

# Vision-Based 3D Baseball Swing Trajectory Reconstruction and Swing Performance Analysis

Yi-Hsuan Lu<sup>[0009-0009-8035-0476]</sup> and Wei-Ta Chu<sup>[0000-0001-5722-7239]</sup>

National Cheng Kung University, Tainan, Taiwan  
p76124126@gs.ncku.edu.tw, wtchu@gs.ncku.edu.tw

**Abstract.** This study presents a 3D baseball swing reconstruction system that estimates 3D swing trajectories based on synchronized dual-view videos. It is a pure vision-based method without the requirement of other sensors. 2D keypoints of the baseball bat are detected, and 3D swing trajectories are constructed based on associated keypoints from two views. Based on 3D swing trajectories, we calculate several swing metrics like attack angle and bat speed, and compare them with commercially available sensors to show the effectiveness of the proposed method. Furthermore, we introduce the BaseballSwing3D dataset that contains synchronized dual-view swing videos associated with 3D coordinates of the bat's head point and the tail point. This dataset serves as a quantitative benchmark for validating the effectiveness of a 3D swing reconstruction method.

**Keywords:** Keypoint detection · 3D reconstruction · Swing trajectory · Swing metrics.

## 1 Introduction

Digitizing athletes' actions and analyzing multimodal data have significantly advanced athletes' training strategy, performance evaluation, and team tactics analysis. In this work, we focus on analyzing swing behaviors in baseball through computer vision techniques.

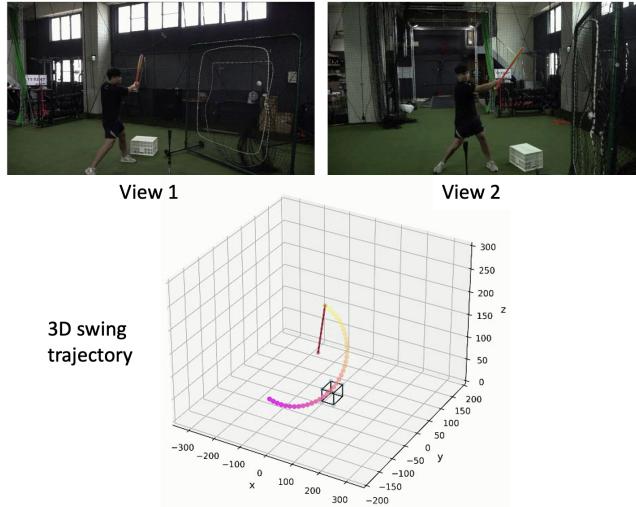
Computer vision technologies enable non-intrusive motion tracking without requiring players to wear any devices. For example, KinaTrax<sup>1</sup> employs multi-view high frame rate synchronized cameras to capture pitching and batting motion. Multi-view motion tracking systems such as Hawk-Eye<sup>2</sup> have been widely adopted in multiple sports events like baseball and tennis for tracking ball trajectories, athlete movements, and foul ball detection.

Although the Hawk-Eye system is one of the most prevalent commercial systems, it relies on multi-view synchronized camera arrays with high installation costs and professional calibration to ensure accuracy. In this work, we aim to reconstruct the 3D swing motion of a batter using two arbitrarily positioned cameras. This is a much lower-cost setting that allows easy installation

---

<sup>1</sup> <https://www.kinatrax.com>

<sup>2</sup> <https://www.hawkeyeinnovations.com>



**Fig. 1.** A sample 3D swing trajectory constructed from videos in two different views.

and maintains high-precision 3D reconstruction. We will mainly reconstruct 3D swing trajectories based on dual-view videos. Figure 1 shows a sample 3D swing trajectory constructed from videos in two different views.

Based on the 3D trajectories, we calculate several swing metrics and quantitatively compare them with the data estimated by a commercial system. To further quantitatively validate the effectiveness of 3D reconstruction, we collect a 3D swing dataset called *BaseballSwing3D*, which includes synchronized dual-view videos and 3D annotations.

Overall, our contributions can be summarized in the following:

- 3D swing reconstruction: We detect keypoints from each view, and filter out noise considering the characteristics of the baseball bat. Keypoints from two views are associated, and 3D coordinates of these keypoints are estimated over time to construct 3D swing trajectories.
- Swing metrics: Based on 3D swing trajectories, we calculate several important swing metrics and verify the accuracy of these metrics by comparing them with existing commercial products.
- The *BaseballSwing3D* dataset: We introduce a dual-view baseball swing dataset for benchmarking 3D swing trajectories.

## 2 Related Works

### 2.1 2D Keypoint Detection

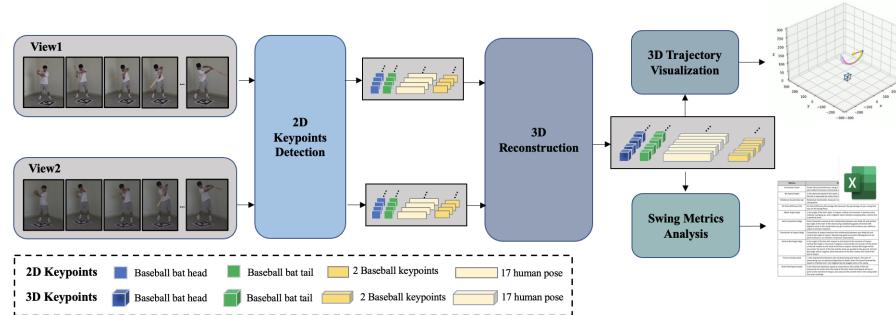
2D object keypoint detection aims to accurately localize keypoints of objects from images and has been widely applied in sports analysis, human pose estimation, and motion capture.

To balance accuracy and inference speed, YOLO-Pose [7] was developed based on the YOLO (You Only Look Once) [11] object detection framework to provide real-time human keypoint detection. YOLO-Pose [7] leverages a single-network end-to-end training strategy, allowing it to simultaneously detect objects and keypoints without requiring additional components. In this work, we fine-tune the YOLO-Pose model to detect keypoints on the player and the baseball bat.

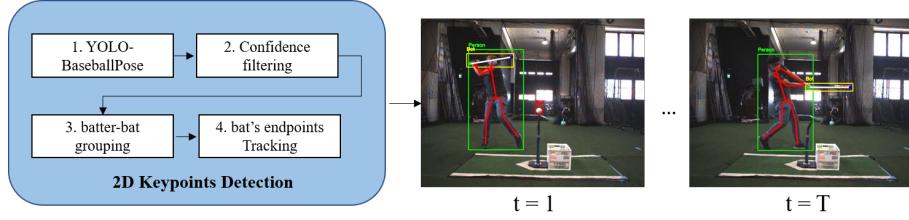
## 2.2 3D Reconstruction

3D reconstruction aims to recover three-dimensional structures from 2D images, where triangulation-based methods play a crucial role in this process. The triangulation method leverages multi-view geometry to compute 3D coordinates through geometric constraints and camera parameters and is widely applied in sports analysis, motion capture, autonomous driving, and robotic vision. Traditional triangulation-based 3D reconstruction [3] relies on multi-view calibrated cameras and epipolar geometry to estimate depth and reconstruct 3D object structures. Structure-from-Motion (SfM) [13, 12] reconstructs a sparse 3D point cloud by estimating camera poses and scene structure. SfM is primarily based on detecting and matching feature points across multiple images. However, SfM can only generate sparse point clouds, making it unsuitable for applications requiring accurate keypoint positions. In this work, we mainly rely on the triangulation method to do 3D reconstruction.

Specific to baseball, although there have been some works related to the pitcher's pose [1] or the hitter's pose [10][9], there is little work directly related to 3D baseball swing analysis. For golf, most works focus on the pose of the player [6][5]. Therefore, in this work, we thus focus on 3D reconstruction for baseball swings.



**Fig. 2.** The framework of the proposed system consists of four modules: 2D keypoints detection, 3D reconstruction, 3D trajectory visualization, and swing metrics analysis.



**Fig. 3.** The proposed filtering process for 2D keypoint detection.

### 3 Methodology

We aim to develop a low-cost, high-precision baseball swing analysis system. As shown in Figure 2, the system consists of four main stages: 2D keypoint detection, 3D reconstruction, swing metrics analysis, and 3D trajectory visualization.

Given two synchronized videos from two different views that contain the complete swing process, we first employ a 2D keypoint detector to estimate the locations of keypoints of the player and the endpoints of the bat. Then, based on the spatial relationship between two cameras, the 2D keypoint sequences are mapped to obtain a 3D keypoint sequence. Finally, the 3D keypoint sequence is visualized to show a 3D swing trajectory.

#### 3.1 Keypoint Detection and 3D Reconstruction

We adopt the YOLO-Pose model [7] to detect keypoints on the player and the baseball bat. This detector was primarily designed for detecting human pose. Therefore, we fine-tune the YOLO-Pose architecture using a custom annotated dataset that includes the two endpoints of the baseball bat, i.e., the bat head and the bat tail. The fine-tuned detector is referred to as YOLO-BaseballPose.

During the inference stage, there may be other people or other baseball bats in the background. To reduce these noises, we propose a filtering process to maintain highly accurate detection, as illustrated in Figure 3. The object set detected by the YOLO-BaseballPose model in a video frame is denoted as  $\mathcal{D}$ :

$$\mathcal{D} = \{O_i = (\mathbf{B}_i, \mathbf{K}_i, c_i, s_i) \mid i = 1, 2, \dots, N\} \quad (1)$$

where  $\mathbf{B}_i = (x_i, y_i, w_i, h_i)$  is a bounding box,  $\mathbf{K}_i = \{(x_k, y_k, v_k) \mid k = 1, 2, \dots, M\}$  is a set of keypoints in  $\mathbf{B}_i$ ,  $c_i$  denotes the class of this object, i.e., a human or a baseball bat, and  $s_i$  denotes the confidence value of the object  $O_i$ . The number of keypoints  $M$  in each bounding box  $\mathbf{B}_i$  differs depending on the object category. A human has 17 keypoints, while a baseball bat has 2 keypoints, representing the bat head and bat tail. Based on these results, we first filter out objects with low confidence scores by setting a confidence threshold  $\tau_s$ . Next, to filter out irrelevant people, we consider the association between each detected person and each detected bat. Specifically, we calculate the Euclidean distance between the

wrist keypoint of a detected human and a detected bat's tail keypoint. If the Euclidean distance is larger than a pre-defined threshold  $\tau_d$ , this batter-bat pair is invalid and ignored. If multiple batter-bat pairs are valid, only the pair with the smallest distance is selected. This filtering process makes the proposed system more practical to be used in a real batting practice environment.

Assume that we have calibrated cameras of two views and obtained the intrinsic and extrinsic parameters. Through the above process, we finally get the baseball bat's head points  $\mathbf{H}^{(1,t)}$  and  $\mathbf{H}^{(2,t)}$  from the  $t$ th frames of the 1st and the 2nd views, respectively. We can then employ the triangulation method [3] to estimate the 3D coordinate of the bat's head based on the intrinsic and extrinsic camera parameters. Similarly, we can estimate the 3D coordinate of the bat's tail based on the tail points  $\mathbf{L}^{(1,t)}$  and  $\mathbf{L}^{(2,t)}$  from two views. By connecting the temporal sequences of 3D head points and tail points, respectively, we can get the 3D trajectory of the bat. Figure 1 shows the 3D trajectory of the bat's head in a swing.

In fact, we also estimate the 3D coordinates of the batter's 17 keypoints in the same way. We will show the visualization results of both the player and the bat in the evaluation section.

### 3.2 Swing Metrics

The 3D reconstruction process mentioned above is kind of standard. The contribution of this work thus mainly comes from the quantitative process for swings described below. Several key metrics are widely used to evaluate swing performance. While several commercial tools employ sensors to estimate these metrics, most of them have not released rigorous experimental validation in the literature. In this work, we aim to do quantitative verification.

**Bat Speed (BS)**  $\nu_{BS}$ . BS is the observed speed of the sweet spot of the bat at contact (the instant that the bat hits the ball). The sweet spot of the bat is around six inches from the tip of the bat. Let  $\{\mathbf{H}^{(t*)}, \mathbf{L}^{(t*)}\}$  denote the 3D coordinate vectors of the bat's head and tail at time  $t^*$ , respectively. The time  $t^*$  is the instant that the impact occurs. For every time instant  $t = 1, \dots, t^*, \dots, T$ , several important values are calculated first.

- Bat length:

$$D_t = \|\mathbf{H}^{(t)} - \mathbf{L}^{(t)}\|, \quad (2)$$

where  $\|\cdot\|$  denotes the L2 distance. Theoretically, the bat lengths in the 3D space at different time instants are the same. However, they may vary slightly because of detection and 3D reconstruction noises.

- Sweet spot position: The location 6 inches (15.24 cm) from the bat head.

$$\mathbf{S}_t = \mathbf{H}^{(t)} + (\mathbf{L}^{(t)} - \mathbf{H}^{(t)}) \left( \frac{15.24}{D_t} \right). \quad (3)$$

- Hand position: The location 6 inches (15.24 cm) from the bat's tail.

$$\mathbf{A}_t = \mathbf{L}^{(t)} + (\mathbf{H}^{(t)} - \mathbf{L}^{(t)}) \left( \frac{15.24}{D_t} \right). \quad (4)$$

- Angular speed: The change of the bat’s tail in angle over a short time period  $\Delta t$  around the impact. It is estimated by

$$\omega_t = \frac{\|\mathbf{L}^{(t+1)} - \mathbf{L}^{(t)}\|}{\Delta t}, \quad (5)$$

where  $\Delta t = \frac{1}{\text{FPS}}$ , and FPS is the frame rate.

Based on the these values, the bat speed  $V_{BS}$ <sup>3</sup> is calculated as:

$$\nu_{BS} = \omega_t \cdot r_S, \quad (6)$$

where  $r_S = \|\mathbf{S}_{t^*} - \mathbf{L}^{(t^*)}\|$  is the distance from the sweet spot to the bat tail at the time instant  $t^*$ .

**Peak Hand Speed (PHS)**  $\nu_{PHS}$ . PHS is the maximum speed measured near the bat handle (approximately six inches from the knob), typically occurring just before contact as the wrists unhinge [15]. Based on the values mentioned above, the hand speed at the instant  $t$  is calculated as:

$$\nu_{HS,t} = \omega_t \cdot r_P, \quad (7)$$

where  $r_P = \|\mathbf{A}_t - \mathbf{L}^{(t)}\|$  is the distance from the hand position to the bat tail.

The maximum value among all frames is selected as the peak hand speed  $\nu_{PHS}$ :

$$\nu_{PHS} = \max_t \nu_{HS,t} \quad (8)$$

**Vertical Bat Angle (VBA)**  $\theta_{VBA}$ . The VBA measures the inclination of the bat relative to the horizontal plane at the moment of contact, reflecting the bat’s tilt or loft during the swing. We first calculate the bat direction vector:

$$\mathbf{V}_{t^*} = \mathbf{H}^{(t^*)} - \mathbf{L}^{(t^*)}, \quad (9)$$

where  $\mathbf{H}^{(t^*)}$  and  $\mathbf{L}^{(t^*)}$  are the 3D coordinates of the bat’s head and tail at the contact time  $t^*$ , respectively.

Let  $\mathbf{V}_{t^*}^z$  denote the vertical component of  $\mathbf{V}_{t^*}$ . We calculate the inclination angle relative to the horizontal plane:

$$\theta_{VBA} = \arcsin \left( \frac{\mathbf{V}_{t^*}^z}{\|\mathbf{V}_{t^*}\|} \right) \times \frac{180}{\pi}, \quad (10)$$

where  $\|\mathbf{V}_{t^*}\|$  is the magnitude of the vector  $\mathbf{V}_{t^*}$ .

**Attack Angle (AA)**  $\theta_{AA}$  : It is the angle of the swing path relative to the horizontal plane (side view) at contact. Positive angles indicate an upward swing, and negative angles a downward swing. Assuming that the bat contacts the ball at the instant  $t^*$ , we calculate the sweet spot motion vector first:

$$\mathbf{M}_{t^*} = \mathbf{S}_{t^*+1} - \mathbf{S}_{t^*}, \quad (11)$$

---

<sup>3</sup> <https://www.patrickjonesbaseball.com/blog/why-is-bat-speed-so-confusing>

where  $\mathbf{S}_{t^*+1}$  and  $\mathbf{S}_{t^*}$  represent the 3D coordinates of the bat's sweet spot at the  $(t^* + 1)$ th and the  $t^*$ th frames, respectively.

We then calculate the inclination angle relative to the horizontal plane:

$$\theta_{AA} = \arcsin\left(\frac{\mathbf{M}_{t^*}^z}{\|\mathbf{M}_{t^*}\|}\right) \times \frac{180}{\pi} \quad (12)$$

where  $\mathbf{M}_{t^*}^z$  is the vertical component of  $\mathbf{M}_{t^*}$ , and  $\|\mathbf{M}_{t^*}\|$  denotes the magnitude of the motion vector.

### 3.3 3D Trajectory Visualization

In the visualization part, a straight line segment is visualized to connect the head and the tail of the bat, and the motion trajectory of the head is plotted. Sometimes detection noise or miss are unavoidable. To make the trajectory smoother, we filter out noisy keypoints and interpolate missing keypoints.

**Reasonable bat length.** Specifically, the distance between the head and the tail should be within a reasonable range. We calculate the Euclidean distances  $d_t$  between the bat's head  $\mathbf{H}^{(t)}$  and the bat's tail  $\mathbf{L}^{(t)}$  over time. We then sort all distances  $d_t$ 's,  $t = 1, \dots, T$ , and calculate the first quartile ( $Q_1$ ) and the third quartile ( $Q_3$ ) of this distance distribution. The interquartile range is defined as  $IQR = Q_3 - Q_1$ . At the time instant  $t$ , we say the segment connecting the head and the tail is reasonable if its length  $d_t$  meets the criterion:  $Q_1 - 1.5 \times IQR < d_t < Q_3 + 1.5 \times IQR$ . This constraint filters out unusually large or small bat lengths.

**Swing trajectory fitting.** The integral of the bat over the whole swing process can be thought of as a swing plane around the hitter's body<sup>4</sup>. Therefore, we approximate the bat's head trajectory as on a fitted semi-elliptical plane<sup>5</sup>:

$$\begin{bmatrix} x_s(\theta) \\ y_s(\theta) \\ z_s(\theta) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + R \times \begin{bmatrix} a \cos(\theta) \\ b \sin(\theta) \\ c \sin(\theta) \end{bmatrix}, \quad \theta \in [0, \pi] \quad (13)$$

where  $(x_0, y_0, z_0)$  represents the position of the fulcrum of the swing, i.e., the place where the player's hands hold the bat. The parameters  $a$ ,  $b$ , and  $c$  control the major axis, minor axis, and vertical variation of the swing trajectory, respectively. The variable  $\theta$  ranges from 0 to  $\pi$  to describe the complete swing process. The rotation matrix  $R$  is applied to align the trajectory with the semi-elliptical plane.

The rotation matrix  $R$  is constructed using the  $X$ ,  $Y$ , and  $Z$  Euler angles  $(\psi, \theta, \phi)$  [4] and is defined as:

$$R = R_X(\psi)R_Y(\theta)R_Z(\phi) \quad (14)$$

---

<sup>4</sup> <https://reurl.cc/A3de5Q>

<sup>5</sup> <https://rocklandpeakperformance.com/an-analytical-look-at-baseball-swing-plane/>

where the individual rotation matrices are given by:

$$R_X(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 \cos(\psi) & -\sin(\psi) \\ 0 \sin(\psi) & \cos(\psi) \end{bmatrix}, \quad (15)$$

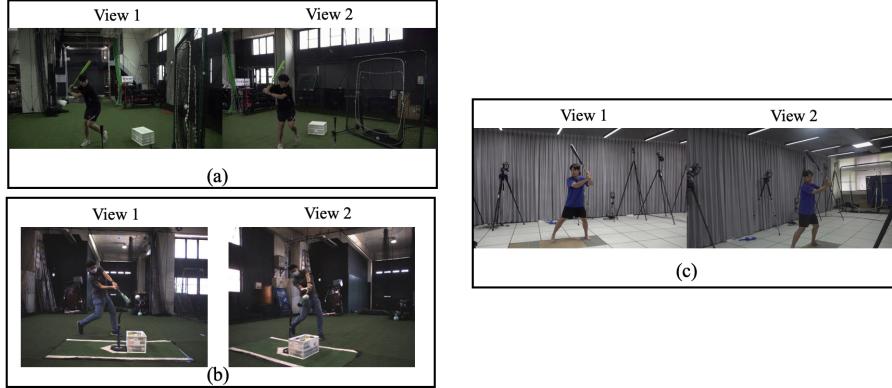
$$R_Y(\theta) = \begin{bmatrix} \cos(\theta) & 0 \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 \cos(\theta) \end{bmatrix}, \quad (16)$$

$$R_Z(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

To determine the parameters  $(x_0, y_0, z_0, a, b, c, \psi, \theta, \phi)$ , we employ the BFGS algorithm [2] to minimize the loss function:

$$\mathcal{L} = \sum_{\theta} \|\mathbf{K}_s(\theta) - \mathbf{K}_{obs}(\theta)\|^2, \quad (18)$$

where  $\mathbf{K}_s(\theta) = (x_s(\theta), y_s(\theta), z_s(\theta))$  represents the coordinates of the fitted swing trajectory, and  $\mathbf{K}_{obs}(\theta)$  denotes the coordinates of the reconstructed (observed) bat's heads over the trajectory. The loss function  $\mathcal{L}$  measures the deviation between the fitted and predicted keypoints and is minimized.



**Fig. 4.** Examples of diverse data in the BaseballSwing3D Dataset.

## 4 BaseballSwing3D Dataset

To validate the effectiveness of our system, we construct the BaseballSwing3D dataset. This dataset includes videos captured from two views, by two types of cameras, and with environmental variations.

**Table 1.** Dataset collection settings in different scenarios.

Subset	(a)	(b)	(c)
Camera	Consumer	Industry	Consumer
#videos (per view)	4	30	102 (static)
#frames (per video)	1.5k	0.6k	0.5k
3D GT	No	No	Yes
Frame rate	120	30	120
Blast Motion	No	Yes	No
Hitting Ball	Yes	Yes	No

We collect data in various environments and with different setups, as illustrated in Figure 4 and Table 1. We employ two types of cameras: consumer video cameras (Sony FDR-AX700) and industrial cameras (IMAGINGSOURCE DFK 33GX287). About the filming environments, swings in both indoor and outdoor scenes, with high-light and low-light conditions, are captured. Careful camera calibration is performed to ensure accurate alignment between dual-view cameras and maintain high-quality 3D reconstruction.

In the subset (a), from each of the two views, 4 videos were captured by a Sony FDR-AX700 camera. The frame rate is 120. In the subset (b), from each of the two views, 30 videos were captured by an IMAGINGSOURCE DFK 33GX287 camera. The frame rate is 30. In both subsets, the player swings to hit the ball on a batting tee. Notice that the environments and the hitters in (a) and (b) are different (see Figure 4(a) and Figure 4(b)). In addition, in the subset (b), the player swings with a Blast Motion sensor attached to the tail of the bat, so that swing metrics can be collected.

In the subset (c), from each of the two views, 102 videos were captured by a Sony FDR-AX700 camera. This subset is specially designed to collect 3D ground truths with the Vicon Motion Capture System. The player is asked to show the swing pose at different time instants. He is asked to keep static when showing the pose (see Figure 4(c)) so that the Vicon system can more accurately capture the 3D coordinates of the bat’s head and tail. In this subset, the player doesn’t hit a ball but just shows different hitting poses.

Note that although theoretically the Vicon system can capture 3D coordinates of the bat’s head and tail in dynamic swings, it is very time-consuming to manually fix capturing errors in a large number of video frames. Moreover, we need to carefully synchronize the Vicon’s cameras and our Sony cameras. These heavy environmental settings limit us to static batting poses in the subset (c).

## 5 Evaluation

### 5.1 Experimental Settings

We fine-tune YOLOv8-Pose for keypoint detection. The fine-tuned model, referred to as YOLO-BaseballPose, is capable of simultaneously detecting 17 human keypoints and 2 keypoints on the baseball bat (the head and the tail).

We evaluate 3D keypoint detection in terms of Mean Per Joint Position Error (MPJPE), which quantifies the average Euclidean distance between the predicted 3D keypoints and the ground-truth 3D keypoints. We also compute the Procrustes aligned Mean Per Joint Position Error (PMPJPE). Before calculating PMPJPE, the predicted 3D keypoints and ground truth were optimally aligned using a rigid transformation that includes rotation, scaling, and translation. This alignment process eliminates absolute positional discrepancies, allowing the evaluation to focus purely on the relative positions between keypoints.

We compare the estimated swing metrics with those from Blast Motion [8], a widely used commercial batting analysis tool. This comparison shows how accurately our vision-based system can achieve, compared to commercial devices.

**Table 2.** Performance of 3D reconstruction of the bat’s head and tail keypoints, in terms of MPJPE and PMPJPE in cm.

Dataset	Baseball Bat Head		Baseball Bat Tail	
	MPJPE ↓	PMPJPE ↓	MPJPE ↓	PMPJPE ↓
Subset (c)	9.99 ± 2.03	6.01 ± 2.14	6.86 ± 1.82	2.10 ± 0.91

## 5.2 3D Keypoint Detection

Based on the subset (c), we validate the 3D keypoint detection results in terms of MPJPE and PMPJPE, by calculating the difference between the detected keypoints with the ground truths collected by Vicon.

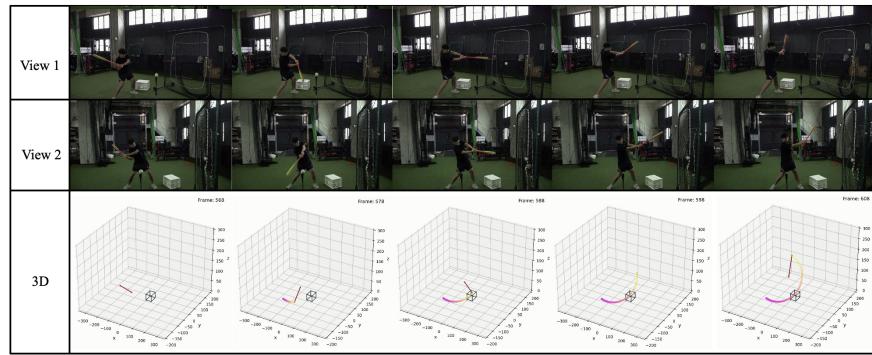
Table 2 shows that, on average, the PMPJPE for the bat’s head is around 6 cm, and the PMPJPE for the tail is around 2 cm. Note that the diameter of the bat’s head is 7 cm. This confirms the effectiveness of our low-cost vision method, not to mention that the Vicon system also has measurement errors.

## 5.3 Swing Metrics

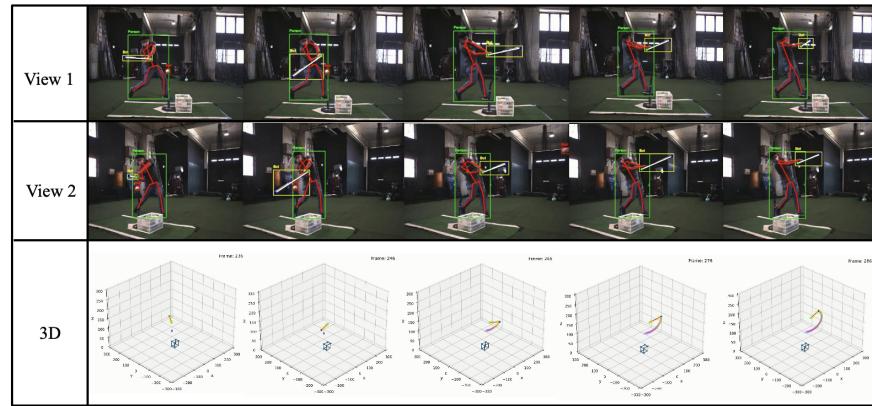
To validate the estimated swing metrics, we use Blast Motion as a reference due to the lack of ground-truth data. A Blast Motion [8] IMU sensor was attached to the batter’s bat handle to collect these metrics. The validation involves the same batter using different bats in different colors and with different weights.

Table 3 shows differences between the estimated swing metrics and Blast Motion’s sensor, based on the subset (b). BS, PHS, AA, and VBA denote the bat speed, peak hand speed, attack angle, and vertical bat angle, respectively. Al Bat means Aluminum Bat. The average differences for the four metrics are 4.37 MPH (miles per hour), 2.52 MPH, 9 degrees, and 2.67 degrees, respectively. These results demonstrate the comparability of our vision-based method with a commercial tool commonly used by professional baseball teams.

Are such estimations correct enough? From the coach's or the player's perspective, it is difficult to say what degree of estimation errors is tolerable. In [14], they measured the difference between the values output by the Blast Motion sensor and the values estimated by a Vicon system. Their results showed that the differences in BS, AA, and VBA are 4 to 6 miles per hour, 2 to 3.5 degrees, and 1.6 to 3.2 degrees (PHS is not provided). Comparing our estimation differences with these values, we can infer that our low-cost vision-based method is comparable with commercial Blast Motion sensors.



**Fig. 5.** A sample 3D swing trajectory reconstructed from videos in the subset (a).



**Fig. 6.** A sample 3D swing trajectory and the player's skeleton reconstructed from videos in the subset (b).

**Table 3.** Comparison of swing metrics difference between our system and Blast Motion [8].

Bat Type	BS Error (MPH)	PHS Error (MPH)	AA Error (degrees)	VBA Error (degrees)
AI Bat Blue	4.0	3.1	11	4
AI Bat Green	0.5	1.8	10	1
AI Bat Coffee	0.9	6.0	6	2
AI Bat Gray	2.1	3.3	13	0
Wooden Bat 1	9.6	0.4	14	4
Wooden Bat 2	9.1	0.5	0	5
<b>Average Error</b>	<b>4.37</b>	<b>2.52</b>	<b>9</b>	<b>2.67</b>

#### 5.4 3D Trajectory Visualization

We visualize the reconstructed 3D swing trajectories in Figure 5. Our vision-based system accurately reconstructs complete swing trajectories in low-light conditions. We can also visualize the skeleton of the player and the bat, as illustrated in Figure 6. This visualization allows coaches to analyze the player’s swing movements, adjust attack angles, and obtain data-driven insights for improving the player’s techniques.

## 6 Conclusion

We have presented a system that analyzes videos from two views to accurately reconstruct the 3D trajectory of a baseball bat during the swing. We detect 2D keypoints from each view, and reconstruct 3D coordinates of the bat’s head and tail over time. Based on the 3D trajectories, we calculate several swing metrics and compare them with a commercial sensor to validate the effectiveness of the proposed estimation.

In addition, we collect the BaseballSwing3D dataset, encompassing a wide range of swing scenarios captured under diverse conditions. Associated with precise 3D annotations, this dataset serves as a reliable benchmark for future research in 3D swing analysis.

**Acknowledgement.** This work was funded in part by the National Science and Technology Council, Taiwan, under grants 114-2622-E-006-028, 114-2221-E-006-047-MY3, 114-2425-H-006-004, 114-2637-8-218-002, 113-2622-E-006-029, 113-2634-F-006-002, and 112-2221-E-006-136-MY3.

## References

- Chiu, Y.W., Huang, K.T., Wu, Y.R., Huang, J.H., Hsu, W.L., Wu, P.Y.: 3d baseball pitcher pose reconstruction using joint-wise volumetric triangulation and baseball customized filter system. *IEEE Access* **12**, 117110–117125 (2024)

2. Chong, E.K.P., Zak, S.H.: An Introduction to Optimization. John Wiley & Sons Inc. (2013)
3. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
4. Hashim, H.A.: Special orthogonal group  $so(3)$ , euler angles, angle-axis, rodriguez vector and unit-quaternion: Overview, mapping and challenges. arXiv:1909.06669 (2024)
5. Lee, M.H., Zhang, Y.C., Wu, K.R., Tseng, Y.C.: Golffpose: From regular posture to golf swing posture. In: Proceedings of International Conference on Pattern Recognition. pp. 387–402 (2024)
6. Liao, C.C., Hwang, D.H., Koike, H.: Ai golf: Golf swing analysis tool for self-training. IEEE Access **10**, 106286–106295 (2022)
7. Maji, D., Nagori, S., Mathew, M., Poddar, D.: Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2637–2646 (2022)
8. Motion, B.: Swingtracker blast motion imu technology. <https://blastmotion.com> (2021), accessed: Feb. 11, 2025
9. Oh, S., Kim, H.: Accurate baseball player pose refinement using motion prior guidance. ICT Express **11**(3), 411–416 (2025)
10. Piergiovanni, A., Ryoo, M.S.: Recognizing actions in videos from unseen viewpoints. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. pp. 4124–4132 (2021)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
12. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4104–4113 (2016)
13. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM siggraph 2006 papers. pp. 835–846 (2006)
14. Stewart, E., Stewart, M., Arachchige, S.N.K.K., Turner, A., V, R.F.B., Knight, A., Johnson, J., Younger, T., Chander, H.: Validation of a bat handle sensor for measuring bat velocity, attack angle, and vertical angle. International Journal of Kinesiology & Sports Science **9**(2), 28–32 (2021)
15. Welch, C.M., Banks, S.A., Cook, F.F., Draovitch, P.: Hitting a baseball: A biomechanical description. Journal of Orthopaedic & Sports Physical Therapy **22**(5), 193–201 (1995)