A Loop-Closure-based Inertial Motion Capture, with Application to Sports Swing Measurements

Kiyoshi Irie[1]

***Abstract*— This study presents a system for measuring motions using an inertial measurement unit (IMU) and a camera. IMU-based motion capture systems are advantageous in terms of cost, but measurable motions are limited in length because of rapidly accumulating errors in the process of double integration. We propose the correction of such errors with the assistance of an external monocular camera. Our approach is similar to graph-based simultaneous localization and mapping for mobile robots, where loops in the trajectory are detected and closed to correct the accumulated errors over the trajectory. We apply the idea to motion estimation by developing and integrating three key technologies: pose-graph optimizer, loop detector, and time synchronizer. We evaluate our system through measurements of table tennis swings. The experimental results show that our sensor fusion approach significantly reduces the errors in inertial motion estimations, with an acceptably small additional cost.**

I. INTRODUCTION

We aim to develop an affordable, easy-to-use motion capture system that can be an alternative to expensive optical motion capture systems. A cost-effective approach to motion estimation is the integration of acceleration and angular velocity measurements from an inertial measurement unit (IMU), which has been used for handy motion capture for sports [1]. For example, systems for measuring golf swing [2] and baseball pitching [3] [4] by using a wearable IMU have been developed. The advantages of IMU-based systems for sports measurements, over optical (camera-based) ones, are twofold. First, IMUs can capture data at high sampling rate, thus they are suitable for measuring fast motions. Second, IMUs are not affected by the problem of occlusions that cameras suffer from.

However, a major drawback of existing IMU-based motion capture systems is that the length of a measurable motion is limited to only a few seconds (i.e., typically only a single swing can be measured). The main causes for the limitation are that the estimation error of double integration rapidly accumulates over time [5] [6]. This research aims to overcome the problem by developing an error correction method, such that we can apply an IMU-based method to longer sets of motions, such as a rally in racket sports.

A. Related Work

The problem of accumulating errors is a major concern in an IMU-based motion estimation. The existing methods typically exploit prior knowledge on the motion. For example, zero velocity update algorithms, which detect when the IMU is motionless to compensate for the velocity drift, are being successfully applied to human gait analysis [7] [8]. The algorithms are based on the knowledge that the foot velocity becomes zero when the foot is in contact with the ground.

If no zero velocity points can be found during the motion, we need a different approach. Sagawa et al. employed a single loop-closure approach for the baseball pitching motion measurement [4]. They constrained the target motion with a known initial state and a known final state to correct the accumulated errors with respect to these constraints.

We need to correct errors without such prior knowledge on the motion to realize a versatile motion capture system. Incorporating external sensors is a straightforward approach. A number of camera–IMU hybrid systems to capture human body motions have been proposed. Such methods typically use multiple cameras [9] or a depth camera [10] [11] to obtain 3D information. IMU measurements are used as auxiliary data.

While systems for tracking the human body motion typically exploit prior knowledge on the human kinematic model, such a model cannot be directly applied to motion tracking of an object. Visual markers, such as those used in optical motion capture systems, can also be used for hybrid motion capture systems. Zhang et al. developed a real-time table tennis racket tracking system by combining a racket-mounted IMU and two high-speed cameras [12]. In their system specially designed visual marks are drawn on the racket surface to accelerate the video data processing.

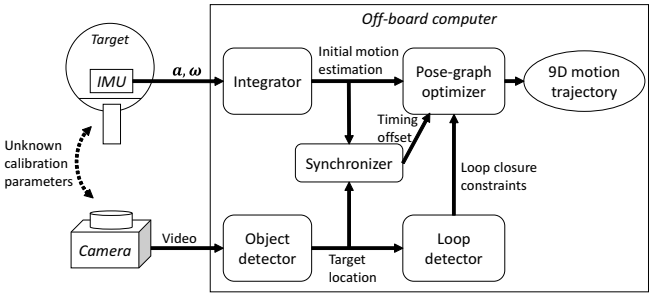


Fig. 1. Overview of our motion capture system. Our system incorporates measurements from an IMU attached to the target and from an external camera pointing at the target.

B. Contribution

This study proposes a novel low-cost, IMU-based motion estimation system for a rigid-body object such as a sports equipment. Our approach for error correction is to combine an external camera and an IMU (see Fig. 1 for the system

[1]Kiyoshi Irie is with Future Robotics Technology Center, Chiba Institute of Technology, 2-17-1, Tsudanuma, Narashino-shi, Chiba, Japan irie@furo.org

overview). Data from the IMU and the camera are fused together based on the framework of a graph-based IMU motion estimation [13], where "loops" in the motion are detected and closed to correct the accumulated errors. The advantages of our camera–IMU system can be summarized as follows:

- a small additional cost compared to pure-inertial methods: only a monocular camera is required, and
- minimized installation effort compared to camera-based methods because no extrinsic camera calibration and no visual markers are required.

We verify our system through experiments and show that the sensor-fusion strategy significantly improves the accuracy of the inertial motion estimation.

II. SYSTEM OVERVIEW

Fig. 1 depicts the overview of our proposed system. We employ an IMU attached on the target object and an external camera pointing at the target to estimate the 9-degree-of-freedom (DoF) time-series state (attitude, velocity, and position) of the target. Our approach for error correction is to detect and close "loops" in the motion [13]. The idea is inspired by graph-based simultaneous localization and mapping (SLAM), which is a map-construction technique for mobile robots [14] [15]. In the graph-based SLAM, the accumulated errors are corrected by detecting "loops" — the places where the robot has visited more than twice. We expand the idea to motion capture and detect the time points when the target has been in the same state during the motion.

We detect loops in the motion by using an external camera pointing at the target. The target object is detected in each video frame using a deep-learning-based object detector. The detected locations and appearances of the object are matched between frames to detect when the loops have occurred. The loop detection algorithm is presented in detail in Section IV.

The detected loop time points must be correctly assigned to the IMU data frames. To this end, we developed a data-driven method that synchronizes two different sensors' clocks. The synchronization algorithm is presented in Section V.

The IMU observations and the detected loops are the input to the pose-graph optimizer. We first obtain initial estimates by integrating the inertial measurements to construct a graph, then correct the accumulated errors in a batch (or offline) manner. The procedure is reported in detail in Section III.

III. POSE-GRAPH OPTIMIZER

A. Problem Definition

We want to estimate a time series of the 9-DoF states (attitude $\boldsymbol{u}$, velocity $\boldsymbol{v}$, and position $\boldsymbol{p}$) of an IMU

$$\{\boldsymbol{x}_i\}_{i=1}^n, \quad \boldsymbol{x}_i := \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{v}_i \\ \boldsymbol{p}_i \end{bmatrix},$$

from the inertial measurements (the 3D acceleration $\{\boldsymbol{a}_i\}_{i=1}^n$ and the 3D angular velocity $\{\boldsymbol{\omega}_i\}_{i=1}^n$) and other external observations.

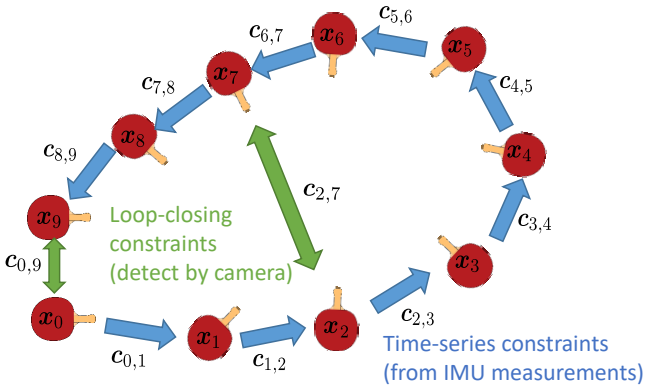


Fig. 2. Sample pose-graph-representation used in our graph-based trajectory estimation. A node $\boldsymbol{x}_i$ denotes a 9-DoF state (position, velocity, orientation) of time i. A constraint $\boldsymbol{c}_{i,j}$ denotes the relative state between the two nodes based on observations. We combine time-series incremental constraints and loop-closing constraints detected using an external camera.

We employ a graph-based formulation for the trajectory estimation problem [13] [16]. Fig. 2 depicts a sample pose graph employed in the formulation. We define a node representing the IMU state for each time frame. We add edges representing the constraints based on the observations on the relative state between two nodes. For each pair of consecutive nodes, we add an edge using the measurements by the IMU. We also add a loop-closing edge between a pair of nodes for which a loop is detected by the external camera. Observation errors are weighted according to their confidence; we find the optimal node sequence that minimizes the weighted sum of the observation errors.

B. Initial Estimate by Naive Integration

We initialize the graph nodes by a naive integration of the inertial data. Assuming that the time step Δt is small and the initial state is given, we incrementally calculate the state of the IMU at each time step as follows:

$$\begin{aligned} \bar{\boldsymbol{u}}_{i+1} &= \bar{\boldsymbol{u}}_i * (\boldsymbol{\omega}_i \Delta t), \\ \bar{\boldsymbol{v}}_{i+1} &= \bar{\boldsymbol{v}}_i + (\boldsymbol{R}_i \boldsymbol{a}_i - \boldsymbol{g}) \Delta t, \\ \bar{\boldsymbol{p}}_{i+1} &= \bar{\boldsymbol{p}}_i + \bar{\boldsymbol{v}}_i \Delta t. \end{aligned}$$

Here, we represent the attitudes by the rotation vectors [17]. The concatenation of the two rotations is denoted by the $*$ operator. $\boldsymbol{R}_i$ represents the rotation matrix representation of the attitude $\boldsymbol{u}_i$; thus, $\boldsymbol{R}_i \boldsymbol{a}_i$ represents the acceleration transformed into the world frame. Note that we need to compensate for the acceleration of the gravity $\boldsymbol{g}$ before integrating accelerations.

C. Three-step Error Correction

The abovementioned initial estimates usually contain significant accumulated errors. We correct these errors using a least-squares method. Rather than optimizing all the 9-DoF states at once, we take three-step procedure in which the attitude, velocity, and position are corrected in this order for computational efficiency [16].

1) Loop Closure for Attitude: First, we correct the accumulated attitude errors throughout the motion. We denote the observed relative attitude between nodes i and j by $\boldsymbol{c}_{i,j}^{(\mathrm{u})}$ and the information matrix (serves as the weight) for the constraint by $\boldsymbol{\Omega}_{i,j}^{(\mathrm{u})}$. We minimize the weighted sum of the squared error as follows:

$$\begin{aligned}\boldsymbol{U}^* &= \min F^{(\mathrm{u})}(\boldsymbol{U}),\\ F^{(\mathrm{u})}(\boldsymbol{U}) &:= \sum_{<i,j>\in\mathcal{C}^{(\mathrm{u})}} \boldsymbol{e}_{i,j}^{(\mathrm{u})}(\boldsymbol{U})^\top \boldsymbol{\Omega}_{i,j}^{(\mathrm{u})} \boldsymbol{e}_{i,j}^{(\mathrm{u})}(\boldsymbol{U}),\\ \boldsymbol{e}_{i,j}^{(\mathrm{u})}(\boldsymbol{U}) &:= (-\boldsymbol{c}_{i,j}^{(\mathrm{u})}) * (-\boldsymbol{u}_i) * \boldsymbol{u}_j.\end{aligned}$$

We apply the Gauss–Newton method for this nonlinear least-squares problem [13].

2) Loop Closure for Velocity: Second, we correct the errors in the velocity sequence. Similar to the attitude, we denote the observed velocity difference between nodes i and j by $\boldsymbol{c}_{i,j}^{(\mathrm{v})}$ and minimize the following objective function as:

$$\begin{aligned}\boldsymbol{V}^* &= \min F^{(\mathrm{v})}(\boldsymbol{V}),\\ F^{(\mathrm{v})}(\boldsymbol{V}) &:= \sum_{<i,j>\in\mathcal{C}^{(\mathrm{v})}} \frac{1}{2}\boldsymbol{e}_{i,j}^{(\mathrm{v})}(\boldsymbol{V})^\top \boldsymbol{\Omega}_{i,j}^{(\mathrm{v})} \boldsymbol{e}_{i,j}^{(\mathrm{v})}(\boldsymbol{V}),\\ \boldsymbol{e}_{i,j}^{(\mathrm{v})}(\boldsymbol{V}) &:= -\boldsymbol{c}_{i,j}^{(\mathrm{v})} - \boldsymbol{v}_i + \boldsymbol{v}_j.\end{aligned}$$

For this problem, unlike the attitude correction, the Hessian matrix $\boldsymbol{H}$ for $F^{(\mathrm{v})}(\boldsymbol{V})$ can be easily calculated (as shown in the Appendix); therefore, we apply the Newton method whose convergence speed is much faster than the Gauss–Newton method.

The procedure of the optimization using the Newton method is described herein. In each iteration step in the Newton method, we calculate $\boldsymbol{H}$ and construct a linear system

$$\boldsymbol{H}\boldsymbol{d} = -\nabla F_{\boldsymbol{V}}^{(\mathrm{v})} \tag{1}$$

and obtain the solution of the system $\hat{\boldsymbol{d}}$. Then, we update the current estimate of velocity list by

$$\hat{\boldsymbol{V}} \leftarrow \hat{\boldsymbol{V}} + \hat{\boldsymbol{d}}. \tag{2}$$

We iterate the procedure until convergence to obtain the optimal velocity sequence $\boldsymbol{V}^*$.

In our experience, 2–3 iterations are usually sufficient. While the Gauss–Newton for attitude correction may require several tens of iterations depending on the data.

3) Close Loop for Position: Finally, we correct the errors in the position sequence. We use the same formulation as the velocity correction:

$$\begin{aligned}\boldsymbol{P}^* &= \min F^{(\mathrm{p})}(\boldsymbol{P}),\\ F^{(\mathrm{p})}(\boldsymbol{P}) &:= \sum_{<i,j>\in\mathcal{C}^{(\mathrm{p})}} \frac{1}{2}\boldsymbol{e}_{i,j}^{(\mathrm{p})}(\boldsymbol{P})^\top \boldsymbol{\Omega}_{i,j}^{(\mathrm{p})} \boldsymbol{e}_{i,j}^{(\mathrm{p})}(\boldsymbol{P}),\\ \boldsymbol{e}_{i,j}^{(\mathrm{p})}(\boldsymbol{P}) &:= -\boldsymbol{c}_{i,j}^{(\mathrm{p})} - \boldsymbol{p}_i + \boldsymbol{p}_j.\end{aligned}$$

We again apply the Newton method for this optimization problem.

D. Constraints

The constraints are defined herein by the observations on the relative state between the two nodes. In our system, we employ three types of constraints: sequential, loop-closing, and "anchor" constraints.

1) Sequential constraints: A sequential constraint is added for each IMU observation at time i by

$$\begin{aligned}\boldsymbol{c}_{i,i+1}^{(\mathrm{u})} &= \boldsymbol{\omega}_i \Delta t\\ \boldsymbol{c}_{i,i+1}^{(\mathrm{v})} &= (\boldsymbol{R}_i \boldsymbol{a}_i - \boldsymbol{g})\Delta t\\ \boldsymbol{c}_{i,i+1}^{(\mathrm{p})} &= \bar{\boldsymbol{v}}_i \Delta t.\end{aligned}$$

These constraints connect the nodes in a chain throughout the trajectory.

2) Loop-closing Constraints: Loop-closing constraints are defined by the external observations on the relative state between a pair of nodes. For example, we construct *same position* constraint as follows if the positions of nodes i and j are the same:

$$\boldsymbol{c}_{i,j}^{(\mathrm{p})} = [0,0,0]^\top.$$

We also incorporate the same idea as a zero velocity update. For example, we add a constraint if the node i is detected as motionless, and the initial velocity is also zero:

$$\boldsymbol{c}_{0,i}^{(\mathrm{v})} = [0,0,0]^\top.$$

How we detect loops from the camera observations are presented in Section IV.

3) Anchor Constraints: We introduce constraints that work like an anchor for ships. The idea behind is that the accuracy of estimation by integration is high at first, and rapidly decreases over time. We globally constrain the nodes for the first few seconds such that their good initial estimates are not destructed by the other constraints. We refer to this type of constraints as the "anchor" constraints. We define a constraint for node i and its time-decaying weight as:

$$\boldsymbol{c}_{0,i}^{(\mathrm{p})} = \bar{\boldsymbol{p}}_i, \quad \Omega_{0,i}^{(\mathrm{p})} = \frac{1}{w_i^2}\boldsymbol{I}_3.$$

In our experiments, we define $w_i \propto i\Delta t$ such that the weight rapidly decreases with time.

IV. LOOP DETECTION

Here, we describe our loop detection algorithm using an external camera. The camera is assumed to record the target object motion. Loops are detected by the similarity of the object in the video frames. Our assumption is as follows: if the objects appears similarly in two images, then the 3D position of the object should also be close.

The loop detection procedure is elaborated herein. First, we detect the target object in each video frame by using YOLO [18], which is a fast deep-learning-based object detector, to obtain the bounding boxes of the target object. Second, we extract the image patches in each frame using the detected bounding boxes. We then perform an exhaustive match between the image patches and evaluate the similarity between the two images patches. We denote the image patch

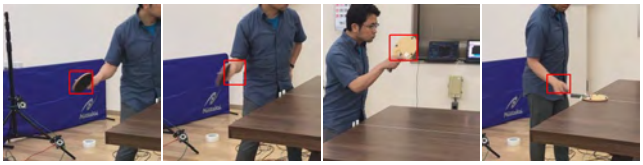


Fig. 3. Sample table tennis racket detection results by YOLO [18].

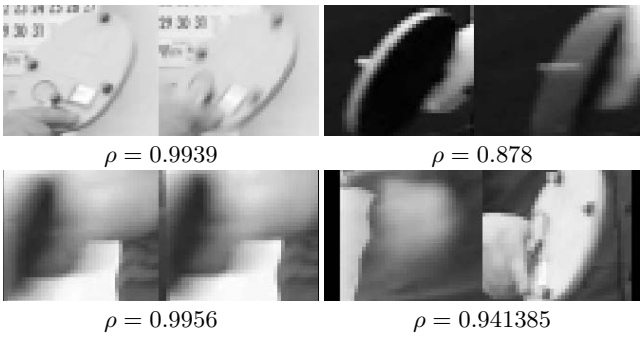


Fig. 4. Sample loop detection results. ρ denotes the cosine similarity between the image patches.

extracted from frame i as I_i. Two frames i and j are considered as a loop if the object locations in the frames are close enough, and the cosine similarity between I_i and I_j

$$\rho_{i,j} = \frac{\sum_{x,y} I_i(x,y) \cdot I_j(x,y)}{\sqrt{\sum_{x,y} I_i(x,y)^2 \cdot \sum_{x,y} I_j(x,y)^2}},$$

is above a threshold (ρ_{thre}). We employed $\rho_{\mathrm{thre}} = 0.98$ in our experiments.

We use two types of loops: we consider the loop as *same position* and zero velocity if the loops are detected between two consecutive frames. We constrain the relative position between two frame to be zero, and the velocity at the frame to also be zero; otherwise, we consider the loop as *same position*, where we only constrain the relative position between two frames.

Fig. 3 and Fig. 4 show the exemplary racket detection results by YOLO and the image patch matching results, respectively. The object detection results by YOLO are not always accurate. We also observed that the object sometimes cannot be detected because of occlusions or severe blur. Nevertheless, we found the loop detection helpful in correcting the inertial estimation errors.

V. Camera-IMU Timing Synchronization

In our system, the data from the IMU and the camera are time-stamped using each sensor's clock. However, they are wirelessly connected; hence, exactly synchronizing the clocks is difficult. We try to determine the time offset between the sensor clocks (δ_T) to synchronize data in post-processing.

Accordingly, we propose a data-driven, multi-sensor timing synchronization method based on statistical dependence. An intuitive explanation for this method is that the data from an IMU and a monocular camera are in different modalities; nevertheless, some correlation must exist between the data because they both originate from the same motion. More specifically, when the target is in motion, both the IMU and the camera should detect some motion, and when the target is motionless, neither of them should detect a motion. Such a correlated relation between the data can be broken when the clocks are unsynchronized. Note that directly comparing two different motions would not work because the 3D to 2D projection is a *nonlinear* process. Hence, our method estimates and maximizes the statistical dependence that can measure such nonlinear correlations.

Our method measures the statistical dependence between the 3D velocity estimations from the IMU and the 2D velocity (in the image coordinate frame) from the camera. The velocity estimates from the IMU and the camera at the synchronized clock time t (using δ_T) are denoted as $\boldsymbol{x}_t^{(\delta_T)}$ and $\boldsymbol{y}_t$, respectively.

Under the assumption that the velocity data pairs $\{\boldsymbol{x}_i^{(\delta_T)}, \boldsymbol{y}_i\}_{i=1}^m$ are generated from the joint probability $p(\boldsymbol{X}^{(\delta_T)}, \boldsymbol{Y})$, we find the optimal time offset by maximizing the objective function f that measures the statistical dependence.

$$\hat{\delta_T} = \underset{\delta_T}{\operatorname{argmax}}\, f(\boldsymbol{X}^{(\delta_T)}, \boldsymbol{Y})$$

For f, we employ squared-loss mutual information (SMI), which is a variation of mutual information (MI). The ordinary MI and SMI are defined as follows:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) := \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},$$

$$\mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y}) := \frac{1}{2} \iint p(\boldsymbol{x})p(\boldsymbol{y}) \left(\frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} - 1 \right)^2 \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}.$$

Both the MI and the SMI take large values when the dependence between two random variables is high. Compared to the MI, the advantage of SMI is that it is robust against noises and outliers because it does not contain a logarithm [19]. The robustness is crucial because as discussed in Section IV, the detected object locations from the camera contain significant noises. We employ LSMI [20] [21], which is a computationally efficient, kernel-based estimator, to estimate the SMI.

VI. Experiments

A. Experimental Setup

We evaluated our system through measurements of table tennis racket swings. Fig. 5 shows the experimental setup. The sensors used were a TDK ICM-20469 attached to a table tennis racket, and a smartphone camera (Apple iPhone 7) mounted on a tripod. Optitrack Prime 13 cameras were used to obtain the ground truth. The data from the IMU and the camera were sent to a laptop computer, and the experiments were conducted offline.

We employed an Espressif ESP-WROOM-02 microprocessor having SPI and WiFi functions to interface the IMU to the computer. Fig. 6 illustrates the specification and outlook of the IMU–microprocessor system.

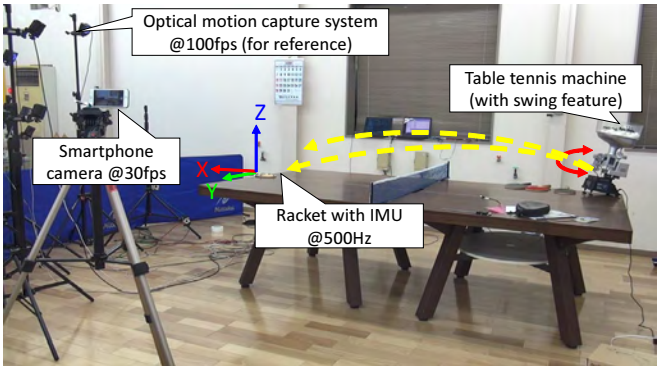
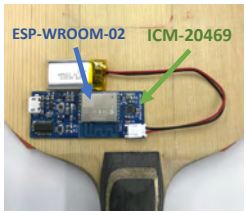


Fig. 5. Experimental setup.



IMU	TDK ICM-20469
Measurement range	±4000°/s ±30g
Data acquisition rate	500Hz
MCU	ESP-WROOM-02
MCU-IMU I/F	SPI
Data transmission	TCP over WiFi

Fig. 6. Outlook and specification of the IMU used in the experiments.

B. Practice Swings without Balls

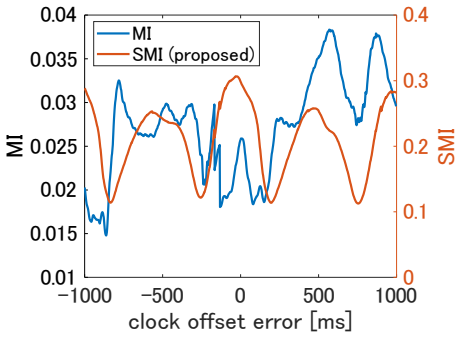
First, we conducted experiments involving the measurement of practice swings without hitting balls. We collected the data of 10 repetitive forehand swings. The datum duration was approximately 15 s and the video recorded had 471 frames.

1) Loop Closure Detection: Our method detected 59 *same position* and 3 *zero velocity* loops in the motion, excluding the loops detected when the racket was placed on the table. See Fig. 4 for sample loop detection results.

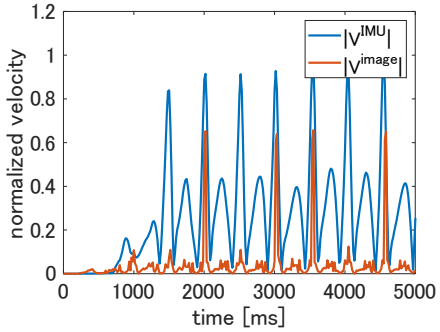
2) Timing Synchronization: Our method successfully synchronized IMU data and video frames. Fig. 7 shows the fluctuation of the statistical dependence measures (MI and SMI) with respect to the clock error. The estimated SMI periodically changed with the same period as the measured motion. The plot of SMI appeared to be smooth compared to MI and exhibited a clear peak around the correct clock offset.

3) Estimation Accuracy: We compared our method with the ground truth and several comparative methods: a) naive integration without error correction, b) the method proposed by Sagawa et al. [4], which corrects the integration error using the known initial and final states. We also compared with trajectories by integration using filtered attitude using c) the Madgwick filter [22] with fixed gain (=0.05).

Table I shows a comparison of the estimation errors of the proposed and comparative methods. Fig. 8 depicts the estimated trajectories. The mean and maximum errors of the proposed system were 0.06 m and 0.12 m, respectively, and were significantly smaller than those in the comparative methods. Because the motion was highly dynamic, the Madgwick filter did not improve the integration accuracy.



a) Comparison of the dependence measures



b) Velocity estimations aligned by the proposed method

Fig. 7. Time synchronization result from the practice swing experiment. While the plot of SMI exhibited a clear peak around the correct clock offset, the MI looked jagged, probably because of the noise in data.

TABLE I

ESTIMATION ERRORS ON THE PRACTICE SWING MEASUREMENT.

	Max error [m]	Mean error [m]
Naive integration	2.32	0.74
Sagawa et al.	0.28	0.16
Madgwick filter	4.53	1.19
Proposed	**0.12**	**0.06**

TABLE II

BREAKDOWN OF THE PROCESSING TIME OF THE PROPOSED SYSTEM.

Object detection	7.5 s
Image patch matching	4.4 s
Clock synchronization	1.7 s
Pose-graph optimization	3.2 s
Total	16.8 s

4) Processing Time: Our system spent a total of 16.8 s to process the data. Table II presents the breakdown. The time can be shorten by performing object detection in real-time.

C. Sequential Strokes Using a Table Tennis Machine

Next, we measured the strokes that hit balls served by a table tennis machine. The machine was configured to swing its ball launcher such that the player needed to use his footwork to hit the balls. The collected motion data included 10 sequential forehand strokes with a duration of approximately 14 s.

This scenario was more challenging than the previous

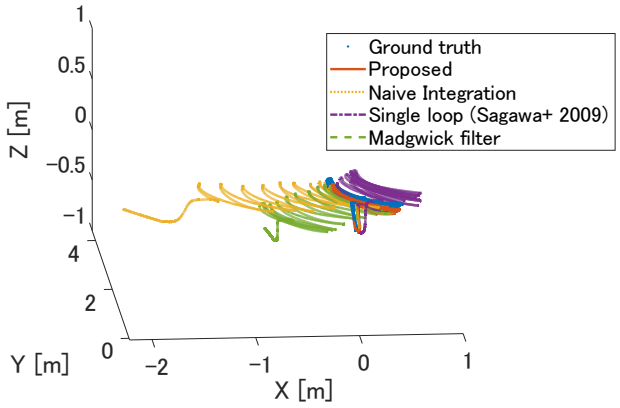


Fig. 8. Estimated trajectories from the practice swing experiments.

TABLE III
ESTIMATION ERRORS ON TABLE TENNIS STROKES WITH BALL HITS.

	Max error [m]	Mean error [m]
Naive integration	60.10	16.14
Sagawa et al.	9.50	4.79
Proposed	**0.37**	**0.15**

one because the player needed to keep moving and adjust his swing for each ball, making the swings more diverse. Furthermore, the ball impacts added noises to the IMU measurements. For these data, 13 *same position* and no *zero velocity* loops were detected by our system.

We again compared our method with the ground truth and the comparative methods. Table III summarizes the errors. The mean and maximum errors of the proposed system were 0.15 m and 0.37 m, respectively. Although all methods resulted in an increased error compared to the previous experiment, ours was the smallest in terms of the degradation amount.

Fig. 9 and Fig. 10 display the estimation results. The shape of the estimated trajectory looked locally accurate, and the estimated racket states matched well with the video. Therefore, the data can be used for skill evaluations, such as analyzing swing directions, velocities, and racket surface orientations.

VII. CONCLUSIONS

This study proposed a low-cost IMU-camera motion capture system. The incorporation of loop observations by a monocular camera significantly improved the accuracy of the inertial motion estimation, even though the motions were not perfectly repetitive. Our system is still not as accurate as optical motion capture systems; nevertheless, the estimation results make us hopeful that the system will be useful for sports skill evaluations.

The remaining topics for further research include exploring how much repetitiveness is needed for the system. We are not yet confident as regards whether or not the system is effective for a more random set of motions like in a table tennis match. Further research could also tackle relaxing the repetitiveness assumption. Enhancing the power

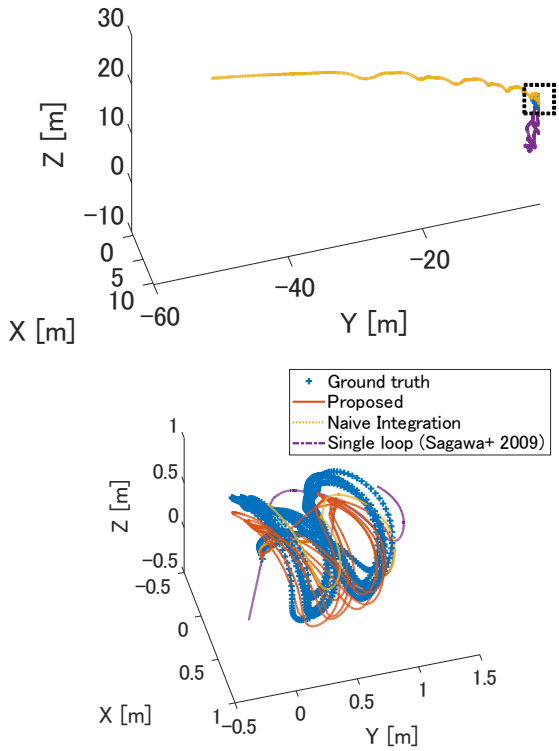


Fig. 9. Top: Estimated trajectories of sequential table tennis strokes with ball hits. Bottom: Zoomed view of the boxed area.

of error correction by incorporating single-view depth observations [23] [24] is a promising direction.

APPENDIX: CALCULATION OF HESSIAN IN EQ. (2)

The gradient of $F^{(\mathrm{v})}(\boldsymbol{V})$ is denoted by

$$\nabla F^{(\mathrm{v})}_{\boldsymbol{V}} := \begin{bmatrix} \frac{\partial F^{(\mathrm{v})}(\boldsymbol{V})}{\partial \boldsymbol{v}_1} \\ \vdots \\ \frac{\partial F^{(\mathrm{v})}(\boldsymbol{V})}{\partial \boldsymbol{v}_n} \end{bmatrix}. \tag{3}$$

The k-th block of the gradient can be calculated as follows:

$$\frac{\partial F^{(\mathrm{v})}(\boldsymbol{V})}{\partial \boldsymbol{v}_k} = \sum_{<i,j>\in\mathcal{C}^{(\mathrm{v})}} \frac{\partial F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_k} \tag{4}$$

$$F^{(\mathrm{v})}_{i,j}(\boldsymbol{V}) := \frac{1}{2}\boldsymbol{e}^{(\mathrm{v})}_{i,j}(\boldsymbol{V})^{\top}\boldsymbol{\Omega}^{(\mathrm{v})}_{i,j}\boldsymbol{e}^{(\mathrm{v})}_{i,j}(\boldsymbol{V}) \tag{5}$$

$$\frac{\partial F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_k} = \boldsymbol{\Omega}^{(\mathrm{v})}_{i,j}\boldsymbol{e}^{(\mathrm{v})}_{i,j}(\boldsymbol{V})\frac{\partial \boldsymbol{e}^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_k}. \tag{6}$$

This can be easily calculated using

$$\frac{\partial \boldsymbol{e}^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_k} = \begin{cases} \boldsymbol{I}_3 & (k=i) \\ -\boldsymbol{I}_3 & (k=j) \\ \boldsymbol{0}_{3\times 3} & (\text{otherwise}) \end{cases}.$$

Note that we can omit the calculation for constraints unrelated to k.

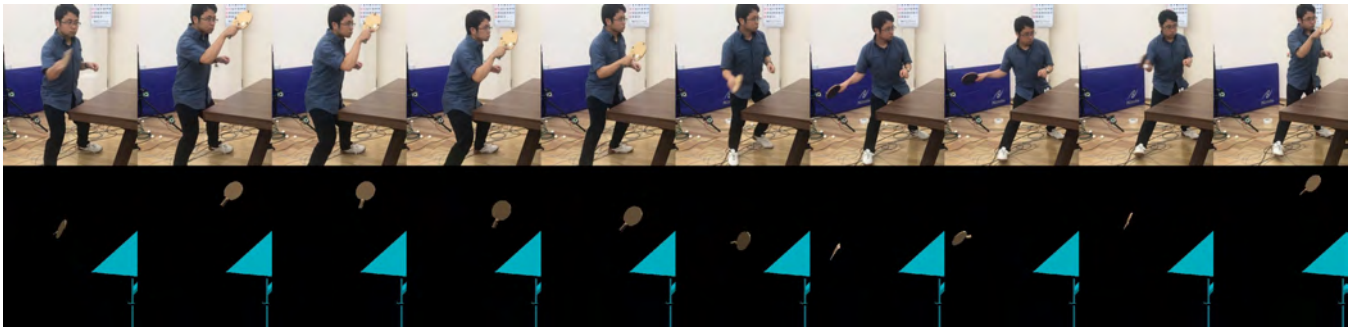


Fig. 10. Visualization of the estimated racket position and orientation. The estimated motion matched well with the images from the smartphone camera.

By calculating the second derivative, we obtain

$$\frac{\partial^2 F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial^2 \boldsymbol{v}_i} = \frac{\partial^2 F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial^2 \boldsymbol{v}_j} = \boldsymbol{\Omega}^{(\mathrm{v})}_{i,j}, \tag{7}$$

$$\frac{\partial^2 F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_j} = \frac{\partial^2 F^{(\mathrm{v})}_{i,j}(\boldsymbol{V})}{\partial \boldsymbol{v}_j \partial \boldsymbol{v}_i} = -\boldsymbol{\Omega}^{(\mathrm{v})}_{i,j}. \tag{8}$$

The Hessian matrix can be constructed by adding the above values to the corresponding blocks. Note that the Hessian matrix should be constructed in a sparse matrix form so that computationally efficient solver, such as sparse Cholesky [25], can be applied for solving Eq. (2).

REFERENCES

[1] N. Ahmad, R. A. Raja Ghazilla, N. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *International Journal of Signal Processing Systems*, vol. 1, pp. 256–262, 01 2013.

[2] K. King, S. Yoon, N. Perkins, and K. Najafi, "Wireless MEMS inertial sensor system for golf swing dynamics," *Sensors and Actuators A: Physical*, vol. 141, no. 2, pp. 619–630, 2008.

[3] R. McGinnis and N. C Perkins, "A highly miniaturized, wireless inertial measurement unit for characterizing the dynamics of pitched baseballs and softballs," *Sensors*, vol. 12, pp. 11 933–11 945, 2012.

[4] K. Sagawa, S. Abo, T. Tsukamoto, and I. Kondo, "Forearm trajectory measurement during pitching motion using an elbow-mounted sensor," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 3, no. 4, pp. 299–311, 2009.

[5] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Technical Report, 2007.

[6] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position andorientation estimation," *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, 2017.

[7] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38–46, 2005.

[8] L. Ojeda and J. Borenstein, "Non-GPS navigation for security personnel and first responders," *Journal of Navigation*, vol. 60, no. 3, pp. 391–407, 2007.

[9] G. Pons-Moll, A. Baak, T. Helten, M. Mller, H. Seidel, and B. Rosenhahn, "Multisensor-fusion for 3d full-body human motion capture," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 663–670.

[12] K. Zhang, Z. Fang, J. Liu, Z. Wu, and M. Tan, "Fusion of vision and imu to track the racket trajectory in real time," in *IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017, pp. 1769–1774.

[10] T. Helten, M. Mller, H. Seidel, and C. Theobalt, "Real-time body tracking with one depth camera and inertial sensors," in *International Conference on Computer Vision*, 2013, pp. 1105–1112.

[11] Z. Zheng, T. Yu, H. Li, K. Guo, Q. Dai, L. Fang, and Y. Liu, "Hybridfusion: Real-time performance capture using a single depth sensor and sparse IMUs," in *European Conference on Computer Vision (ECCV)*, 2018.

[13] K. Irie, "A graph optimization approach for motion estimation using inertial measurement unit data," *ROBOMECH Journal*, vol. 5, no. 1, p. 14, 2018.

[14] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[15] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[16] K. Irie, "Error correction of inertial motion estimation through three-step loop closures," in *Proceedings of Robotics Symposia*, 2019, pp. 205–212, (in Japanese).

[17] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," Technical Report, Stanford University, 2006.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[19] M. Sugiyama, "Machine learning with squared-loss mutual information," *Entropy*, vol. 15, no. 1, pp. 80–112, 2013.

[20] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinformatics*, vol. 10, no. 1, pp. 1–12, 2009.

[21] T. Sakai and M. Sugiyama, "Computationally efficient estimation of squared-loss mutual information with multiplicative kernel models," *IEICE Trans. on Information and Systems*, vol. E97-D, no. 4, pp. 968–971, 2014.

[22] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *IEEE International Conference on Rehabilitation Robotics*, 2011, pp. 1–7.

[23] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Conference on Neural Information Processing Systems (NIPS)*, 2014.

[24] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6602–6611.

[25] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate," *ACM Trans. Math. Softw.*, vol. 35, no. 3, pp. 22:1–22:14, 2008.