrics on the gameplays returned by the query. The scatter plot shows the correlation between two of the available metrics. Both the histograms and the scatter plot can be brushed to filter the gameplays (light blue regions).

he needs to add the keyword instance `team:Red Sox` to the keyword list. The `vs` keyword can be used to find gameplays of a user defined match. If one wants to analyse gameplays of games between New York Yankees and New York Mets, he needs to create the keyword instance `vs:Yankees, Mets`. We observe that the order of the teams is not important in this example. Similarly, the user can specify queries with the dates, the pitchers, the batters, the fielders, the game and gameplay identifiers of interest.

The `limit` and `sample` keywords are used to reduce the amount of data allowed to be returned from a query. The first limits the current query to the first *n* returned gameplays and the second samples the query result uniformly to get *n* distributed gameplays. We also allow the use of wildcards and regular expressions. For example, one can write `pitcher:.*Pineda` to query Michael Pineda gameplays. For convenience, we use Python syntax for the regular expressions.

Keywords Manager. The Keywords Manager (shown in Figure 2a) is a widget that is dynamically updated whenever the user adds or removes a keyword from the actual list of active keywords used to describe a query. The user can add a new keyword to the list writing it in the text input and pressing the *add button*, the button with the plus sign icon shown in Figure 2a, or pressing return in the keyboard. To remove a keyword from the list the user simply needs to click on the undesired item. Once the list of keywords to be composed is done, the user can query the database using the *query button*, the button with the magnifier icon shown in Figure 2a. The keyword based queries in the list are them composed using an AND logical operator. For example, if the keywords manager has two active items, `team:Padres`, `batter:.*Amarista`, the query will return all gameplays that were played by San Diego Padres and has Alexi Amarista as the batter. The Keywords manager also contains an information panel that shows the number of gameplays returned by the executed query and selected using the metrics and the spatial filters that will be described next.

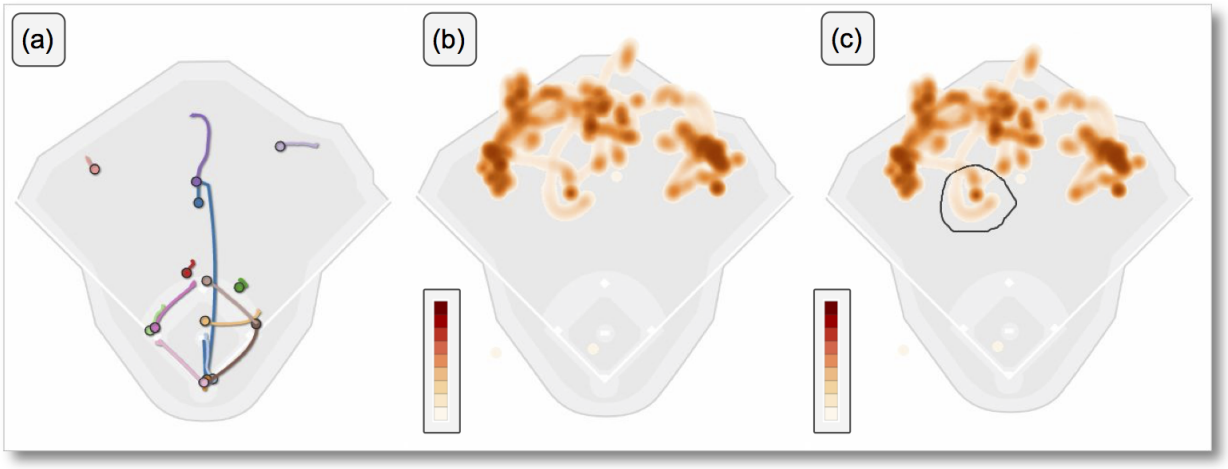

Figure 4: The Gameplay Viewer can be used to visualize the tracking data or filtering the currently selected gameplays using a spatial criteria. **(a)** Shows the individual gameplay visualization mode that can be used to review the behaviour of the players and the ball (shown in blue) through the rendering of their paths. **(b)** Shows the multiple gameplays visualization mode that can be used to understand global patterns on the positioning of the center field using a heat map. **(c)** Shows the user-defined spatial selection criteria represented by the region in black.

4.2 Filtering Gameplays

Once a query is performed, in order to allow the user to filter and explore the returned data, addressing tasks T3 and T4, we developed two components of the StatCast Dashboard user interface: the Metrics Viewer and the Gameplay Viewer. Also, to address the requirements described by task T2, the Gameplay Viewer supports two operation modes.

Metrics Viewer. The Metrics Viewer (shown in Figure 2b and highlighted on Figure 3) is composed of a set of histograms and a scatter plot. Each histogram shows the distribution of a particular metric from the currently loaded set of gameplays. Since the metrics computed in a baseball game are closely related to the player position, we also provide a category selector that allows for the user to change the set of histograms currently displayed. We also provide selectors to allow the user to change the axes of the scatter plot. Using the charts, the user can filter the loaded gameplays by brushing the desired ranges of each statistics that he is interested in. For example, if the user selects the gameplays where a fielder had a speed between 14 and 20 miles per hour, he has to click and drag the mouse over the interval on the *top speed* histogram of the fielder category (left chart, Figure 3). Also, the user can brush the scatter plot to filter gameplays (right chart, Figure 3).

After filtering the gameplays based on the desired statistics, the user can press the *load tracking data* button, the one with the arrows icon shown in Figure 2b, to load the tracking data of the filtered gameplays on the Gameplay Viewer.

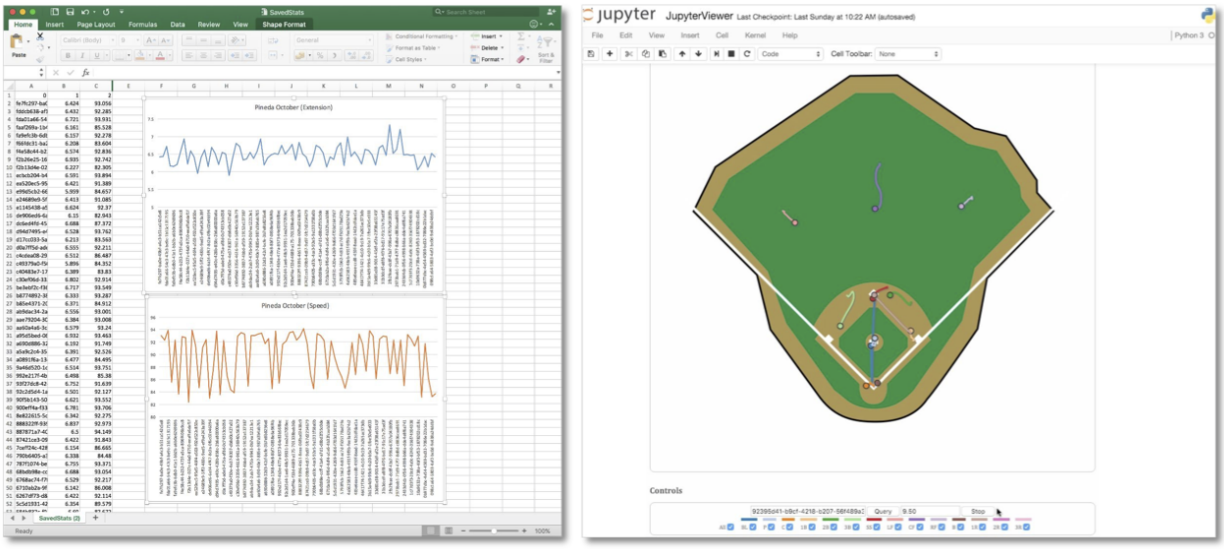

Figure 5: Detailed analysis of data. Left image: The dataset queried and filtered using the StatCast Dashboard can be exported in the `csv` format and used to perform analytics in external software. Right Image: The widgets of the StatCast Dashboard interface can be used inside Jupyter.

Gameplay Viewer. The Gameplay Viewer (shown in Figure 2c) can be used both to visualize the tracking data and to filter the gameplays currently loaded on the field based on a user-defined spatial criteria. The widget presents a top view of the baseball field that is used as render context to display the gameplay tracking data. The tracking data can be visualized over the field using two different modes. The individual gameplay visualization mode shows the paths stored on the StatCast tracking dataset using poly lines (see Figure 4a) and the group gameplay visualization mode uses a heat map to show multiple gameplays' tracking information at the same time (see Figure 4b-c). The player positions to be considered on both visualization modes can be chosen using the positions selector right below the field diamond (see Figure 2c). For example, if one wants to visualize the fielder's movement on a set of gameplays, he needs to mark the left, center and right fielders check-boxes.

Using the field, the user can also define polygons of interest to filter the currently loaded tracking data. In order to do that, the user just needs to click and drag the mouse over the field to draw the desired geometry. Once the geometry is defined, all tracking points that fall inside the polygon are identified and their associated gameplays are stored. Since the number of tracking points on a set of gameplays is typically huge, the selection of tracking points inside the user defined polygon is optimized by the use of a kd-tree, a generalization of a binary search tree that stores points in k-dimensional space [21]. Every time a set of gameplays is loaded on the Gameplay Viewer, each tracking point present in the gameplay is added to the kd-tree. Once the kd-tree is built and the user defines the interest region, the bounding box of the region is computed and used to select the tracking points that can potentially be inside the region and the final set of points is selected performing a point inside polygon geometric test.

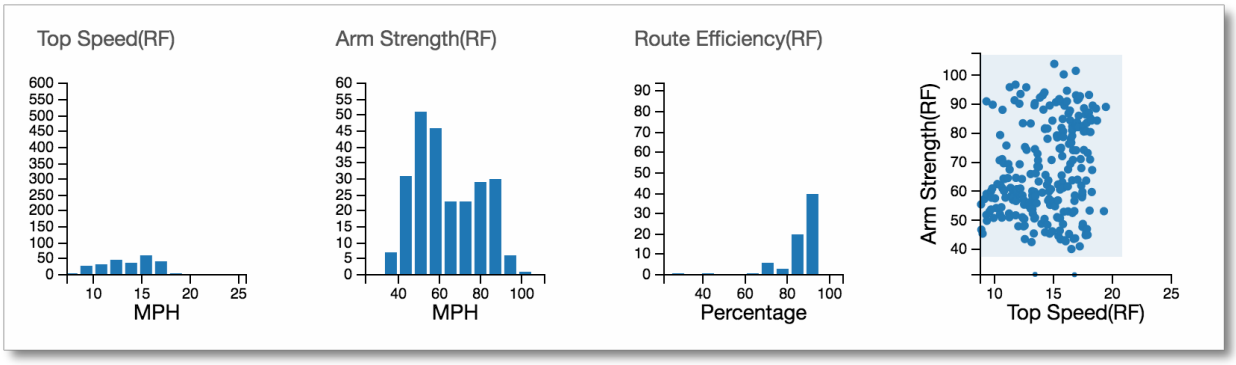

Figure 6: In a sample use place, we want to find all plays during the season in which Bryce Harper fielded the ball. We can do this by selecting all plays for which is arm strength and top running speed are non-null.

4.3 Detailed Analysis of Data

We realize the need to drill down into the data for the purposes of performing detailed statistical analysis. We kept this in mind as we designed the StatCast Dashboard, and in particular we decided to enable flexible data paths for analytics using other tools. To achieve that, the metrics and the tracking information of gameplays selected using the StatCast Dashboard visual interface can be downloaded in `csv` format to be exported to other systems. The left hand side image on Figure 5 shows the pitch metrics from Michael Pineda in October loaded on Microsoft Excel. Also, the widgets we implemented in the StatCast Dashboard can all be embedded in Jupyter, which supports R and python for further analysis. This enables detailed analysis of all the aspects of the data, including manipulations of the wealth of numerical data as well as the use of graphical widgets for display and data selection and filtering. The right hand side image on Figure 5 shows the tracking data of a gameplay loaded on Jupyter using the StatCast Dashboard Gameplay Viewer. Because the Dashboard data is hosted using MongoDB, the MongoClient function from the `pymongo` module can access all the separate data collections, including the games collection, the tracking data, and the metrics collections. By using the Dashboard to collect the file names of the relevant plays, Python code can parse out data and create aggregate statistics such as averages or outliers of the data. Data can be processed by efficient Python libraries such as *e.g.* `matplotlib`, a rich plotting library can be used for graphing the data; or `scikit-learn`, which can be used to perform higher level machine learning functions such as clustering the data. This flexibility was used to produce the use case presented in Section 5.

5 Example Use Case

This section provides an example of how a quantitative researcher could use the dashboard to explore, query, and export data for statistical analysis. An exciting new application of the StatCast data is the ability to more precisely quantify and assess players' defensive abilities. For decades, baseball's defensive metrics depended entirely on coarse game event data, such as the number of

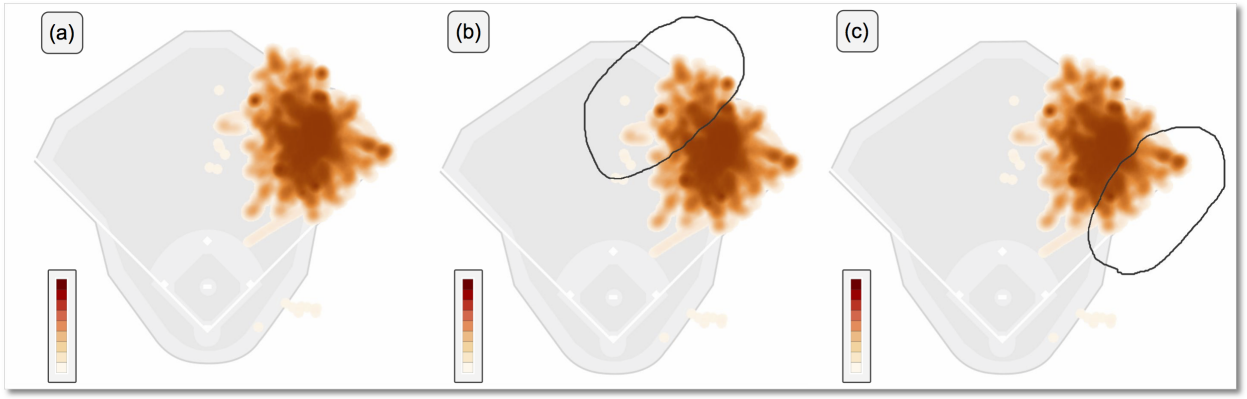

Figure 7: We can further refine our query on plays involving Bryce Harper using the Gameplay Viewer. Here, we split Harper's plays based on whether he moves to his left or to his right. The results of these sub-queries can be exported and analyzed for spatial variation in Harper's play characteristics.

balls a player caught (or didn't catch), the number of fielding errors he committed, and the number of baserunners he threw out. In recent years, defensive metrics have evolved to include subjective assessments from gameplay analysts, as well as batted ball trajectory information. StatCast, by tracking all nine fielders and the ball at high spatiotemporal resolution, provides data for novel defensive analyses based on positioning, reaction time, speed and route efficiency, and arm strength.

For example, we can analyze the positioning of Washington Nationals' right fielder Bryce Harper. Using the dashboard, we first find all plays involving Harper by querying and using a player filter for the right fielder. Metrics for the right fielder, such as top speed and arm strength, are only computed when that player is involved in the play, which in the case of Harper would require him to field the ball during the play. Thus, using the Statistics Viewer, we can select plays where metrics such as arm strength and top speed are non-null (see Figure 6). This produces a sample of all plays during the season in which Bryce Harper fielded the ball.

The Gameplay Viewer can be used to further refine this query and study variation in fielding metrics as a function of space. This variation can reveal inefficiencies in where a fielder is usually positioned—for instance, if a player is slower running to his right than to his left (which is possible, as players need to lead with their non-dominant arm for maximum range when attempting a catch), then he should be placed so that he runs right less frequently than left. For Bryce Harper, we see that for 107 plays, he ran to his left, whereas for 113 plays he ran right (see Figure 7). For each spatial query, we can export a `csv` containing the available metrics for each play in the query results, which enables myriad downstream statistical analyses.

With Harper's data, we see nearly equal maximum top speeds of 19.45 mph and 19.29 mph for moving right and moving left, respectively. Likewise, Harper shows equal top arm strength (101.5 mph) from both regions of the field. Combined with the nearly even split (113 to 107) or plays in each direction, we see no evidence that Harper is systematically sub-optimally positioned in the outfield, though the analysis performed was quite cursory.

6 Discussion

StatCast is a first-of-a-kind system. The system uses novel sensors, state-of-the-art game reconstruction software, and technology that glues everything in an end-to-end system (including linking videos of the actual plays) to create a "season library" of unprecedented detail.

In this paper, we describe our first attempts at building an analytics and visualization stack to go with the rest of the system. We also report on a typical analytics workflow.

Many of the "bugs" have been worked out of the initial system, and we are currently at the start of the second season. We believe new metrics will continue to be added, and they will likely to be used more often to highlight exceptional game play. As we are studying the data, we are discovering new ways to use it, sometimes for purposes that we did not anticipate was possible. All of this is exciting, and we hope it will incredibly change the sports we all treasure and love.

Acknowledgment

We would like to thank the team at MLB.com that worked with us on StatCast, in particular Dirk Van Dall, Greg Cain, Rob Engel, Jeremy Braff, Emily Voytek, Cory Schwartz, Andrew Pinter, and Joe Inzerillo and the rest of the talented engineering and stats team at MLB Advanced Media. We would also like to thank David Koop and Huy Vo, who worked with us on Baseball4D [8]. This work was partially supported by MLB.com, the Moore-Sloan Data Science Environment at NYU, the NYU Tandon School of Engineering, NSF awards CNS-1229185 and CCF-1533564, CNPq (Brazil), and FAPERJ (Brazil).

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] J. Borg. Detecting and Tracking Players in Football Using Stereo Vision. Master's thesis, Department of Electrical Engineering, Linköping University, Sweden, 2007.

[4] 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`.

[5] 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.

[6] 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.

[7] J. Cross and D. Sylvan. Modeling spatial batting ability using a known covariance matrix. *Journal of Quantitative Analysis in Sports*, 11(3):155–167, 2015.

[8] C. Dietrich, D. Koop, H. Vo, and C. Silva. Baseball4D: A tool for baseball game reconstruction and visualization. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 23–32, Oct 2014.

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

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

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

[12] 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.

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

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

[15] 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.

[16] 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.

[17] 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.

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

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

[20] 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`.

[21] H. Samet. *The design and analysis of spatial data structures*, volume 199. Addison-Wesley Reading, MA, 1990.

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

[23] 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.