IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 4, 2019, accepted November 8, 2019, date of publication November 13, 2019, date of current version November 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953340

Precise 3D Baseball Pitching Trajectory Estimation Using Multiple Unsynchronized Cameras

JOONGSIK KIM, MOONSOO RA, HONGJUN LEE, JEYEON KIM, AND WHOI-YUL KIM
Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea

Corresponding author: Whoi-Yul Kim (wykim@hanyang.ac.kr)

This research project was supported by the Sports Promotion Fund of Seoul Olympic Sports Promotion Foundation from Ministry of Culture, Sports, and Tourism.

ABSTRACT We developed a method for the precise estimation of the 3D trajectory of a baseball by modeling the movement of the baseball and estimating the capture delay, using multiple unsynchronized cameras. To develop the proposed algorithm, we mimicked the real-world process of capturing a baseball in simulation space, and analyzed the capture process using a multiple unsynchronized camera system. We represented the movement of the baseball using a piece-wise spline model, and predicted the position of the baseball in the subframes in a manner which is robust to position error and change in direction of movement of the baseball. This method accurately predicts the baseball position over time by modeling the movement of the baseball in a real baseball game environment, and improves the accuracy of the reconstructed 3D baseball trajectories. We defined an objective function to estimate the capture delay, and estimate the optimal capture delay parameter using non-linear optimization method. In addition, we evaluated the performance of the proposed method in simulation space and in a real-world situation. The experimental results show that the proposed method can estimate a 3D baseball trajectory precisely using a multiple unsynchronized camera system and is robust to variations in capture delay, both in the simulation space and in real-world situations.

INDEX TERMS Stereo vision, 3D pitching trajectory, multiple unsynchronized cameras, camera calibration.

I. INTRODUCTION

In recent years a number of different baseball pitching analysis systems have been used to help viewers and players better understand the game. Many broadcasters and baseball clubs have adopted the use of trajectory analysis systems, and trajectory information output from these systems can be used as content for baseball broadcasts, VR baseball game data, and pitching training information. Major League Baseball (MLB) has used trajectory information for training professional baseball players and baseball referees [1]. A trajectory analysis system uses different types of image capture devices to detect a baseball. Initial systems simply highlighted the baseball's location and showed its position on the broadcast screen [2]. This type of system only shows discrete ball locations from a specific viewing angle, and cannot obtain other data such as spin, velocity or 3D location. Several studies into the acquisition of 3D information about a ball have been conducted to obtain information that can be used for training.

A 3D baseball pitching analysis system can be categorized as either radar-based or camera-based, depending upon the sensor type. One example of a radar-based system is TrackMan Baseball. TrackMan Baseball uses a military-grade Doppler radar and measures every object in the baseball stadium [3], including the 3D baseball position, speed, spin rate, angle, and baseball player. MLB has been using TrackMan Baseball because it can acquire data about players' movements and is compatible with the Statcast system [1]. Statcast is a tool for analyzing data about the performance of baseball players. However, the radar-based system is difficult to operate in real time because the radar sensor can be overwhelmed with information. For an amateur or senior baseball league the sensor is very expensive compared to camera-based systems [4].

Camera-based systems capture the baseball pitching area using single or multiple cameras installed in the baseball

The associate editor coordinating the review of this manuscript and approving it for publication was Mahammad Abdul Hannan.

stadium. Camera-based systems only detect the trajectory of the ball, and so can be operated in real time. This system only detects 2D baseball locations and the data can be used to reconstruct 3D information, so it is feasible for the system to operate in real time. Rapsodo is an example of a monocular camera-based system [5]. This system analyzes the pitching trajectory in front of a catcher using a camera installed behind the catcher. The system measures the spin rate and axis of a thrown baseball and estimates the entire pitching trajectory from partial data. Since Rapsodo only uses data collected near the catcher, it is difficult for it to estimate all of the pitching trajectory data and the location of the baseball in 3D [4]. This system cannot be used in professional baseball leagues because no devices can be installed on the field. To address these limitations and acquire information from cameras installed outside the baseball field, we developed a multiple-camera based system. We installed cameras near the stand or dugout, and captured images of the area between the catcher and the pitcher. Pitchf/x is a well-regarded multiple camera-based pitch analysis system that uses two cameras installed on the high first and high home plate positions [6]. To obtain 3D pitching trajectory information, this system detects ball positions in 2D and estimates the coefficients of the 3D trajectory using a physical pitching model calibrated for each camera. However, this system only shows the 3D trajectory shape, and cannot calculate the location of the ball in 3D in the field. This system also requires a long calibration process and it is not possible to change the cameras' positions after calibration.

A stereo-based pitch analysis system has been developed for the calculation of baseball locations in 3D [7], [8]. This stereo-based system calculates baseball locations in 3D using triangulation. The system has fewer restrictions related to the camera installation and the extrinsic parameter calibration process, because it can use the 2D baseball positions as input to the calibration process. In this paper, we describe our research into the implementation of a low-cost trajectory analysis system with easy installation and minimum space requirements. Our objective is to produce a system which is applicable to as many baseball leagues as possible, from amateur to professional. To achieve this objective, we investigated the acquisition of 3D baseball trajectories via triangulation with a multiple-camera system. However, triangulation with multiple cameras typically requires the synchronization of the timings of image captures between cameras.

Since unsynchronized cameras do not capture images at exactly the same time, we cannot be certain that the objects captured by each camera have the same position; therefore, a basic assumption underlying the process of triangulation is not met. An unsynchronized camera system cannot accurately calculate 3D baseball trajectories, even if the calibration of the cameras has been performed perfectly. Table 1 shows that the average error of the baseball location in 3D of an unsynchronized system increases depending upon the difference in image capture timing.

TABLE 1. 3D reconstruction error by capture timing difference.

Capture timing difference (frame)	-0.75	-0.5	-0.25	0	+0.25	+0.5	+0.75
3D error (cm)	5.07	3.4	1.70	0.53	1.75	3.43	5.12

As shown in Table 1, if the timing of multiple cameras is out by more than one frame, the 3D reconstruction error becomes greater than 6cm. Given that the diameter of a baseball is around 7cm, such an error is not acceptable.

To synchronize multiple cameras using a hardware-based method, an additional signal generator that sends synchronization triggers to the cameras is used. However, hardware-based methods cannot handle the capture delay that occurs after each camera module receives the captured signal. Algorithm-based synchronization methods estimate the temporal offsets between multiple unsynchronized cameras, correct for the temporal offsets, and calculate extrinsic parameters. To estimate the temporal offsets, these methods capture images of the moving object and calculate the temporal offset of the trajectory of the moving object [8]–[16]. The temporal offset is the capture delay between the unsynchronized cameras, and several methods for estimating temporal offset in frame units have previously been proposed [10], [11]. If the computed temporal offset is in units of subframes, the locations of the moving object that are not actually captured by the camera are needed. To calculate the location of a moving object at a subframe level, the movement of the object over time is modeled, and the object location is predicted using an object trajectory model. To model the trajectory of the moving object linear, cubic and quadratic models have been proposed [8], [12], [14]. In particular, optimization algorithms have been used to estimate the optimal temporal offset parameters [8], [12]. A method for calculating temporal offsets using information about fewer objects has also been proposed [15]. The periodic motion of moving objects in tennis or ping pong [13] and in free fall motion [16] has also been used to estimate temporal offsets. However, as shown in Figure 1, since a baseball is a small object having a high velocity, its shape has severe motion blurs in the image. Therefore, it is difficult to detect its center accurately, so the ball movement model that is robust to the position error and motion change of a baseball is needed.

Since previous linear and cubic modeling methods only consider the positions of objects in two consecutive frames, modeling accuracy is reduced if the direction of motion of the baseball changes, or if the estimated position of the baseball is incorrect. In addition, since baseball pitching is neither periodic nor a free falling motion, it is hard to apply the previous method for estimating 3D baseball trajectories. The work most closely related to our method involved modelling the movement of the baseball with a quadratic model using the positions of the baseball and estimating the temporal offset using line search optimization [8]. However, this study

FIGURE 1. Baseball image captured during an actual game. The image resolution is 2048 × 512 and the baseball radius is about 3 pixels.

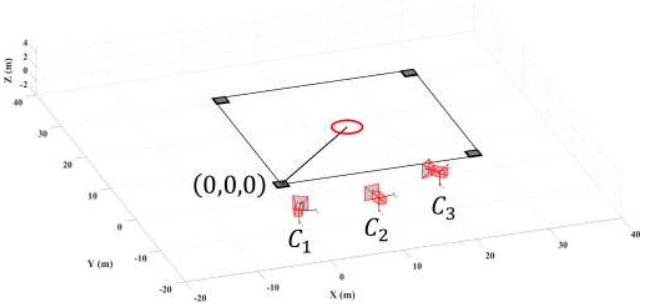

FIGURE 2. Results of the algorithm applied in a simulated baseball stadium with a multiple-camera configuration. The home plate is the origin in the simulation coordinates.

did not take into account the problem that uniform acceleration movement in a 3D space is not fully represented by a quadratic model in the image plane, due to the transformation of perspective. In addition, since linear optimization was used, the accuracy of the estimation of the temporal offset was limited.

Here we describe a 3D baseball trajectory estimation method using multiple unsynchronized cameras. The proposed method makes the following contributions:

(1) It analyzes a multiple unsynchronized camera system in a simulated space that mimics a real-world situation, describes the proposed algorithm, and verifies the method's feasibility in a simulation space.
(2) It models a baseball's trajectory using piece-wise spline function, an approach that is robust to error in the estimation of the center and changes in the direction of movement of the baseball. An objective function for optimizing capture delay and extrinsic parameters, using a non-linear optimization algorithm is defined and analyzed.
(3) The proposed method's performance is evaluated using both simulation and a real-world implementation, and the experimental results show that the proposed method can estimate the 3D baseball trajectory robustly with various capture delay parameters.

The remainder of the paper is organized as follows. Section II describes the simulation configuration of the proposed method. Section III describes the proposed calibration method based on an unsynchronized camera system. Section IV presents the experimental results and Section V presents our conclusions.

II. ANALYSIS OF THE MULTIPLE UNSYNCHRONIZED CAMERA SYSTEM IN SIMULATION

This section describes the implementation of the multiple unsynchronized-camera system in simulation and the capture process of the unsynchronized system.

A. CONFIGURATION OF THE SIMULATION ENVIRONMENT

To construct a simulation environment that resembles the real world, we first define the home plate as the origin in the 3D simulation space. The remaining elements of the stadium such as the first, second, and third bases are then defined according to the standard baseball field diagram in the official baseball rules [17]. Since no capture device is allowed to be installed on the baseball field, the cameras are placed near either the dugout or stands. Figure 1 shows an example of the simulation space for the multiple-camera system set-up results. As shown in Figure 2, the 3D location of the i^{th} camera in the simulation space is defined as $C_i (i = 1, 2, \ldots)$ and we assume that every camera is a simple pinhole camera model [18].

B. IMPLEMENTATION OF THE CAPTURE PROCESS OF A MULTIPLE UNSYNCHRONIZED-CAMERA SYSTEM IN A SIMULATION SPACE

Here we define the unsynchronized system and describe the implementation of the unsynchronized system in a simulation space. We then obtain the 2D baseball points captured by cameras using the projection matrix of each unsynchronized camera. In this work, we deem a system to be unsynchronized when there are non-zero capture delays between cameras. Since a capture delay is defined as the difference in capture timing between two cameras, we select C_1 as a reference camera to define capture delays for all other cameras. In other words, in the unsynchronized system, the i^{th} $(i = 2, 3, \ldots)$ camera has a non-zero capture delay relative to C_1 and these capture delay parameters are denoted by τ_i (ms). The capture moment for the baseball in an unsynchronized system is illustrated in Figure 3.

As shown in Figure 3(a), we assume that C_2 and C_3 have a negative capture delay τ_2 $(\tau_2 < 0)$ and a positive capture delay τ_3 $(\tau_3 > 0)$ relative to the capture timing of the reference camera, and these three unsynchronized cameras capture the location of the baseball at time t_5. Compared to the location captured by C_1 in Figure 3(b), C_2 captures a baseball located at $t_5+\tau_2$ on the same trajectory. Similarly, C_3 captures a baseball located at $t_5 + \tau_3$ on the same trajectory. Even though unsynchronized cameras attempt to capture the baseball at the same time, captured baseballs have different 3D locations, although they all lie on the same trajectory of the baseball captured by a reference camera. Therefore,

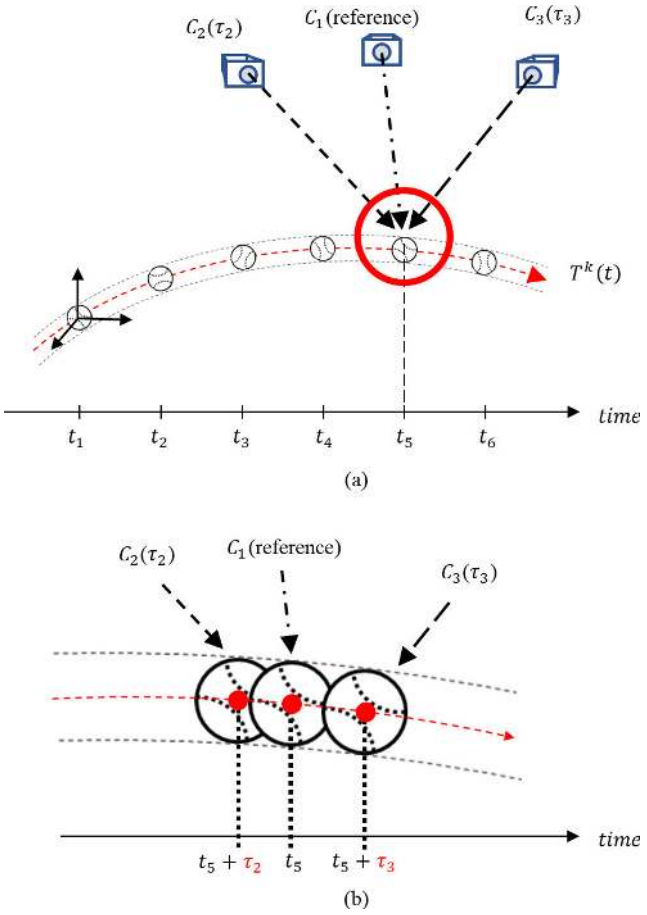

FIGURE 3. Illustration of the moment of capture in a multiple unsynchronized-camera system.

if we know the 3D trajectory of the baseball captured by the reference camera, we can calculate the 3D baseball location captured by other unsynchronized cameras by considering capture delays in the trajectory.

We can mimic real-world baseball movements in a simulation space; we assume that a thrown baseball has uniform acceleration, hence the 3D baseball movement can be modeled as quadratic equations with respect to time t [19]. Let $\boldsymbol{T}_1^k(t)$ denote the k^{th} 3D baseball trajectory function of C_1 at time t and define $\boldsymbol{T}_1^k(t)$ as

$$\begin{aligned}\boldsymbol{T}_1^k(t) &= \left(x^k(t), y^k(t), z^k(t)\right) \\ &= \begin{pmatrix} x_0^k + v_x^k t + \frac{1}{2}a_x^k t^2, \\ y_0^k + v_y^k t + \frac{1}{2}a_y^k t^2, \\ z_0^k + v_z^k t + \frac{1}{2}a_z^k t^2 \end{pmatrix}\end{aligned} \tag{1}$$

where $\left(x_0^k, y_0^k, z_0^k\right)$ is an initial release point of the reference baseball trajectory and $v_x^k, a_x^k, v_y^k, a_y^k, v_z^k$, and a_z^k are coefficients of the reference baseball trajectory. Equation (1) have six degrees of freedom, so the coefficients can be calculated by fitting more than two known 3D points. In this paper,

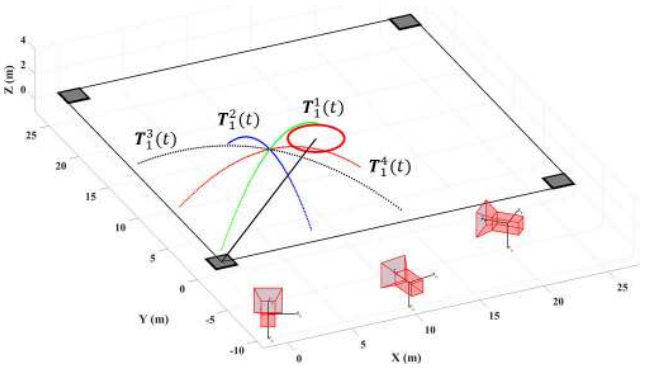

FIGURE 4. The four different 3D baseball trajectories of the reference camera (C_1) in the simulation space.

we selected three points (release, highest, and end points) for $\boldsymbol{T}_1^k(t)$ in the simulation space and generated 3D points, $\mathbf{X}_1^k$ with time interval $1/fps$ s from $t = 0$ to $t = t_{end}$. The t_{end} represents the flight time for the ball until it reaches the strike zone and we set the value by considering the average fastball speed in the MLB. In this paper, fps is $179 frame/s$, t_{end} is 0.6s, and $\mathbf{X}_1^k$ from four different trajectories $\boldsymbol{T}_1^k(t)$ $(k = 1, 2, 3, 4)$ are used in the simulation space as shown in Figure 4.

The k^{th} 3D baseball trajectories captured by the i^{th} $(i = 2, 3, \ldots)$unsynchronized camera that has τ_i can be represented as

$$\boldsymbol{T}_i^k(t) = \boldsymbol{T}_1^k(t - \tau_i), \quad i = 2, 3, .. \tag{2}$$

The 3D baseball points generated from $\boldsymbol{T}_i^k(t)$ are denoted by $\mathbf{X}_i^k$ $(i = 2, 3, \ldots)$, and $\mathbf{X}_i^k$ is calculated in the same manner as $\mathbf{X}_1^k$. After generating the 3D baseball points, we calculate the 2D baseball points projected onto the image plane of each camera. Each camera captures different 3D baseball points $\mathbf{X}_i^k$, so the projected 2D baseball points of the i^{th} camera are calculated using the corresponding 3D baseball points $\mathbf{X}_i^k$ that the camera captures. Let $\mathbf{x}_i^k$ denote the projected 2D baseball points on the image plane of the i^{th} camera from $\mathbf{X}_i^k$. The homogenous coordinate $\tilde{x}_i^k$ of the projected 2D points $\mathbf{x}_i^k$ can be calculated as

$$\tilde{\mathbf{x}}_i^k = \mathbf{K}_i\left[\mathbf{R}_i|\mathbf{t}_i\right]\tilde{\mathbf{X}}_i^k \tag{3}$$

where $\mathbf{K}_i, \mathbf{R}_i$ and $\mathbf{t}_i$ are the intrinsic parameter, rotation matrix, and translation matrix of the ith camera, respectively. In this paper, the parameters in (3) are set empirically to capture the entire pitching area and we assume that each camera has an image resolution of 1280×720 without lens distortion. Figure 5(a) shows an example of the 2D projected points from the three unsynchronized cameras with non-zero capture delays ($\tau_2 < 0$ and $\tau_3 > 0$) and with the same extrinsic parameters ($\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{R}_3, \mathbf{t}_1 = \mathbf{t}_2 = \mathbf{t}_3$).

As shown in Figure 5(b), the 2D points projected on the image plane of the camera that have negative capture delay are located behind the reference position. In contrast, as shown in Figure 5(c), the 2D points projected on the

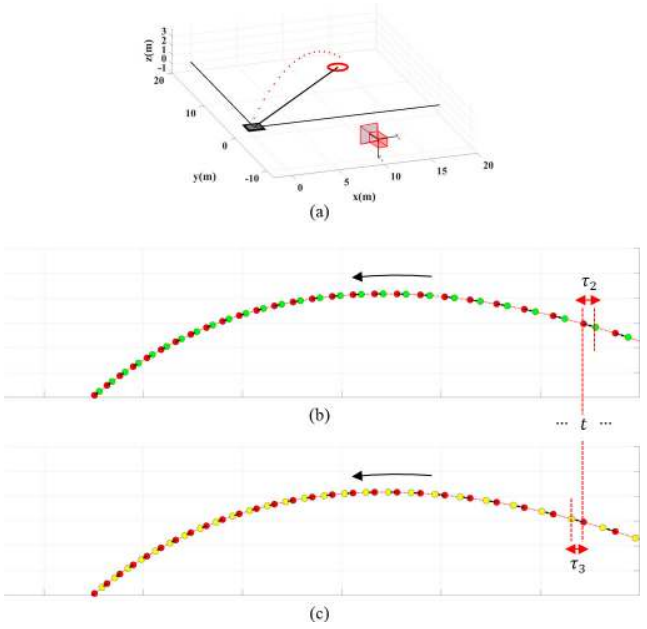

FIGURE 5. Differences in the projected 2D baseball points from the camera with different capture times for the same [R|t]. (a) 3D baseball projection environment in simulation space, (b)-(c) 2D projected point from reference camera (red points), negative capture.

image plane of a camera that has a positive capture delay are located ahead of the reference position. Through this process, we can mimic the real-world process of capturing the location of a baseball using multiple unsynchronized cameras in a simulation space and get 2D baseball points projected onto each camera image plane.

C. ESTIMATING MULTIPLE SOURCES OF NOISE AND THEIR PARAMETERS

In a real-world environment, let us assume that there are three noise sources that may produce errors while estimating 3D baseball positions: capture delay noise, quantization noise, and detection noise. Figure 6 shows an example that considers three noise sources.

The capture delay noise N_c represents a random capture timing inconsistency added to τ_i. Capture delay noise occurs after a camera receives a capture request. When a camera receives a capture request operation, it starts to store image data in the image-streaming buffer. When the buffer is full, the camera acquires image data from the image-streaming buffer. In this process, due to CPU load or camera hardware problems, the image data acquisition time may not be constant; this non-uniform acquisition time is represented as N_c. In this paper, we assume that N_c follows a Gaussian distribution @mathcalN $(0, \sigma_c)$. The quantization noise N_Q occurs while the image sensor digitizes the scene. Since we cannot avoid this type of noise while using digital devices, the lower boundaries of the 3D reconstruction errors are determined by N_Q.

To consider N_Q, we rounded off the projected baseball positions in a process denoted as $Q\left[\mathbf{x}_i^k\right]$. The detection noise N_D occurred when the algorithm detected the 2D center

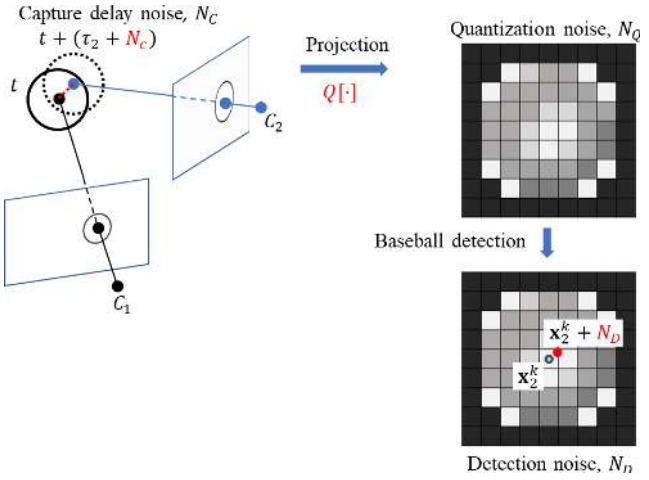

FIGURE 6. The process of adding three noise sources that can occur when cameras capture and detect a baseball in the real world.

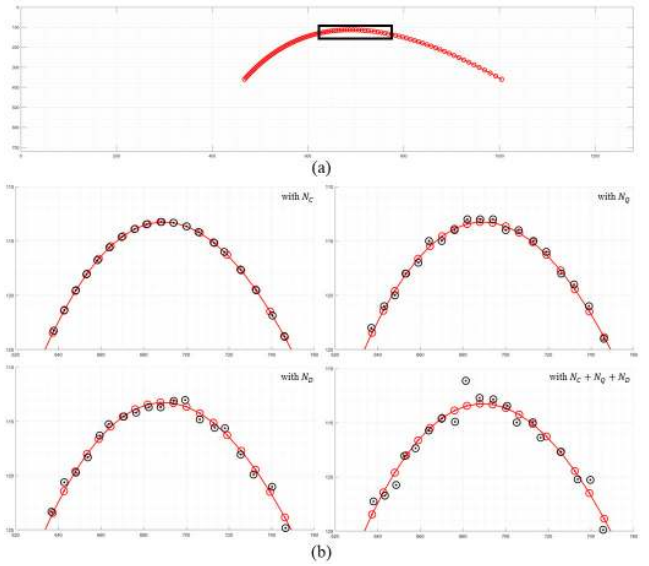

FIGURE 7. The reprojected 2D baseball points under different sources of noise. (a) Entire baseball trajectory points locations and highlighted region (back box), (b) Comparison with the ideal baseball locations (red circle) and locations with added noise (black circle).

position of the baseball in the image. In the real world, it is difficult to accurately find the center position of a baseball because of shadows, image blurring, and the occlusion of the baseball, as shown in Figure 1. Since the size of a baseball is very small compared to the captured image, it can be expected that differences in the center error in the noise model will not be large. Therefore, in this paper, we assume that N_D can be approximated by a Gaussian normal distribution @mathcalN $(0, \sigma_D)$. The N_D added result is represented by $\left(\mathbf{x}_i^k + N_D\right)$. Figure 7 shows the 2D baseball positions with various noise sources.

These 2D baseball points with noise provide the information that can be obtained using a multiple unsynchronized-camera system in a real-world situation. Therefore, in the simulation space, we compute the extrinsic

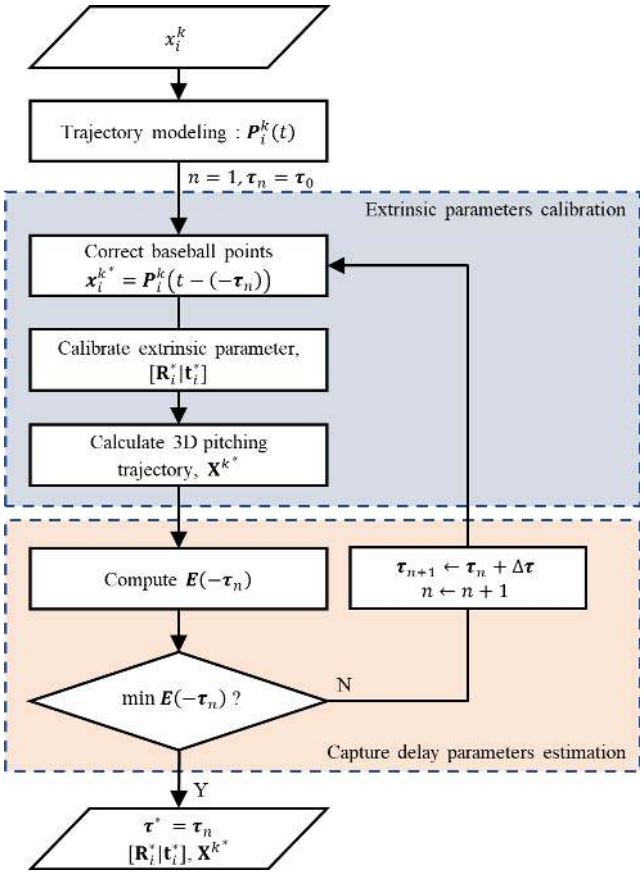

FIGURE 8. Flowchart of the proposed method.

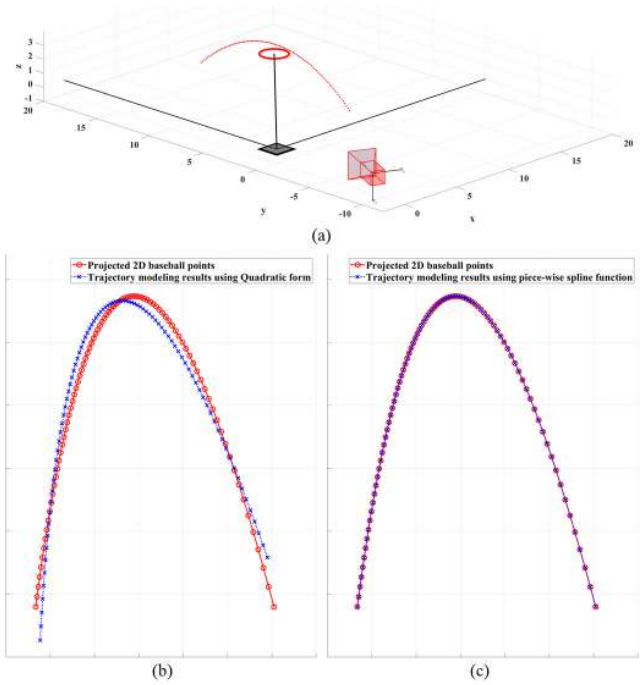

FIGURE 9. Results of trajectory modeling (a) Example of 3D baseball trajectory and camera location in the simulation environment, (b)-(c) Trajectory modeling results based on the modeling method. The quadratic form cannot model the perspective transformed trajectory in the pinhole camera model compared with the proposed piece-wise spline fitting.

parameters of unsynchronized cameras and reconstruct the 3D trajectory information using only $\mathbf{x}_i^k$ as correspondences between cameras.

III. MULTIPLE UNSYNCHRONIZED CAMERA CALIBRATION OPTIMIZATION

In this section we describe the calibration of a multiple unsynchronized-camera system using 2D baseball points. Figure 8 shows the process of the proposed method for accurately estimate the trajectory in multiple unsynchronized camera system. The algorithm is divided into three parts: a trajectory modeling process; extrinsic parameter calibration; and capture delay parameter estimation. In the trajectory modeling process, the trajectory model $\boldsymbol{P}_i^k(t)$ is calculated as a piece-wise spline function using $\mathbf{x}_i^k$ from the multiple unsynchronized cameras. The $\boldsymbol{P}_i^k(t)$ is are used to correct the baseball position, taking into account the estimated capture delay, $\boldsymbol{\tau}$ in the following process. In the extrinsic parameter calibration process, using the estimated $\boldsymbol{\tau}_n$ and $\boldsymbol{P}_i^k(t)$, we calculate the corrected baseball position, $\mathbf{x}_{ik*}$. The extrinsic parameters are calculated using $\mathbf{x}_{ik*}$ and the 3D information, $\mathbf{X}^{k*}$ about the trajectory is calculated. In the capture delay parameter estimation process, the accuracy of $\mathbf{X}^{k*}$ is calculated. If the reconstruction error of the 3D trajectory is not minimum, the estimated $\boldsymbol{\tau}$ is updated, and the extrinsic parameter calibration process is repeated. The optimization algorithm is used to update $\boldsymbol{\tau}$. Unfortunately, these two problems lack closed-form solutions, so we need to carry out two algorithms iteratively to solve the problems.

A. CALIBRATION OF THE EXTRINSIC PARAMETERS USING 2D BASEBALL POINTS CONSIDERING CAPTURE DELAY PARAMETERS

We calibrated the extrinsic parameters using corrected baseball correspondences considering the estimated capture delay parameters. To correct the 2D baseball points, we needed to obtain two pieces of information: the 2D pitching trajectory model and the capture delay parameters. The trajectory model is necessary to estimate the baseball's position corrected by the capture delay time from the given positions. Since the trajectory model is fitted by a given $\mathbf{x}_i^k$, the modeling algorithm is robust to noise sources and trajectory shape. For example, Miyata et al. modeled the 2D baseball points using quadratic equations [8]. However, the perspective transform of the pinhole camera model makes it difficult to model all 2D trajectories as quadratic equations. The proposed method solves this problem by modeling the trajectories using a piece-wise spline function [26], [27], because the entire form of a 2D trajectory is difficult to model using a single quadratic form. Figure 9 shows the trajectory modeling results. As shown in Figure 9, the quadratic form cannot represent the projected 2D baseball trajectory but the piece-wise spline function can model the way in which the 2D baseball trajectory follows

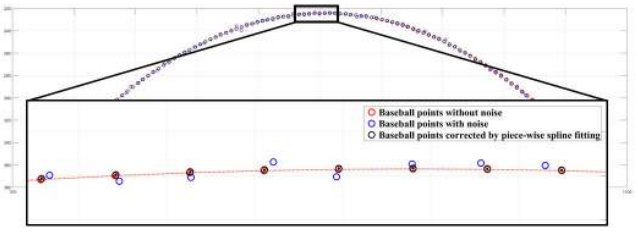

FIGURE 10. The baseball points with noise-source correction results applied using the proposed trajectory-modeling method.

the original trajectory. Since the proposed system captures the baseball trajectory at over 150 fps and there are many baseball points detected on the trajectory, the fitting results are almost the same as the original 2D trajectory. The average distance error between the baseball position and the fitted data is 6.0179 pixels in the quadratic model and 0.0304 pixels in the piece-wise spline model. The average distance error between the baseball position and the fitted data is 6.0179 pixels in the quadratic model and 0.0304 pixels in the piece-wise spline model.

In addition, the trajectory model has to be robust to noise sources; Figure 10 shows the trajectory modeling results of the 2D baseball points with noise sources. In Figure 10, the proposed method can model the 2D baseball points in a manner which is robust to noise sources and trajectory fitting results and produces results which are almost the same as the original trajectory. After modeling the 2D projected baseball trajectory, the 2D projected baseball points are converted to 2D points that incorporate the capture delay parameters. In this process, if the trajectory is divided into too many pieces, the local fitting error in the trajectory is reduced, but global motion of trajectory may not be considered. Therefore, in this work, we used empirically determined spline pieces $=$ 8 and equation order $=$ 3 as the parameters of the piece-wise spline function.

If the 2D baseball points captured by the camera have $\boldsymbol{\tau}$, it is possible that the 2D baseball captured by synchronized cameras is calculated by inverse delay at time $\boldsymbol{\tau}$. Let the converted 2D baseball points and trajectory model be denoted by $\mathbf{x}_i^{k*}$ and $\boldsymbol{P}_i^k(t)$. In the n^{th} iteration step, $\mathbf{x}_i^{k*}$ that considers $\boldsymbol{\tau}_n$ can be calculated as

$$\mathbf{x}_i^{k*} = \boldsymbol{P}_i^k(t - (-\boldsymbol{\tau}_n)) \tag{4}$$

To use $\mathbf{x}_i^{k*}$ as the correspondence between cameras, we must calibrate the extrinsic parameters. The fundamental matrix $\boldsymbol{F}_{ij}$ between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ is calculated using a five-point algorithm [20] and the MLESAC algorithm is used to reject outliers [21]. Afterwards, the essential matrix $\boldsymbol{E}_{ij}$ is computed from $\boldsymbol{K}_i$, $\boldsymbol{K}_j$, and $\boldsymbol{F}_{ij}$, and the rotation and translation matrixes are calculated by the decomposition of $\boldsymbol{E}_{ij}$. These new rotation and translation matrixes are denoted by $[\boldsymbol{R}_i^*|\boldsymbol{t}_i^*]$ And the 3D baseball trajectory $\mathbf{X}^{k*}$ is calculated from these extrinsic parameters by using the DLT algorithm [18]. Through this process, if the capture delay parameters are provided in the n^{th} iteration step, we can convert the 2D baseball points by considering the capture delay parameters, calibrating extrinsic parameters, and calculating the 3D baseball trajectory.

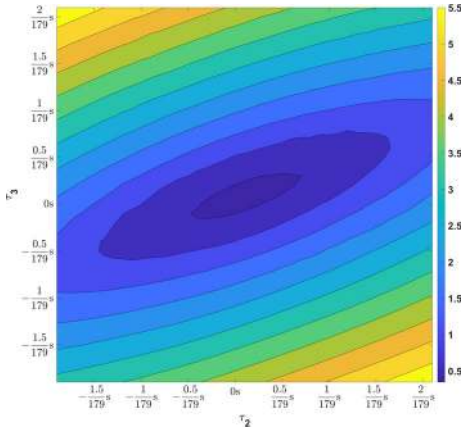

FIGURE 11. Objective function space of the proposed cost function.

B. ESTIMATION OF THE OPTIMAL CAPTURE DELAY PARAMETERS

To obtain accurate capture delay parameters, we need to define a measurement that can determine whether the estimated capture delay parameters are correct. As mentioned before, the extrinsic calibration accuracy is high when the 2D baseball points are synchronized. Therefore, if the capture delay parameter can be estimated precisely, the calibration accuracy increases because the synchronized 2D baseball points can be calculated when the capture delay parameters are correct. The calibration accuracy can be computed using the error of the Euclidean distance between $\mathbf{X}^k$ and $\mathbf{X}^{k*}$. Using this constraint, when the n^{th} capture delay parameters are given, we define the cost function $E(-\boldsymbol{\tau}_n)$ considering $\boldsymbol{\tau}_n$ as

$$E(-\boldsymbol{\tau}_n) = \frac{1}{K}\sum_{k=1}^{K} Dist\left(\mathbf{X}^k, \mathbf{X}^{k*}\right) \tag{5}$$

where $Dist(\mathbf{X}, \boldsymbol{Y})$ is the Euclidean distance between two 3D points. As shown in equation (5), the higher the calibration accuracy, the closer $E(-\boldsymbol{\tau}_n)$ is to 0 and this process can be represented by minimizing problems denoted as

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}_n} E(-\boldsymbol{\tau}_n) \tag{6}$$

where the $\boldsymbol{\tau}^*$ is the optimal capture delay parameters. In the n^{th} iteration step, if $E(-\boldsymbol{\tau}_n)$ is not the minimum value, the iteration step is increased and $\boldsymbol{\tau}_n$ is updated. However, if the $E(-\boldsymbol{\tau}_n)$ has its minimum value then the algorithm stops and the optimal $\boldsymbol{\tau}^* = \boldsymbol{\tau}_n$ value and extrinsic calibration results are obtained. The proposed method adapts the optimization algorithm to update $\boldsymbol{\tau}_n$ properly. To select the optimization algorithm, we verify the objective function space of $E(\boldsymbol{\tau}_n)$. Figure 10 is the objective function space of $E(-\boldsymbol{\tau}_n)$, $\boldsymbol{\tau}_n = \{\boldsymbol{\tau}_2, \boldsymbol{\tau}_3\}$.

As shown in Figure 11, the proposed objective function is a nonlinear convex function. Since the proposed object

TABLE 2. The list of test parameters in the simulation space.

System setting	Noise sources	**N1** (without noise) **N2** (N_c) **N3** (N_Q) **N4** ($N_D(\sigma_D = 0.5)$) **N5** ($N_c + N_Q + N_D(\sigma_D = 0.5)$) **N6** ($N_D(\sigma_D = 1.0)$) **N7** ($N_c + N_Q + N_D(\sigma_D = 1.0)$)
Algorithm	Trajectory fitting model	**F1** (Quadratic equation) **F2** (Piece-wise spline function)
	Optimization method	**O1** (Quasi-Newton method) **O2** (Nelder–Mead simplex method)

function is not differentiable, this paper adopts nonlinear convex optimization algorithms that does not require differential algorithm such as the Quasi-Newton method [22] and Nelder–Mead simplex algorithm [23] to estimate $\boldsymbol{\tau}^*$. In this work, we used the optimization algorithm from the MATLAB®optimization toolbox.

IV. EXPERIMENTAL RESULTS

The proposed method's performance for calibrating a multiple unsynchronized-camera system is conducted in a simulation space and the real world. The experiments were evaluated with two measures: a capture delay parameter estimation error and a 3D reconstruction error. Since the main idea of the proposed method is to estimate the capture delay and restore the baseball trajectory before the capture delay can occur, we measure the capture delay estimation accuracy. In addition, the 3D trajectory reconstruction accuracy of an unsynchronized system can be measured by the extrinsic calibration accuracy. The proposed method was run on a PC with AMD Ryzen 7 2700X 3.70 GHz CPU and 64 GB of RAM.

A. CALIBRATION PERFORMANCE EVALUATION IN A SIMULATION SPACE

In a simulation space, we set three cameras and manually changed the τ_2 and τ_3 from -2 frames to $+2$ frames with a 0.5 frame interval. After setting the capture system, we measured the capture delay estimation error and extrinsic calibration error in various noise sources, trajectory fitting models, and optimization methods. The capture delay estimation error can be calculated by the difference between $\boldsymbol{\tau}$ and $\boldsymbol{\tau}^*$ and the 3D reconstruction error can be calculated by averaging the 3D distance between $\mathbf{X}^k$ and $\mathbf{X}^{k*}$. Table 2 shows the various parameters used in this experiment.

As shown in Table 2, we had seven presets for the noise source combinations and four extrinsic calibration algorithms, depending on the combination of trajectory fitting model and optimization method. For example, algorithm [**F2O2**] means that 2D baseball points are modeled as a piece-wise spline and extrinsic parameters are optimized by the Nelder–Mead simplex method. The proposed method is represented by **F2O1** and **F2O2**. For each selected noise preset and calibration algorithm, the error matrix is calculated as shown in Figure 12.

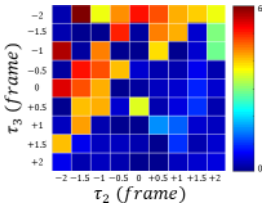

FIGURE 12. Example of an error matrix for performance evaluation.

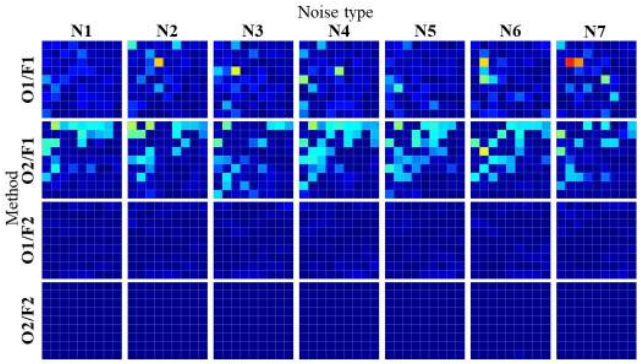

FIGURE 13. Performance evaluation results of estimated τ_2.

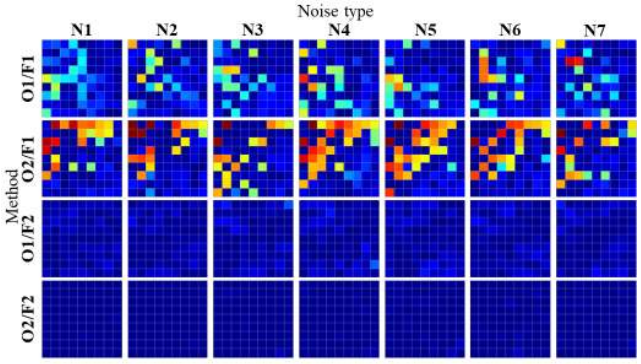

FIGURE 14. Performance evaluation results of estimated τ_3.

As shown in Figure 12, the error matrix has dimensions 9×9, and the rows and columns of the matrix represent different τ_2 and τ_3 values, which are selected manually before the synthetic 3D baseball trajectories are generated in the simulation space ($\tau_2, \tau_3 \in \{-2, -1.5, -1, -0.5, 0, +0.5, +1, +1.5, +2\}$). The m^{th} element of the n^{th} row represents the estimation error of τ_2, τ_3 or the 3D reconstruction error when the capture delay was set to $m^{\text{th}}\tau_2$ and n^{th} τ_3, respectively. The capture delay estimation error of τ_i was calculated using the absolute difference between τ_i and τ_i^*. The 3D reconstruction error was calculated using $E(-\boldsymbol{\tau}^*)$. The closer the color of the matrix element is to red, the larger the error. Figures 13–15 show the error matrices of the τ_2 and τ_3 values and the 3D reconstruction error, respectively. The horizontal and vertical axes of each figure represent the noise presets and calibration algorithms used for the test.

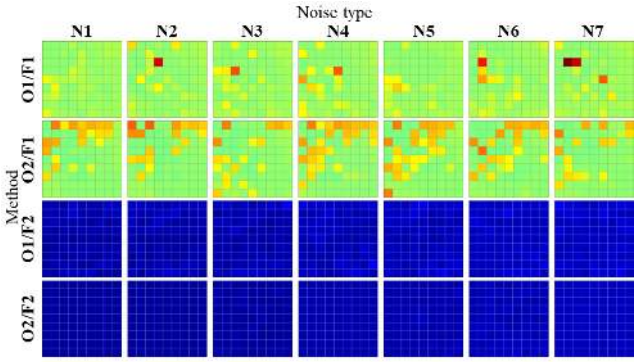

FIGURE 15. Performance evaluation results of the 3D reconstruction.

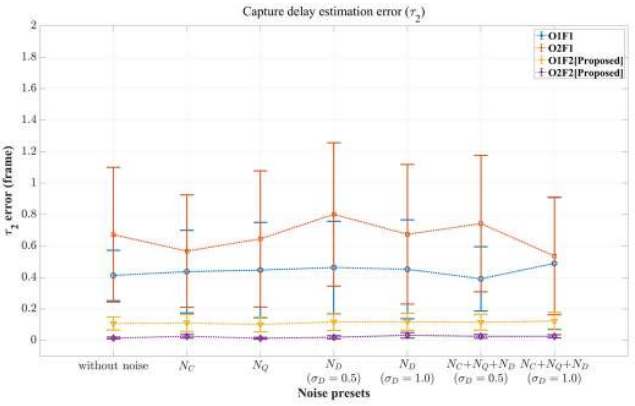

FIGURE 16. Average performance evaluation results of estimated τ_2.

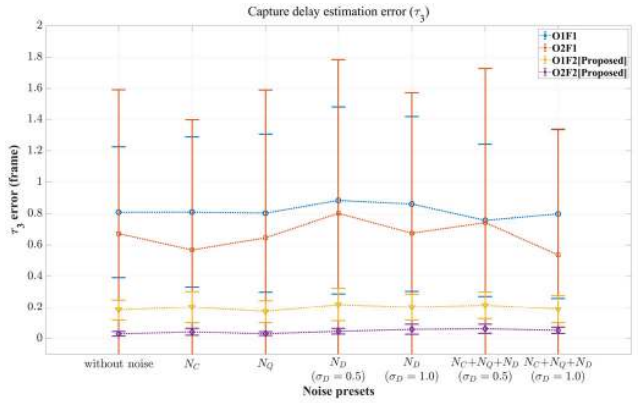

FIGURE 17. Average performance evaluation results of estimated τ_3.

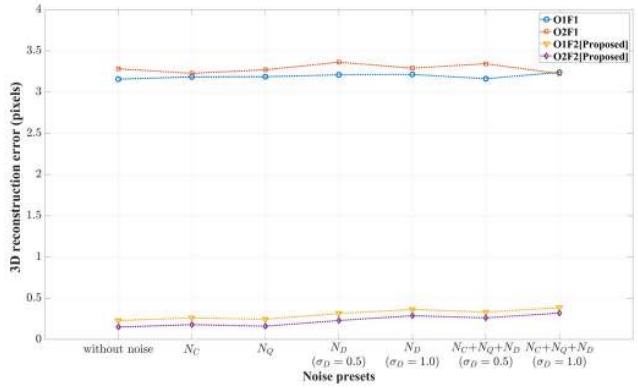

FIGURE 18. Average performance evaluation results of the 3D reconstruction.

As shown in Figures 13 and 14, the capture delay estimation error is close to zero for all noise presets when the 2D baseball trajectory is modeled by **F2**. The optimized delay parameter error increases when the 2D baseball trajectory is modeled using **F1** because the 2D baseball location has not been precisely corrected to its original location. This is because **F1** cannot model the 2D baseball trajectory accurately. Figure 15 shows the 2D baseball location restoration results compared to **F1** and **F2** when all noise sources are added to the projected 2D baseball locations. Figures 16–18 shows the average error of the estimated capture delay parameter and the 3D reconstruction at each test. The test results are represented using the mean and standard deviation of all estimated error values.

As shown in Figures 16 and 17, the τ_2 and τ_3 optimization results of the proposed method have the best performance with a low standard deviation. This indicates that the proposed method has reliably robust estimations for various capture delay changes and noise sources. When $N_D$occurs, the error is high because the 2D baseball location changes the most. However, the proposed method can robustly and accurately estimate the capture delay parameters even with the addition of various noise sources, because the proposed baseball trajectory fitting model reliably restores the 2D baseball location that contains noise sources to its original location better than the previous model. As can be seen in Figure 18, the proposed method also has the best performance in 3D reconstruction. It is noteworthy that when using the previous trajectory fitting method, even if the optimal capture delay time is estimated, the standard 3D reconstruction accuracy is low compared to the proposed method. When 2D trajectories are modeled by a simple quadratic equation, it is difficult to accurately reconstruct the 3D baseball trajectory information even if the capture delays are estimated using the optimization process. Figure 18 shows that the proposed 2D trajectory modeling method provides good performance regardless of the optimization method.

B. REAL-WORLD CALIBRATION PERFORMANCE EVALUATION

We verified the performance of the proposed method by testing the algorithm at a real baseball field with two trajectory-fitting models, as shown in Table 2. We captured real baseball pitches using two high-speed cameras at a baseball stadium in Seoul. We installed the cameras below the first dugout roof and set the camera lens to capture the entire pitching area. We used a Toshiba Telicam BU405M camera with an AZURE-1040Z3M zoom lens, and captured the pitching area at 175 fps with 2048×1024 px resolution.

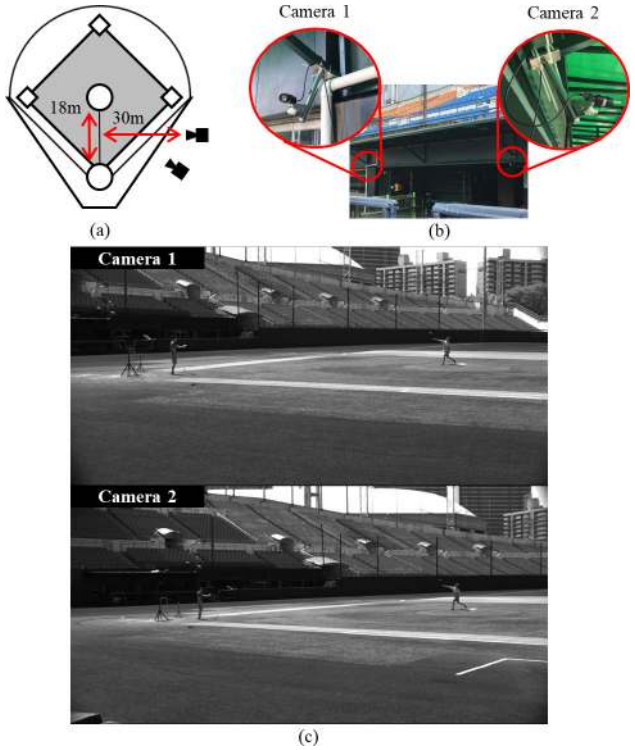

FIGURE 19. Real-world test system environment (a) Baseball field and camera positions, (b) High-speed camera installation behind the dug-out, (c) Captured image examples.

This camera supports automatic synchronization using a bus synchronization technique, and the user can specify the capture delay between cameras. We implemented the capture program using Toshiba TeliCamAPI. This experiment also was run on a PC with AMD Ryzen 7 2700X 3.70 GHz CPU and 64 GB of RAM. The intrinsic parameters of the cameras were calculated using the method of Zhang et al. [24]. Figure 19 shows the camera installation environment and the corresponding captured images.

To evaluate the performance of the proposed method at various capture delay times, we set four capture delay presets $\left(\frac{1}{3}\,frame, \frac{1}{2}\,frame, \frac{2}{3}\,frame, 1\,frame\right)$. In addition, we obtained six different pitch trajectories in the pitching area for use in extrinsic calibration and trajectory reconstruction. Figure 20 shows the pitching trajectories that were used for calibration. The center of the baseball was detected using Kim's algorithm [25]. This baseball detection algorithm detects the peak positions as the center of the baseball from the difference image without using the shape of the baseball. In a continuous image captured by a high-speed camera, it can be assumed that the background does not change, so a moving object region can be obtained from the difference image between two continuous images. In order to detect a baseball in a manner which is robust to detection error and occlusion, we used Hong's baseball tracking algorithm [28]. Figure 21 represents the baseball center detection results. The same trajectories between two cameras are represented by the

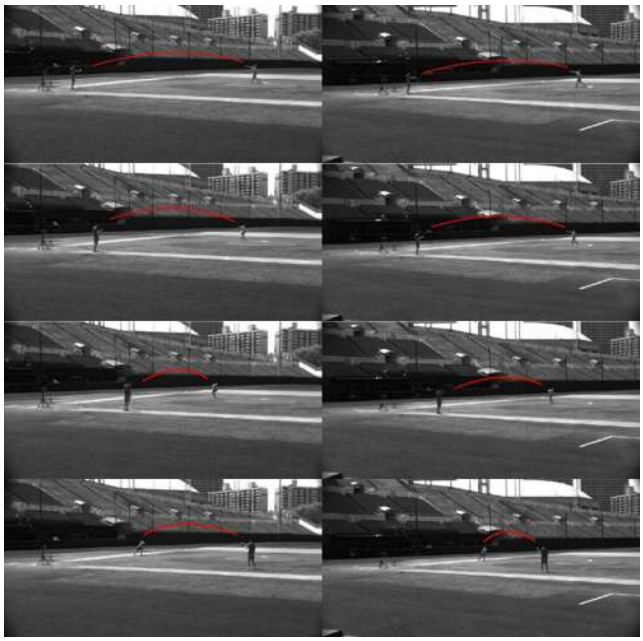

FIGURE 20. Pitching trajectories for calibration. (Left column: Camera 1, Right column: Camera 2).

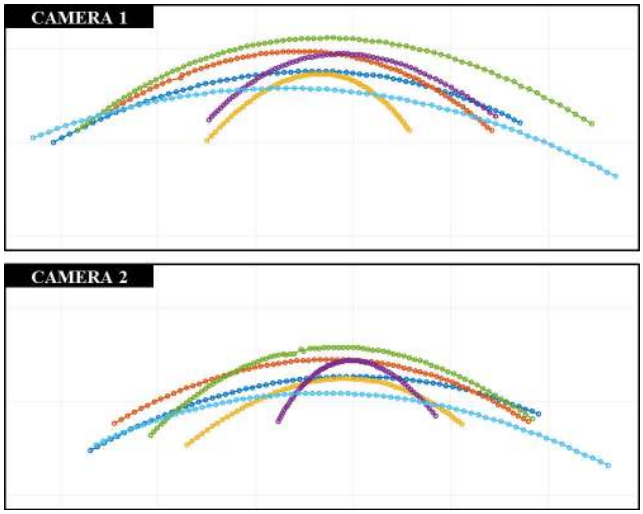

FIGURE 21. Detected 2D baseball points for extrinsic calibration.

same color and the baseball points between the two cameras correspond in the order of left to right within the individual same trajectory.

In Figure 21, we can see that the noise sources considered in the simulation situation are included in the 2D center of the baseball detection results. In particular, some baseball-center detection results deviated from the trajectory because shadows occurred on the baseball or the baseball disappeared into a background of a similar color. These center detection errors are represented by N_D in the previous process and this error can be corrected in the trajectory modeling process as tested in the simulation experiment. We apply the optimization method by modifying the cost function of the

TABLE 3. Calibration results in a real-world experiment ($\tau_2 = \frac{1}{3}$*frame*).

Fitting model	Trial	τ_2^* (frame)	$\lvert\tau_2 - \tau_2^*\rvert$ (frame)	3D reconstruction error (pixel)
F1	01	-0.2215	0.1119	0.1704
	02	-0.2737	0.0596	0.1843
	03	-0.3144	0.0190	0.1685
	04	-0.2220	0.1113	0.1688
F2 (proposed)	01	-0.3256	0.0078	0.1116
	02	-0.3259	0.0074	0.1085
	03	-0.3432	0.0098	0.1079
	04	-0.3242	0.0091	0.1111

TABLE 4. Calibration results in a real-world experiment ($\tau_2 = \frac{1}{2}$*frame*).

Fitting model	Trial	τ_2^* (frame)	$\lvert\tau_2 - \tau_2^*\rvert$ (frame)	3D reconstruction error (pixel)
F1	01	-0.4165	0.0835	0.1599
	02	-0.4030	0.0970	0.1656
	03	-0.4266	0.0734	0.1620
	04	-0.3409	0.1591	0.1629
F2 (proposed)	01	-0.5078	0.0078	0.0937
	02	-0.4985	0.0015	0.0930
	03	-0.4945	0.0055	0.0951
	04	-0.5085	0.0085	0.0927

TABLE 5. Calibration results in a real-world experiment ($\tau_2 = \frac{2}{3}$*frame*).

Fitting model	Trial	τ_2^* (frame)	$\lvert\tau_2 - \tau_2^*\rvert$ (frame)	3D reconstruction error (pixel)
F1	01	-0.6084	0.0582	0.1655
	02	-0.6462	0.0205	0.1820
	03	-0.6405	0.0261	0.1675
	04	-0.6380	0.0287	0.1632
F2 (proposed)	01	-0.6543	0.0124	0.1123
	02	-0.6782	0.0115	0.1058
	03	-0.6753	0.0086	0.1037
	04	-0.6836	0.0169	0.1018

proposed objective function using a reprojection error instead of the 3D distance error since we cannot measure the real 3D location of the pitching trajectories in the real world. The test was conducted with four different capture time delay settings ($\tau_2 = \frac{1}{3}$ *frame*, $\frac{1}{2}$ *frame*, $\frac{2}{3}$ *frame*, 1 *frame*) and each optimization process was performed four times to evaluate the average performance. The real-world calibration performance was evaluated with the capture delay estimation error and an extrinsic calibration error that was similar to testing in the simulation space. The real-space extrinsic calibration error was calculated using the average reprojection error of all trajectory points. Tables 3–6 show the performance evaluation results of each capture delay parameter.

As shown in Tables 3–6, the proposed method achieves the highest accuracy in capture delay estimation and extrinsic calibration results for all capture delay settings. In particular, the real-world test results are very similar to the experimental results in the simulation environment, which indicates that the calibration method of the unsynchronized system that is verified in the simulation environment can be used effectively

TABLE 6. Calibration results in a real-world experiment ($\tau_2 = 1$*frame*).

Fitting model	Trial	τ_2^* (frame)	$\lvert\tau_2 - \tau_2^*\rvert$ (frame)	3D reconstruction error (pixel)
F1	01	-0.8337	0.1663	0.1721
	02	-1.1912	0.1912	0.2263
	03	-0.8777	0.1223	0.1725
	04	-0.9183	0.0817	0.1701
F2 (proposed)	01	-1.0010	0.0009	0.0931
	02	-0.9978	0.0022	0.0915
	03	-0.9933	0.0067	0.0914
	04	-1.0075	0.0075	0.0920

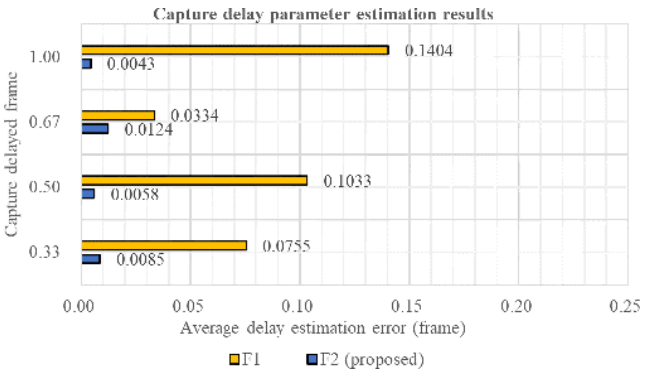

FIGURE 22. Errors in average capture delay parameter estimation.

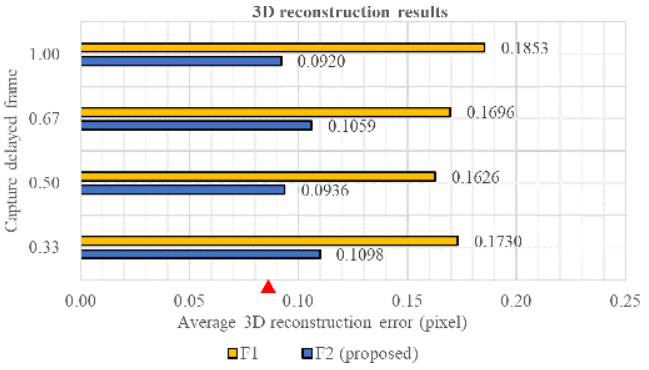

FIGURE 23. Errors in average 3D reconstruction. The 3D reconstruction error of the synchronized system is denoted by a red triangle.

in real-world situations. Figure 22 and 23 summarizes the results. It shows that the average capture delay estimation error of the proposed method is 0.0077 frames and that of the previous method is 0.0881 frames; the proposed method can estimate the capture delay parameters about 10 times more accurately than the previous method. In addition, the average calibration error of the proposed method is 0.1003 pixels and that of the previous method is 0.1726 pixels. Considering that the calibration accuracy of the synchronized system is 0.083 pixels, the proposed method is able to reconstruct the 3D pitching trajectories to error of 0.02 pixels only whereas the previous method shows an error of about 0.09 pixels. The performance comparison between previous method and proposed method as summarized in Table 7.

TABLE 7. Comparison of the experimental results between the proposed method and [8] in real-world experiments.

	[8]	Proposed method	Performance improvement
Capture delay estimation error (frame)	0.0881	0.0077	10× capture delay estimation accuracy improvement
3D reconstruction error (pixel)	0.1726	0.1003	41% reduction in the 3D reconstruction errors
3D reconstruction error compared with the synchronized system (pixel)	0.09	0.02	77% reduction in the 3D reconstruction errors of the proposed method compared to the synchronized system

V. CONCLUSION

We have developed a 3D trajectory reconstruction method involving modeling the motion of a baseball using piece-wise spline function and optimizing the optimal capture delay parameters in a multiple unsynchronized camera system. We derived a method to analyze the capture process of multiple unsynchronized cameras in a simulation space that mimics the real-world environment. To predict the position of a baseball at a subframe level, and estimate the optimal capture delay parameters precisely, we defined an objective function for capture delay and estimated the capture delay using a non-linear optimization algorithm. The performance of the proposed method has been evaluated in both simulation and real-world situations, and the experimental results show that the proposed method reliably reconstructs a 3D baseball trajectory even in the presence of capture delay.

Compared with other methods, the proposed method provides an effective trajectory reconstruction algorithm which analyzes the multiple unsynchronized camera capture system in the simulation space. The proposed system has shown high accuracy in 3D trajectory reconstruction, due to the use of a piece-wise spline model that can accurately model the trajectory of a baseball in a real baseball game situation. In future work we will consider approaches to perspective transformations of the camera, in order to further improve the modelling of 2D baseball trajectories.

REFERENCES

[1] M. Lage, J. P. Ono, D. Cervone, J. Chiang, C. Dietrich, and C. T. Silva, "Statcast dashboard: Exploration of spatiotemporal baseball data," *IEEE Comput. Graph. Appl.*, vol. 36, no. 5, pp. 28–37, Sep./Oct. 2016, doi: 10.1109/mcg.2016.101.
[2] J. Kim, "Ball trace providing system for real time broadcasting," KR Patents 10-1 291 765, Aug. 2013.
[3] T. Fredrik, "Determination of spin parameters of a sports ball," U.S. Patents 8 845 442 B2, Sep. 30, 2014.
[4] P. Design. (2016). *Rapsodo, Trackman, and Pitch Tracking Technologies—Where We Stand*. [Online]. Available: https://www.drivelinebaseball.com/2016/11/rapsodo-trackman-pitch-tracking-technologies-stand/
[5] K. S. K. Yeo, B. Okur, L. R. Vijayanand, "Systems and methods of analyzing moving objects," U.S. Patents 9 955 126 B2, Apr. 24, 2018.
[6] M. S. White and A. Alt, "Tracking an object with multiple asynchronous cameras," U.S. Patents 8 705 799 B2, Apr. 22, 2014.
[7] Y. Zhang, J.-S. Kim, and W.-Y. Kim, "A novel method of stereo camera calibration for baseball pitching trajectory," in *Proc. ICCE*, Seoul South Korea, Oct. 2016, pp. 26–28.
[8] S. Miyata, H. Saito, K. Takahashi, D. Mikami, M. Isogawa, and H. Kimata, "Ball 3D trajectory reconstruction without preliminary temporal and geometrical camera calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, Hi, USA, Jul. 2017, pp. 164–169, doi: 10.1109/cvprw.2017.26.
[9] V. Renò, N. Mosca, M. Nitti, T. D'Orazio, C. Guaragnella, D. Campagnoli, A. Prati, and E. Stella, "A technology platform for automatic high-level tennis game analysis," *Comput. Vis. Image Understand.*, vol. 159, pp. 164–175, Jun. 2017, doi: 10.1016/j.cviu.2017.01.002.
[10] A. Kumar, P. S. Chavan, V. K. Sharatchandra, S. David, P. Kelly, and N. E. O'Connor, "3D estimation and visualization of motion in a multi-camera network for sports," in *Proc. Irish Mach. Vis. Image Process. Conf.*, Sep. 2011, pp. 15–19, doi: 10.1109/imvip.2011.12.
[11] Y. Piao, and J. Sato, "Computing epipolar geometry from unsynchronized cameras," in *Proc. 14th Int. Conf. Image Anal. Process. (ICIAP)*, Modena, Italy, Sep. 2007, pp. 10–14, doi: 10.1109/iciap.2007.4362823.
[12] M. Noguchi and T. Kato, "Geometric and timing calibration for unsynchronized cameras using trajectories of a moving marker," in *Proc. IEEE Workshop Appl. Comput. Vis. (WACV)*, Austin, TX, USA, Feb. 2007, pp. 21–22, doi: 10.1109/wacv.2007.27.
[13] H. Matsumoto, J. Sato, and F. Sakaue, "Multiview constraints in frequency space and camera calibration from unsynchronized images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 1601–1608, doi: 10.1109/CVPR.2010.5539779.
[14] S. Tamaki and H. Saito, "A calibration method for performance analysis in table tennis," in *Proc. 20th Korea-Japan Joint Workshop Frontiers Comput. Vis.*, 2014, pp. 330–334.
[15] C. Albl, Z. Kukelova, A. Fitzgibbon, J. Heller, M. Smid, and T. Pajdla, "On the two-view geometry of unsynchronized cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4847–4856, doi: 10.1109/CVPR.2017.593.
[16] C. Gu, G. Sun, Y. Fen, D. Tian, Y. Peng, X. Li, and Y. Cong, "Motion-based pose estimation via free falling," in *Proc. 33rd Youth Academic Annu. Conf. Chin. Assoc. Automat. (YAC)*, Nanjing, China, May 2018, pp. 673–676, doi: 10.1109/YAC.2018.8406457.
[17] S. Alderson. (2019). *Official Baseball Rules*. [Online]. Available: https://content.mlb.com/documents/2/2/4/305750224/2019_Official_Baseball_Rules_FINAL_.pdf
[18] R. Hartley, and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge, U.K.: Cambridge Univ. Press, 2003.
[19] R. Cavallaro, M. Hybinette, M. White, and T. Balch, "Augmenting live broadcast sports with 3D tracking information," *IEEE MultiMedia*, vol. 18, no. 4, pp. 38–47, Apr. 2011, doi: 10.1109/mmul.2011.61.
[20] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–777, Jun. 2004, doi: 10.1109/tpami.2004.17.
[21] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000, doi: 10.1006/cviu.1999.0832.
[22] W. C. Davidon, "Variable metric method for minimization," *SIAM J. Optim.*, vol. 1, no. 1, pp. 1–17, 1991, doi: 10.1137/0801001.
[23] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–mead simplex method in low dimensions," *SIAM J. optimization*, vol. 9, no. 1, pp. 112–147, 1998, doi: 10.1137/s1052623496303470.
[24] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.
[25] J. Kim and W.-Y. Kim, "Baseball detection method robust to occlusion in hitting area based on high speed camera," in *Proc. IW-FCV*, Tokyo, Japan, Feb. 2018, pp. 21–23.
[26] C. R. de Boor, "A practical guide to splines," in *Applied Mathematical Sciences*. New York, NY, USA: springer-verlag, 1978, doi: 10.1007/978-1-4612-6333-3.
[27] J. Lundgren. (Jun. 14, 2019). *SPLINEFIT MATLAB Central File Exchange*. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/71225-splinefit
[28] H. Lee, J. Kim, J. Kim, J. Yu, and W.-Y. Kim, "Start-end time detection in baseball videos for automatic pitching trajectory analysis," in *Proc. Int. Conf. Electron. Inf. Commun. (ICEIC)*, Auckland, New Zealand, Jan. 2019, pp. 1–4, doi: 10.23919/elinfocom.2019.8706498.

JOONGSIK KIM received the B.S. degree in information and communication electronic engineering from Soongsil University, Seoul, South Korea, in 2013. He is currently pursuing the Ph.D. degree in electronics and computer engineering with Hanyang University, Seoul. His research interests include machine learning, parking assist systems, stereo vision, and 3D trajectory analysis.

MOONSOO RA received the B.S. and Ph.D. degrees in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2011 and 2019, respectively. He is currently working as a Founding Member and CTO of LightVision Inc. His research interests include pattern recognition, machine learning, autonomous vehicle, and video surveillance.

HONGJUN LEE received the B.S. degree in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include generative adversarial networks, data augmentation, perspective distortion correction, and stereo vision.

JEYEON KIM received the B.S. degree in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include face recognition, deep learning architecture, stereo vision, and depth estimation.

WHOI-YUL KIM received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1989. From 1989 to 1994, he was with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas. He joined Hanyang University, in 1994, where he is currently a Professor with the Department of Electronic Engineering. His research interests include health monitoring using mobile devices, visual surveillance, virtual devices, machine vision systems, advanced driver assistance systems, and 3D vision systems in sports as well as home appliances.