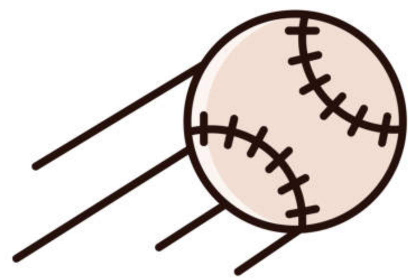
Computer Vision in Baseball

The Evolution of Statcast

Luke McElroy



Computer Science Capstone
Saint John's Unviersity
December 2022

Abstract

Computer vision is being used in sports to gain competitive advantages and enhance fan experiences. Major League Baseball is incorporating new computer vision technologies to its Statcast system with the goal of bringing more analytical data to the game. This paper analyzes the new technologies brought in, how they obtain data through cameras and algorithms, and what the data is being used for. Technical aspects discussed include the creation of point clouds by using the ICP algorithm, Kalman filtering, and position tracking using cameras. Due to the multitude of new data brought in with these techniques, players are getting much better and competitiveness is rising. To conclude this paper, we will look specifically at how pitchers are using this data to get better and how Major League Baseball is using this data to create better fan experiences for the future.

CONTENTS

LIST OF FIGURES

I. Introduction

Computer vision is a field of artificial intelligence that trains computers to track and interpret the real world. Computer vision is applied to many fields and the sports world is one that reaps its benefits. Major League Baseball (MLB) has been using computer vision to track velocity, spin rate, movement, and many other factors over the past 15 years. This article discusses the technology behind tracking this data, algorithms used to obtain the data, what the data is used for, and how computer vision technologies have grown throughout the years in the MLB.

II. Basic Measurements

Two of the most basic measurements that are calculated at all levels of baseball are velocity– how fast the ball is traveling, and movement– what direction the ball is going in. These are obtained using through the use of radar and LiDAR guns.

A. Radar Guns

Radar, which stands for radio detection and ranging, emits a radio wave and upon impact with an object the wave bounces back to the radar gun. These radio waves have frequencies which are used to calculate the speed and direction of an object. Waves that reflect off an object moving away from the radar gun will return at a lower frequency, and waves reflecting off an object moving towards the radar gun will return at a higher frequency [11].

B. LiDAR Scans

LiDAR, which stands for light detection and ranging, works in the same way as radar except it sends a laser which is reflected rather than a radio wave [16]. By multiplying the time it takes for the laser to reflect back to the sensor with the speed of light, the distance from the LiDAR sensor to the object can be calculated [11]. “These guns take several hundred samples per second, and they are extremely accurate” [11]. The many distance samples are then compared to calculate the speed of the object.

III. Statcast

Thanks to the data that radar and LiDAR guns are able to produce, in 2015 the MLB launched its Statcast system. This system provided a way for players and fans to compare data such as, “batters sprint times and pitch distances, and even learn the spin rate for pitchers’ curveballs” [3]. As technology grew, so did the Statcast system along with the technical aspects used within it.

IV. The Old Statcast

In the original Statcast system, the MLB used two cameras combined with its radar and LiDAR scans to locate a baseballs position. The cameras were located behind home plate and behind first base. A large challenge in tracking these baseballs was the various light conditions that occur throughout a game. Games can be sunny, rainy, at night, and there are many different background subjects that can interfere with the detection of a baseball, as seen in Figure 1. Due to these factors, the MLB tracking algorithm did not solely use pattern matching, a common process to detect objects in images, but instead used known properties of a ball’s flight to infer its path [5]. André Guéziec, CEO of Triangle Software, wrote about this process. When tracking a baseball, the algorithms look for a potential ball position by comparing pixels in multiple images and locating significant differences in these images. For example, in one picture a group of pixels may be green, representing the grass of the field. A few pictures later when the ball crosses that point, those same pixels are now white representing a significant difference. “If the current pixel deviates significantly from a predicted value, the system registers the change as motion” [5]. This projected trajectory of the ball is then fitted with the actual existing trajectory by using another process called Kalman filtering.

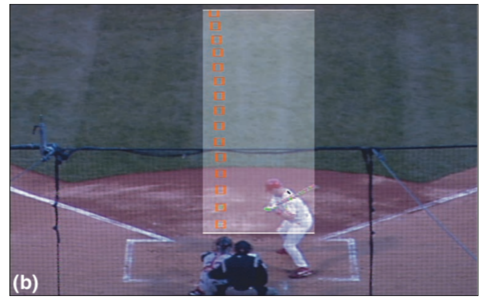


Figure 1. As a pitch is tracked there are various backgrounds that it travels through. Four examples shown in this image are green grass, brown dirt, shadows, and the black webbing of nets which all can interfere with a cameras ability to track a baseball. [5]

A. Kalman Filtering

Kalman filtering, invented in 1960 by Rudolph E Kalman [5], is an algorithm for estimating the state of a system using past observations and current measurements. In simpler terms, “Kalman filtering can be seen as a particular approach to combining approximations of an unknown value to produce a better approximation” [14]. There are two steps that this system uses. First, the prediction step, finds a plausible trajectory for the ball. “To acquire a plausible trajectory, the system delays the image processing by a small number of fields so that the algorithm can look both ahead and back a few fields — corresponding to a timelapse of perhaps one tenth of a second — for potential ball positions that form a consistent trajectory” [5]. Once this trajectory is found, step two, the update step, takes the measurements from the next image and tests them to see how they best fit with the existing trajectory. This process recursively repeats during the entire flight path of the ball, resulting in a tracked play [20] [14]. This is not only used to track pitches but also to track high flyballs. With only two cameras used in the early Statcast, flyballs that are hit high into the air can leave the cameras view and the flight path must be predicted until the ball returns to the cameras view. According to MLB’s data science team, this system was “accurate within 1.5 inches on pitches and 15 feet on batted-ball locations” [8].

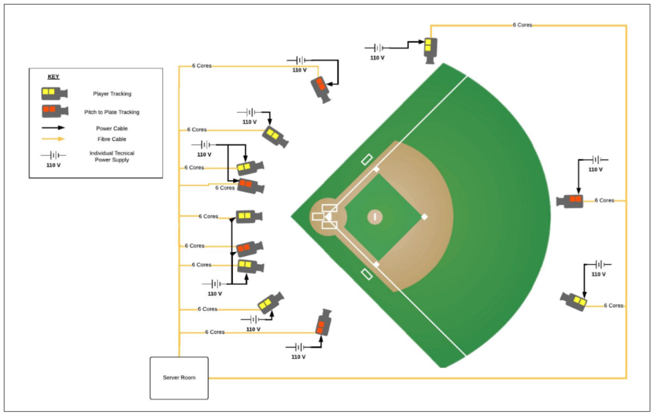


Figure 2. Locations of the 12 cameras in the new Statcast system [7]

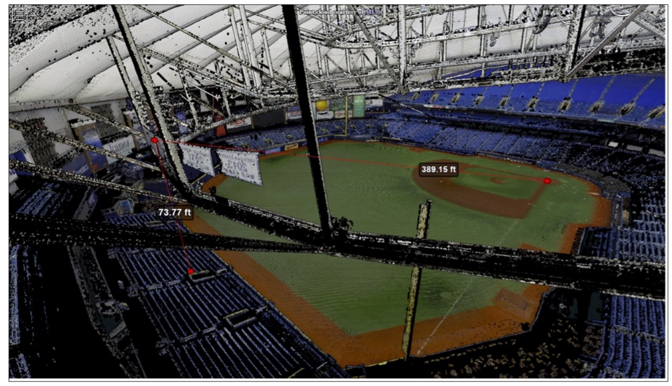


Figure 3. This is a completed point cloud of Tropicana Park, home of the Tampa Bay Rays. Image is taken from Clay Nunnally's study on LiDAR trajectory tracking. [12]

V. The New Statcast

Knowing that technology would improve, the MLB put a 5 year lifespan on the original Statcast system. Greg Cain, MLB's VP of Baseball Data, is quoted saying, "We also knew technology would advance in this space at a rapid pace, so giving the system a five-year lifespan would allow for us to bring in the latest and greatest technology, to extend on the baseline of the 2015 launch" [3]. In 2020, the MLB partnered with Hawk-Eye innovations, a leading company in broadcast enhancement technologies, to create the new Statcast system.

A. The Hawk-Eye System

Jason Gaedtke, MLB's chief technology officer, gives insight to the new technological changes. "Hawk-Eye incorporates LIDAR technology, similar to the type used in autonomous vehicles. The position and movements of everyone on the field – from pitchers to baserunners to umpires – will be updated 30 times per second at 18 different data points on each person" [10]. One of the main changes enabling this data to be collected is that Hawk-Eye uses 12 cameras in every stadium, rather than two as the old system did. Five cameras are dedicated to pitch tracking and the other seven are dedicated to tracking players and batted balls, as seen in Figure 2 [7]. These cameras allow for a much higher accuracy than the old system, reducing the margin of error to less than 0.5 inches on pitches and 1 foot on batted balls [8]. Another benefit is that instead of using Kalman Filtering to predict measurements of the ball, "The Statcast 2020 system is also able to directly measure both pitch spin rate and spin axis rather than inferring one or both measurements from the observed path of the ball as previous systems have done. This enhancement leads to greater accuracy and opens up new doors for pitch design and development" [7]. This is a huge improvement, much of which can be attributed to LiDAR point clouds.

B. LiDAR Point Clouds

A LiDAR point cloud is a 3D representation of a LiDAR scan containing millions of data points (See Figure 3). Along with the LiDAR scans that are performed during games from the cameras in the stadium to track velocity, movement, and spin rate, MLB also "regularly performs LiDAR scans of all 30 MLB ballparks" [12]. Each scan produces a single point cloud, which are so accurate that they can detect individual people, banners, seats, and any visual reference in the stadium. As Clay Nunnaly, a data scientist for the MLB, says, "A relative measurement within this point cloud is accurate with ISO traceability down to 3 mm" [12]. The problem is that a single scan can't capture an entire MLB stadium – the stadiums are just too big. Many scans from different positions must be made and the individual point clouds from each scan must be combined into one massive point cloud that represents the entire stadium [2]. To do this, computers must estimate the transformation of moving one point cloud onto the other so that they are aligned, essentially moving the clouds so that the same points in the real world are overlapping. The most common process to do this is the iterative closest point algorithm (ICP).

C. The ICP Algorithm

Cyrill Stachniss, a photogrammetry and robotics professor at the University of Bonn, walks through the ICP algorithm and how it's used with point clouds in his YouTube tutorial [19]. This tutorial, along with knowledge gained from other sources [13] [6] [4], allowed me to understand the steps to the ICP process. There are two assumptions that must be made to effectively use the algorithm. The first is to assume that only two point clouds are used. In our case of scanning a baseball stadium, this works in real time. The first LiDAR scan generates a point cloud, and then the second scan generates another point cloud which are then combined using the ICP algorithm. Then the third scan generates a third point cloud, which is combined with the product of the first two. This process repeats until all scans are completed and combined, generating a single point cloud of the entire stadium. The second assumption is that we have known correspondences – we know what points in cloud one represent the same points in cloud two. For example, if scan one and scan two both have the scoreboard in it, we can determine the corners of the scoreboard in both images and know they are the same point in real life, thus corresponding to each other. Correspondents don't need to be known for all the points, but at least a sufficient enough amount to make real connections.

Within the ICP algorithm, we start with two point clouds, say x and y, and their correspondences.

$$\text{POINT CLOUD X} = \{X_1 \ldots X_J\} \tag{1}$$

$$\text{POINT CLOUD Y} = \{Y_1 \ldots Y_I\} \tag{2}$$

$$\text{CORRESPONDENCES C} = \{(i,j)\} \tag{3}$$

The goal is to find a translation (t) and rotation (R) that will minimize the sum of squared errors (Euclidean distance) of the distances between points in Cloud X and Cloud Y by using this formula.

$$\sum_{(i,j)\in C} (Y_i - Rx_j - t)^2 \tag{4}$$

A key next step in this algorithm is to simplify the point cloud sets into only the points that have a correspondence. Many points will not have correspondence, for example one LiDAR scan may be set at an angle to see behind a wall, whereas the other LiDAR scan is positioned in a way that the same wall is blocking the view behind it. All of the data points behind the wall will only be seen in one scan, so they will have no corresponding point in the other scan. The result of this simplification is two point cloud sets, $\{x_n\}$ and $\{y_n\}$, that are in order of correspondence. Point x_1 corresponds to point y_1, x_2 corresponds to y_2, and so on. From here, a simplified version of Equation 4 can be made which is called the rigid body transform. "By definition, a rigid body transform is a mapping from this set to another subset of the Euclidean space, such that the Euclidean distances between points are preserved. Any such mapping can be represented as a composition of one translation and one rotation" [9].

$$\bar{x}_n = Rx_n + t \tag{5}$$

This rigid body transform formula takes the points in point cloud X, rotates and translates them to a new point cloud, $\bar{x}$, all while preserving the Euclidean distances between points. The goal is that these new values will minimize the sum of squared errors between points, combining the point clouds.

Lucky for us, Stachniss explains that there is a direct optimal solution that exists to find the rotation value (R). Direct means that no initial guess is needed and optimal means that no better solution exists [19]. To find the rotation matrix R, the center of masses of both point clouds are found using the following formulas.

$$Y_0 = \frac{\sum Y_n P_n}{\sum P_n} \tag{6}$$

$$X_0 = \frac{\sum X_n P_n}{\sum P_n} \tag{7}$$

For the rest of these formulas, P_n represents correspondence weights of how certain we are that two points correspond together. Once the center of masses are obtained they are used to compute a cross covariance matrix H.

$$H = \sum (y_n - y_0)(x_n - x_0)^T p_n \tag{8}$$

Matrix H is then used to compute a singular value decomposition (SVD) of H, which is a standard mathematical procedure used to decompose H into three separate matrices U, D, and V^T.

$$svd(H) = UDV^T \tag{9}$$

In this formula, U and V represent the rotational matrices and D represents the diagonal matrix. From the SVD the optimal rotation can be extracted by taking V times U transposed, which is a flipped version of U, switching the rows and columns.

$$R = VU^T \tag{10}$$

Now that the optimal rotation matrix (R) is found, the translation vector (t) can be computed. There are three components to the formula - the center of masses (x_0 and y_0) and the rotation matrix (R).

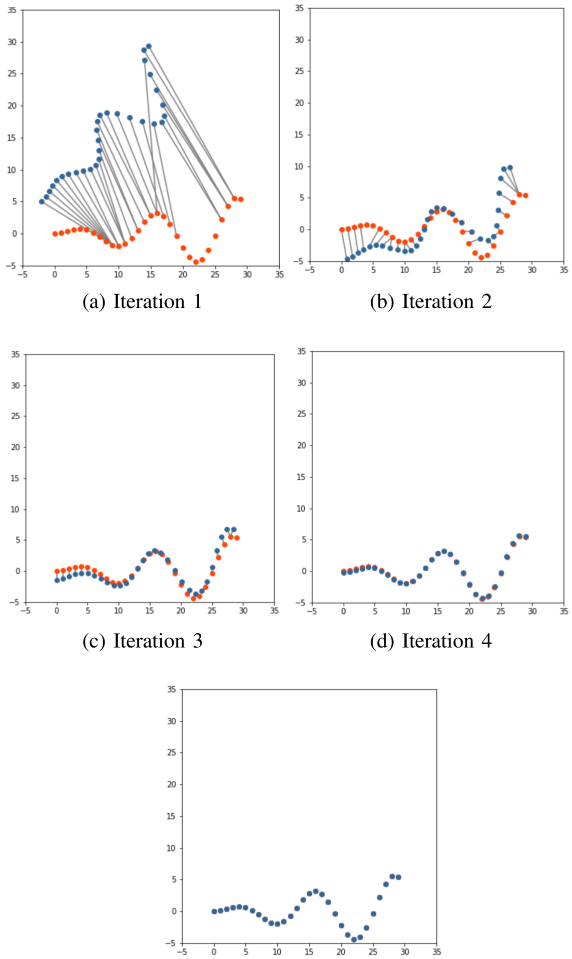
$$t = y_o – R(x_o) \tag{11}$$

Finally, the rotation and translation values are obtained and the point cloud X can be moved on the point cloud Y. The process can be repeated, recomputing the data association and alignment over and over again until the clouds converge and are aligned.

Figure 4 shows the ICP algorithm in use over many iterations. Since all of these mathematical formulas can be quite confusing, Stachniss created interactive code that walks through the steps of the ICP algorithm with visual examples [18]. Plot (a) shows the original corresponding points, with the red and blue points referencing two seperate point clouds. Plot (b) shows the corresponding points after one iteration of the ICP algorithm, and you can see the distances between points are getting smaller, effectively minimizing the sum of squared errors. Plots (c) and (d) show two more iterations of the ICP algorithm, where with each iteration the distances get smaller until plot (e) finally shows the two point clouds converged into one.

VI. Home Run Tracking

Once these point clouds are created for MLB stadiums, one of the things it allows for are accurate home run distance estimates. Clay Nunnally, a baseball scientist for the MLB, ran a case study of how this process works [12]. Avisail Garcia, who at the time was an outfielder for the Tampa Bay Rays, hit a home run on July 20th, 2019, that hit a banner hanging from the ceiling of the Rays stadium. This banner stopped the ball in its flight path, so while it was clear that the ball was a home run, it was unknown how far the ball would have kept travelling. Thanks to a point cloud of the stadium, this distance was able to be estimated as soon as the ball hit the banner. The impact point of the ball on the banner was noted and identified in the LiDAR point cloud. Based on the point cloud, it is known how far from home plate the ball was when



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

(e) Iteration 5 (complete)

Figure 4. Iterations of the ICP algorithm [18]

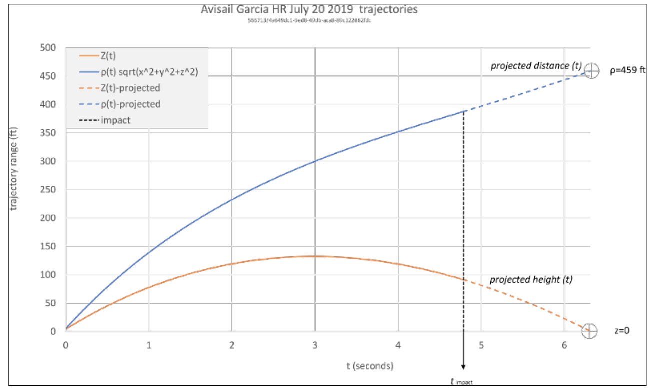


Figure 5. Avisail Garcia's Home Run [12]

it made impact, as well as the height it was at relative to the ground. In Figure 3 you can see these values, with the impact point being 389.15 feet from home plate and 73.77 feet off of the ground. Next, the flight time of the ball is calculated by going frame by frame in the broadcast video, noting when the ball made contact with the bat and when it made contact with the banner. Environmental factors must also be considered in estimating a ball's flight path, but in this case these factors are limited because the Rays stadium is inside in a dome. Nunnally states, "In general, air density and wind conditions help to model the ball's flight path more precisely" [12]. The last thing that is needed to build an estimation model is the high confidence measurements of exit velocity, launch angle, and spray direction of the batted ball that are all obtained using LiDAR and radar scans. Figure 5 shows the ball flight trajectory of Garcia's home run, along with the projected distances that Nunnally calculated during his tests. The black dotted line represents the impact point of the ball with the banner. Everything before the impact line is the ball's observed flight, and everything after the black dotted line is the projections that are made for the rest of the unknown ball flight, as if it never hit the banner. When the projected height reaches zero, the projected distance is noted and that is the final estimated distance for this home run. As seen in the figure, Garcia's home run is projected at 459 feet by this model, which is exactly the projected distance that Statcast announced after he hit it. This is just one case of using LiDAR point clouds to estimate home run distance, but whether the ball hits a banner, goes out of the stadium, or just falls unobstructed into a seat, the distance can be measured using this process.

VII. Future Trends

Looking towards the future, these new technologies will bring advancements to both players and the entertainment industry of Major League Baseball. Players are already starting to use new data gained from point clouds and the ability to track 18 data points on each player to their advantage. Since much of this data is new in the last few years, individual players are looking for ways to separate their game from others, and so far pitchers are finding the most benefits.

A. *Pitch Tunneling*

Pitchers are in direct control of the game with the ball in their hands. While hitters and fielders take a more reactionary approach, the pitcher starts every play and gets to decide what pitch to throw, where to throw it, and how hard. One aspect of new data that is getting a lot of attention is pitcher release points. The 12 Hawkeye cameras can monitor frame by frame exactly when a pitcher releases the ball, and map it to a point cloud. Pitchers are using this data to gain even further control by mastering the concept of pitch tunneling. There is no exact definition of pitch tunneling, but essentially pitchers want to release every pitch at the exact same point so no matter what type of pitch is thrown (curveball, fastball, etc.) it will look the same for as long as possible, confusing the hitter. An article written by Luke Smailes gives a good summary of the goal of pitch tunneling. "To effectively tunnel pitches, the goal is to minimize the distance of two pitches at the commit point while then maximizing the distance between the pitch's final coordinates" [17]. Figure 6 shows an image taken from baseball savant, a website that serves as Statcast's database [1], and shows the mapping of release points for pitcher Jacob

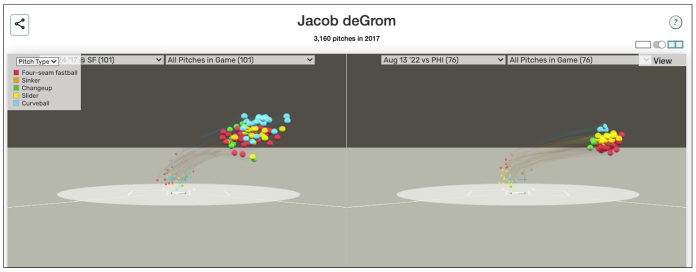


Figure 6. Pitch release points for Jacob deGrom in two games. The left chart is from 2017 and the right chart is from 2022. The improvement of deGrom's pitch tunneling can be seen in this figure [1].

deGrom in two separate games. The map on the left is a game from 2017 and the map on the right is a game from 2022. As you can see, in 2017 he released his pitches inconsistently from various locations, whereas in 2022 his release points are much more condensed thanks to the technology that he could learn from. As more pitchers see this data and learn to master the art of pitch tunneling in the future, the difficulty for hitters will only rise.

B. MLB FieldVision

Not only is this new data benefiting players, but it also opens many new doors in the entertainment industry. MLB FieldVision is the latest project being worked on and was released in 2022. MLB FieldVision is a visual tool used to model games in real time based on the 18 data points taken on each player 30 times per second [15]. These data points are mapped in a simulated stadium, which is created and built to scale thanks to point clouds. Figure 7 shows a live play in MLB FieldVision. It follows the game in real time, and models the players with skeletal figures based on the data point tracking. One of the biggest advantages to this software is that plays can be watched from any angle. On television, there are a limited amount of camera angles to watch a play, but with the 3D representations in FieldVision, users can fly all over the stadium and watch a play from wherever they please. Another future implication of FieldVision is the ability to watch games in virtual reality, or VR. Teams will be able to sell tickets in a virtual setting, placing your person virtually in the seat you buy. It would be great to be able to walk around the stadium freely in VR during a game, but knowing the MLB they will find a way to monetize everything as these features grow.

VIII. The Conclusion

Overall, computer vision has been used in baseball for many years and is only getting better. The increased use of cameras, along with LiDAR and radar, are improving accuracy and allowing point clouds of MLB stadiums to be created. Hawk-Eye innovations is leading the surge with this new technology, allowing the Statcast system to track ball trajectories to a never before seen accuracy. This is all thanks to the algorithms within, which include the ICP algorithm and kalman filtering. The amount of data that computer vision is bringing the sport is exciting and MLB franchises are ready to reap the benefits.

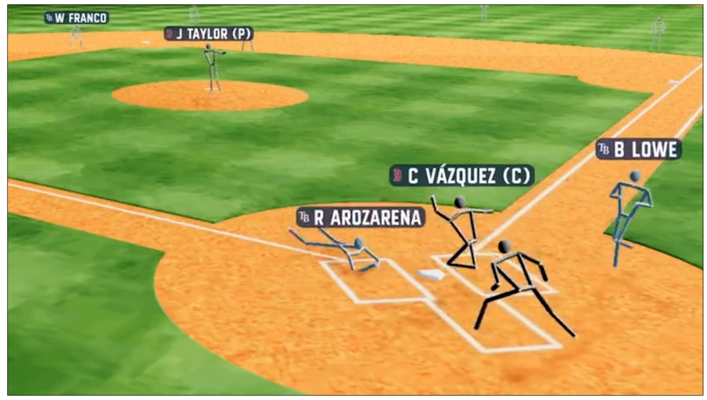


Figure 7. A play between the Tampa Bay Rays and Boston Red Sox as seen in MLB FieldVision [15]

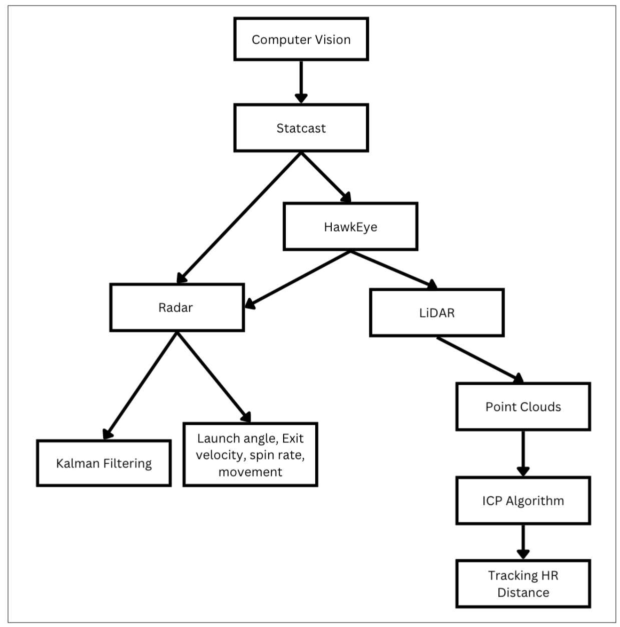


Figure 8. Mental Model: Computer vision is what created Statcast, and Statcast is helped immensely by the Hawk-Eye ball tracking system. This system uses both radar and LiDAR guns to track various player and ball movements. Kalman filtering organizes images, and when combined with radar readings, it helps obtain values such as launch angle, exit velocity, spin rate, and player movement. On the other side, LiDAR scans help replicate MLB stadiums through point clouds. The ICP algorithm combines these point clouds, resulting in models that can accurately track home run distances.

References

[1] Baseball savant: Trending mlb players, statcast and visualizations.

[2] Computerphile. Joining point cloud scans (icp) - computerphile, Dec. 2021.

[3] Quinten Dol. A deeper look at mlb's new statcast, Oct 2020.

[4] Esther Ezra, Micha Sharir, and Alon Efrat. On the icp algorithm. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, SCG '06, page 95–104, New York, NY, USA, 2006. Association for Computing Machinery.

[5] André Guéziec. Tracking pitches for broadcast television. *Computer*, 35(3):38–43, mar 2002.

[6] Henry Haugsten Hansen, Sayed Muchallil, Carsten Griwodz, Vetle Sillerud, and Fredrik Johanssen. Dense lidar point clouds from room-scale scans. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, page 88–98, New York, NY, USA, 2020. Association for Computing Machinery.

[7] Ben Jedlovec. Introducing statcast 2020: Hawk-eye and google cloud, Jul 2020.
[8] Joe Lemire. Soti reveal: 'good eye' will take on a whole new meaning in major league baseball, Mar 2020.
[9] Nikola Milosavljevic. *Rigid Body Transform Geometric and Kinematic Models of Bio Molecules*. Stanford University, 30, March 2004.
[10] Jon Paul Morosi. *Technological Updates for Season Revealed.* MLB.com, 21, July 2020.
[11] Neltronics. *Neltronics – How Does a Speed Camera or Radar Gun Work?*, 2012.
[12] Clay Nunnally. A lidar-validated trajectory for a long home run - medium, Oct 2019.
[13] Shinji Oomori, Takeshi Nishida, and Shuichi Kurogi. Point cloud matching using singular value decomposition. *Artificial Life and Robotics*, 21:149–154, 06 2016.
[14] Yan Pei, Swarnendu Biswas, Donald S. Fussell, and Keshav Pingali. An elementary introduction to kalman filtering. *Commun. ACM*, 62(11):122–133, oct 2019.
[15] SABRvideos. 2022 sabr analytics: Mlb fieldvision demo and deep dive, Apr. 2022.
[16] Artur Schütz, Olaf Poenicke, Patrick Schmett, Klaus Richter, and Franz Kellerstraß. Activity localization and tracking in order picking processes using lidar sensors: Lidar based activity tracking. In *2020 3rd International Conference on Sensors, Signal and Image Processing*, SSIP 2020, page 46–51, New York, NY, USA, 2020. Association for Computing Machinery.
[17] Luke Smailes. A closer look at pitch tunneling, Apr 2021.
[18] Cyrill Stachniss. *ICP*. jupyterNotebook.
[19] Cyrill Stachniss. Icp point cloud registration - part 1: Known data association svd (cyrill stachniss, 2021), Mar. 2021.
[20] YouTube. *Visually Explained: Kalman Filters.*, 28 Jan. 2021.

Appendix

The computer science courses I have taken here at CSBSJU have helped me in a lot of ways. One of the main focuses of the courses I have taken throughout my four years here has been programming, which was not needed much for the research I did in this project. Learning in a technical manner, creating visualization for data, problem solving, and learning about algorithms are the biggest factors that have helped me in this project.

Before coming to college at Saint John's, my writing experience consisted mostly of reflections, essays, and creative writing. Technical writing was a new aspect for me and was something that I got exposed to in my Bioinformatics and Machine Learning classes. In both of these classes, which I took junior year, I had to read plenty of technical papers and reports to gain knowledge for my final project. These papers introduced me to the principles of good technical writing. I have learned about the importance of organization, clarity, and conciseness in technical writing, and have learned how to apply these principles to my own writing.

Another aspect of computer science that my previous course work has taught me is about data visualization. In machine learning I did a project to predict the most accurate march madness tournament outcome. During this research I studied lots of graphs and charts that were built using machine learning. I also took two data visualization courses for my data analytics minor. These courses helped me understand the data gathered by Statcast, and helped me use the baseball savant website to sort data in ways that I needed, such as creating visuals for Jacob deGrom's pitch release points.

Finally, every course that I have taken in the computer science department has taught me about algorithms and problem solving. Algorithms run everything behind the scenes in computer science, and studying so many different algorithms made it easy for me to research and understand Kalman filtering and the ICP algorithm, which I reference in this study. Problem solving is also a basic principle of computer science and can be applied both in the field and in every day life. In this class we read the book "Algorithms to Live By: The Computer Science of Human Decisions" which discussed different algorithms and ways to problem solve logically in day to day life. Reading this book and the discussions we had about it helped me throughout the semester to most efficiently and accurately complete my research and assignments.