2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Synchronization of external inertial sensors and built-in camera on mobile devices

Filip Malawski, Ksawery Kapela, Marek Krupa
Institute of Computer Science
AGH University of Science and Technology
Krakow, Poland
fmal@agh.edu.pl

***Abstract*—The fusion of inertial and visual data is an effective approach to human motion analysis, with applications in areas such as sports or rehabilitation exercise monitoring. Employing wireless, low-cost, external inertial sensors and a built-in camera on mobile devices provides a convenient acquisition system, available for wide range of potential users. In order to take advantage of both data modalities, robust time synchronization is required. We consider consumer-grade devices, for which direct access to internal clocks is not available and only high-level API is provided. At the same time, we aim to avoid event-based synchronization that would require additional user actions. We investigate sources of acquisition errors on mobile devices, and then we propose and evaluate a novel synchronization method for inertial and visual data. Experimental results indicate that the proposed method provides robust synchronization.**

***Index Terms*—IMU, video, synchronization, mobile, multi-modal**

I. Introduction

Multi-modal human motion analysis is a useful tool for assisting sports training, recognition of daily activities, immersive gaming, medical diagnosis, and facilitating rehabilitation [1], [2]. Fusion of data from cameras and inertial measurement units (IMU) is often very beneficial for analyzing different aspects of human motion [3]. Sophisticated setups with advanced measurement systems provide accurate pose estimation [4], however, their use is limited to laboratory settings or professional applications. On the other hand, low-cost IMUs have become popular [5], while at the same time, the latest developments in deep learning allow obtaining accurate pose estimation from RGB videos using built-in cameras on mobile devices [6]. Combining low-cost IMUs and mobile cameras has the potential for making multi-modal motion tracking available for a wide range of users. One of the crucial problems in such a scenario is providing a reliable synchronization procedure in order to effectively combine information from both modalities. Professional motion capture systems use dedicated synchronization protocols, however an acquisition setup including multiple consumer-grade, multi-vendor devices is limited in this regard by available APIs, which rarely include low-level access required for precise clock synchronization. Event-based synchronization is sometimes used for combining data from different devices [7], however, it requires additional actions from users, which is neither convenient nor very reliable without supervision.

In this work, we consider a setup consisting of a single mobile device, and up to five external IMUs, connected wirelessly. The mobile device records both video data from a built-in camera, and inertial data streamed wirelessly from the IMU sensors. The setup is to be used for analysing sports or rehabilitation exercises performed either in sports classes or at home. Multi-modal signal often provides more complete information regarding the performed motion, e.g. in fencing [8]. Our goal is to obtain a precise, fully automatic synchronization, without requiring additional actions from users. We propose methods for effective data synchronization, as well as procedures for the evaluation of different aspects of both acquisition and synchronization.

II. Related work

Methods proposed for synchronization of signals from multiple devices vary significantly depending on specific usage scenarios and constraints that those imply [3]. In general, data synchronization is either based on internal clocks or on detecting common events in each acquired signal. Low-level synchronization based on internal device clocks and dedicated protocols is very effective [9], [10], however, it requires low-level access to the device and communication layer. Therefore such an approach is mostly used in custom prototype solutions [11] or in professional motion capture systems such as Vicon[1].

When using low-cost consumer-grade sensors low-level access is usually not available, hence event-based synchronization is often employed. Methods proposed in the literature vary greatly depending on acquired data modalities. Matching bright flashes in videos is used to synchronize multiple cameras [12]. Yang et al. propose a method for synchronization of a global navigation satellite system with IMU mounted on skiers based on acceleration obtained from both sources [13]. Events in acceleration signals are also used to synchronize wearable motion capture and EMG measurement systems [7]. Wang et al. propose synchronization of EMG, EEG, and IMU signals using an additional force sensor [14]. Another approach

The research presented in this paper was supported by the National Centre for Research and Development (NCBiR) under Grant No. LIDER/37/0198/L-12/20/NCBR/2021.

[1]https://www.vicon.com

to synchronization is to take advantage of corresponding events from IMUs and pressure sensors [15].

Using mobile devices for multi-modal data acquisition has practical value, however, it introduces additional problems related to the mobile operating system, in particular delays in delivering data events. Feng et al. propose a method for synchronizing inertial and video signals from built-in IMU and camera on a single mobile device [16]. Evaluation of synchronization is yet another issue and often requires additional devices. Controlled light flashes can be used for verifying visual sensor synchronization [17]. In another study, authors employ a dedicated rotating device to evaluate angular velocity measured by both camera and IMU [18].

In contrast to the previous works, we consider a scenario in which multiple low-cost external IMUs are connected wirelessly to a mobile device, while video data from a built-in camera are recorded simultaneously. As a design choice, we avoid using event-based synchronization (except for the evaluation of the proposed method). To the best of our knowledge, for such setup, no results for synchronization procedures have been presented in the literature so far.

III. Methods

Our setup consists of up to five external IMUs connected via Bluetooth 5.x to an Android smartphone with a built-in camera (see Fig. 1). The limitation on the number of sensors is due to the Bluetooth protocol version. Our goal is to provide per-frame synchronization of inertial and video data. The approach proposed in this work is based on several assumptions regarding employed devices. First of all, we assume that IMUs are synchronized with each other using a built-in procedure provided by their manufacturer, as this is usually the case for low-cost consumer-grade sensors such as Xsens DOT[2] or Mbientlab MetaMotion[3]. Therefore, we need only to consider the synchronization of data from a single IMU with video data. That being said, it is worth noting, that the proposed method could be adapted for synchronizing multiple sensors as well, as it does not depend on the type of the signal. Secondly, we assume that IMUs internally acquire data in proper time intervals and that they provide correct internal timestamps as well as a packet counter, which can be used to detect missing packets. Finally, we assume, that the IMUs and the built-in camera acquire data with the same sampling rate (60Hz was used in our experiments).

Data from both modalities are recorded on the same mobile device, hence a common clock for data events is available, however only through high-level API. The clock indicates only when a data event is received, not when the data was actually acquired. We anticipate several sources of acquisition and synchronization problems when recording inertial and video data on a single mobile device:

- Bluetooth connection - some data packets may be delivered with delay due to wireless communication, moreover some data packets may be missing.

[2]https://www.movella.com/products/wearables/movella-dot
[3]https://mbientlab.com/metamotions

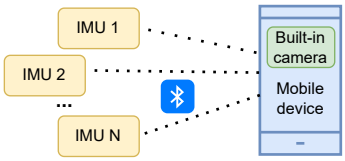

Fig. 1. Multi-modal data acquisition system.

- Built-in camera - there is no guarantee that the mobile camera captures frames in equal time intervals.
- Android operating system - the delay between receiving data and generating an event in Android API may be significant. Moreover, it may vary, depending on the available resources.
- Sampling rate - while we employ devices with the same nominal sampling rate (60Hz) actual sampling rate of each device might be slightly higher or lower, which in longer acquisitions may impact the synchronization.

In order to analyze the impact of the potential sources of synchronization problems we design specific evaluation procedures. Then, we propose and evaluate methods for multi-modal data synchronization for our setup.

A. Sampling stability evaluation

First, we measure the stability of acquiring video data with the built-in mobile camera. We record a view of a stopwatch running on another mobile device equipped with a 120Hz display (twice as fast as the camera acquisition rate). We manually label each frame with a stopwatch value visible in that frame (see Fig. 2). We perform this procedure using both standard camera application pre-installed on the phone as well as with our custom application developed with Android API. While the standard camera application is used as a baseline for this evaluation, a custom application was necessary to obtain the functionality of acquiring timestamps for each video frame, which in turn is needed for the final synchronization with the IMU signal.

We use differences in stopwatch values seen on the video recording to measure actual time intervals between consecutive frames. We compute statistical measures (mean, standard deviation, median, minimum, maximum) to evaluate the stability of video data acquisition. Secondly, we employ, in a similar manner, differences in timestamps recorded with the custom video recording application to measure the stability of receiving video data events. Please note, that the timestamps correspond to Android API data events rather than actual acquisition time, which may be different.

In the case of inertial sensors, as mentioned before, we use only a single IMU, as synchronization between IMUs is handled by their internal software. Data from IMU is recorded on the mobile device using a modified application from the manufacturer - we added the functionality of recording Android system timestamps. Similarly, as in the video recording application, the timestamps correspond to Android data events rather than actual acquisition and therefore may

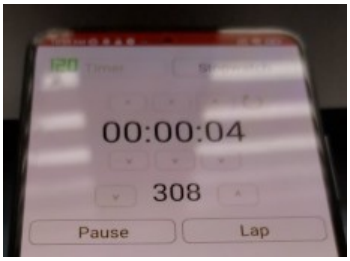

Fig. 2. Stopwatch recorded with the mobile device camera.

differ from internal timestamps provided by the IMU device. Internal timestamps have constant time differences, however, their starting point is arbitrary (resulting from starting the sensor), hence they can't be directly compared with video timestamps. We analyze variability in Android timestamps, which may be caused both by communication delays, as well as by the mobile operating system. It is worth noting, that data received from sensors is preprocessed - we identify missing packets using the packet counter field provided in sensor data and recreate them using linear interpolation (including timestamps).

B. Multi-modal synchronization

Since both video and inertial data are recorded on the same device, ideally, we would match recorded timestamps to find corresponding frames in both modalities. However, Android timestamps do not represent actual acquisition time and their variability is significant. Therefore, we use matching video and inertial data frames by closest timestamps only as a baseline method. To reduce high standard deviation in timestamps, we perform, as an optional step in all methods, moving average filtering of the timestamps. Experimentally we choose window size = 9, as it significantly reduces standard deviation, while it requires a relatively small time context.

Considering that actual acquisition is much more stable than Android timestamps, we can select a synchronization frame and assume constant sampling intervals for other frames. However, such an approach is prone to errors when we select a frame with poorly corresponding timestamps and there is no information available to support selecting any specific frame. Instead, we employ statistical information extracted from multiple frames. In our experiments with single frame-based synchronization, we observed that 'good' synchronization frames occur most often. Therefore, we can compute multiple time offsets between inertial and video data separately using each frame as a synchronization frame and then select the median offset value. However, this approach is not robust to time drift between modalities, which may occur due to slightly different sampling rates. Therefore, our final synchronization method employs windowed median filtering of time offsets. Windowed median offset is computed for each frame separately. The length of the filter window is selected experimentally. Due to employing time windows our proposed method is adaptive during acquisition.

TABLE I
EXAMPLE OF FINDING CORRECT OFFSET BETWEEN FRAMES OF DIFFERENT MODALITIES WITH THE PROPOSED METHOD. CONSIDERED FRAME, TIME WINDOW, AND FINAL COMPUTED OFFSET ARE MARKED IN YELLOW.

IMU		**Camera**				
Data frame index	**Time-stamp**	**Data frame index**	**Time-stamp**	**Closest IMU frame**	**Offset**	**Median offset in window**
...	...	...	...	...	...	...
32	1000	5	1001	32	27	...
33	1015	6	1020	33	27	...
34	1031	7	1036	35	28	...
35	1039	8	1070	37	29	...
36	1061	9	1079	37	28	27
37	1077	10	1087	37	27	...
38	1099	11	1110	38	27	...
39	1131	12	1130	39	27	...
40	1148	13	1142	40	27	...
...	...	...	...	...	...	...

Table I presents an example of matching frames from video data to IMU data with the proposed method. The example considers a single, arbitrary selected frame, however the procedure is the same for each frame. In order to find matching IMU frame for video data frame with index = 9, we first find the closest IMU frames for several frames in the selected time window, using recorded Android timestamps. Then, we compute offsets between indices of IMU and video frames. Finally, for selecting the corresponding IMU frame, we use median offset in the time window (27), instead of baseline offset (28) computed using single frame. By adding the computed offset (27) to the video data frame index (9) we select IMU frame with index = 36. For brevity, in the example, we use window size = 7, however, actual window size is much larger (see Section IV-B).

C. Synchronization evaluation

In order to evaluate the proposed method, we employ event-based synchronization to obtain ground truth for test recordings. The evaluation procedure is as follows. We put the IMU on a flat surface and then every few seconds we apply a small force for a short time (a single push). This is recorded by the built-in mobile camera. Each push event is easily identifiable in both the IMU signal (acceleration peak) and video signal (sensor movement). We manually label push events in signals from both modalities, therefore obtaining ground truth synchronization for selected frames.

IV. EXPERIMENTS

For experiments, we employ Xsens DOT sensors and Samsung Galaxy A52s Android mobile device. For video stability acquisition evaluation we also use the Xiaomi POCO F3 smartphone as a second device. Five IMUs are used in experiments regarding resource usage on mobile device. One IMU is used in synchronization experiments, as all IMUs are synchronized with each other by their internal software.

TABLE II
STATISTICAL MEASURES OF TIME INTERVALS IN VIDEO DATA AND VIDEO TIMESTAMPS (MILLISECONDS)

Device	Source	Mean	Median	SD	Min	Max
Samsung Galaxy	Standard app.	16.71	17	2.01	6	33
	Custom app	16.80	17	2.02	13	33
	Timestamps	16.69	16	4.74	5	38
Xiaomi POCO	Standard app.	16.64	17	1.18	14	27
	Custom app.	22.09	18	8.28	14	50
	Timestamps	22.05	22	2.60	16	31

TABLE III
STATISTICAL MEASURES OF TIME INTERVALS IN ANDROID TIMESTAMPS FOR INERTIAL DATA (MILLISECONDS)

IMUs	Active app.	Mean	Median	SD	Min	Max
1	IMU record.	16.66	26.00	14.83	0	92
	Video preview	16.67	26.00	14.59	0	93
	Video record.	16.66	24.00	14.05	0	91
5	IMU record.	16.67	28.00	15.02	0	89
	Video preview	16.67	26.00	14.02	0	89
	Video record.	16.66	26.00	14.19	0	88

A. Sampling stability evaluation

The first experiment considers the evaluation of the stability of video acquisition and the stability of recording Android timestamps for video frames. For each test case, four-second video (approx. 240 frames) of running stopwatch was recorded and manually labeled with stopwatch values. Table II presents a statistical evaluation of time intervals measured for video acquisition, using both standard camera application and custom video recording application, as well as for timestamps recorded with Android API. In the case of Samsung Galaxy, we can observe that the mean time interval between frames (16.71 ms) is close to expected (16.67 ms due to 60Hz sampling). However, standard deviation (SD = 2.01 ms), minimum value (6 ms), and maximum value (33 ms) indicate that acquisition is not always stable. Both recording apps have similar stability, however, timestamps are much more unstable than actual acquisition (SD 4.74 ms vs. 2.01 ms and 2.02 ms), due to the Android event system. In the case of Xiaomi POCO stability of video data acquired with the custom application is much worse than that of the standard application (SD 8.28 ms vs. 1.18 ms) which indicates that the manufacturer's camera application probably uses custom hardware functions not available in Android API. Xiaomi POCO is not used in further experiments. Fig. 3 depicts time interval variability for Samsung Galaxy. Custom application and timestamps plots are for the same recording, however, we can observe that changes in time intervals are not aligned, which indicates that delays introduced by the Android event system are independent of actual camera acquisition. Filtered timestamps, obtained with a moving average with window size = 9, have much less variability (SD = 0.75 ms).

In the case of IMUs, we evaluate Android timestamps stability with regard to the number of active sensors (1 or 5) and which mobile application is in foreground (IMU

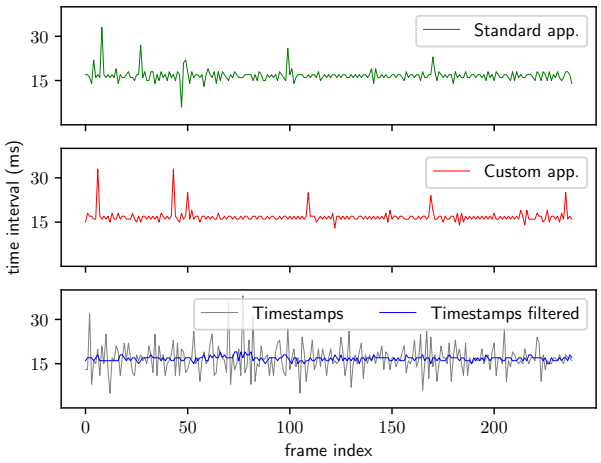

Fig. 3. Measured time intervals for video frames and timestamps (Samsung Galaxy).

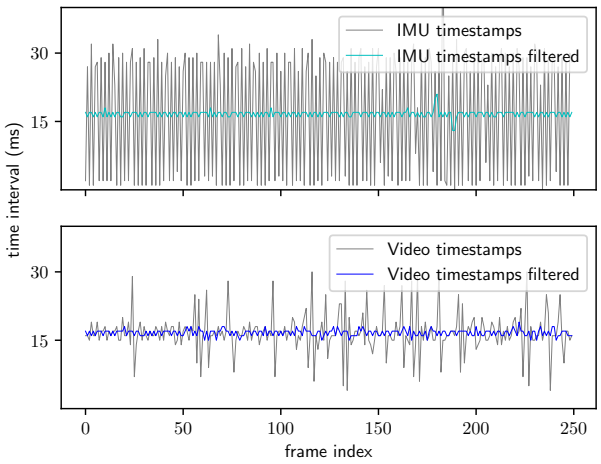

Fig. 4. Time intervals based on Android timestamps acquired for inertial and video data (both plots represent the same recording).

recording application, camera application in preview mode, camera application in recording mode) in order to study the impact of resources usage. Results are presented in Table III. We can observe that using either 1 or 5 sensors has no impact on stability. Interestingly, timestamps are slightly more stable when the IMU recording application is in the background rather than the foreground. Simultaneous video recording has no negative impact. In Fig. 4, in the upper plot, we can observe very high variability in Android timestamps for inertial data, which is in turn greatly reduced when a moving average filter is applied (SD = 0.69 ms). In the lower plot, we can see that the variability of corresponding video timestamps for the same recording is independent of IMU timestamp deviations.

B. Synchronization evaluation

Proposed synchronization methods were evaluated using the protocol described in section III-C. Four test sequences were acquired, each approx. 30 seconds long, with 15 push events

TABLE IV
EVALUATION RESULTS FOR SYNCHRONIZATION METHODS, VALUES PRESENTED IN FRAMES (T. FILT. DENOTES TIMESTAMP FILTERING WITH MOVING AVERAGE)

Method	T. filt.	Mean	Median	SD	Min	Max
Baseline	No	0.81	1	2.20	-14	3
	Yes	0.71	1	2.16	-14	3
Single sync. frame	No	-0.16	0	0.74	-1	3
	Yes	-0.16	0	0.74	-1	3
Median offset - all frames	No	0.10	0	0.71	-1	3
	Yes	0.10	0	0.71	-1	3
Median offset - time windows	No	0.05	0	0.66	-1	3
	Yes	0.03	0	0.65	-1	3

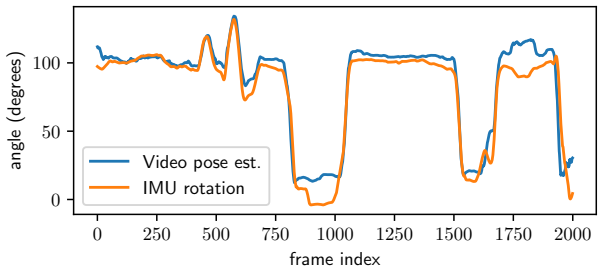

Fig. 5. Shoulder angle estimation in yoga exercise based on pose estimation in video data and rotation provided by IMU mounted on the person. Signals are synchronized with the proposed method.

per sequence, manually labeled in video and inertial data. For the windowed median synchronization, window size = 500 was selected experimentally. Results in Table IV contain aggregated results from all sequences. We can observe that the baseline method (finding closest timestamps) performs poorly, due to high variability in Android timestamps. Employing single synchronization frame yields better results, however, it depends on being lucky with selecting the frame. The median of time offsets computed for all frames provides a further improvement, however, this approach does not take into account possible time drift between data from two modalities. Finally, the median time offset computed in time windows yields the best results, particularly when used jointly with moving average filtering of timestamps. One additional experiment was performed as a proof-of-concept for a practical application of the proposed methods. Fig. 5 depicts shoulder angle during yoga exercise, measured with automatic pose estimation from video (using BlazePose model [19]) as well as with rotation provided by sensor fusion algorithm built in Xsens DOT device. We can observe that the proposed synchronization method enables multi-modal motion analysis.

V. CONCLUSIONS

We identified and evaluated several sources of problems in the synchronization of external IMUs and built-in camera on a mobile device. Based on our experiments, we conclude that the most problematic aspect is the unstable delivery of data events from the mobile operating system. Timestamps recorded with data events have significant variability and do not correspond to actual data acquisition times, which makes synchronization difficult. However, we were able to employ a statistical approach to obtain effective synchronization by using information from multiple frames. Our experiments confirmed, that the proposed method provides good matching of IMU and video frames. Also, it does not require any actions from the user, as is the case with event-based synchronization methods, which makes it convenient to use in practical applications.

It is also worth noting, that while in our experiments we specifically used IMUs and a camera, the proposed method can be applied to any modalities, as it depends only on the recorded timestamps, and not on the data itself. The limitations include sampling rate and missing data frames. The acquisition from all devices should be performed with the same sampling rate or the signals need to be interpolated to a common sampling rate. The signal has to include all frames, otherwise it must be possible to identify missing frames, usually by using provided frame numbers. For example, the proposed method could be used to synchronize multiple IMUs if devices without internal synchronization were used.

REFERENCES

[1] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, "Human action recognition from various data modalities: A review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2022.

[2] Y. Li, C. Wang, Y. Cao, B. Liu, J. Tan, and Y. Luo, "Human pose estimation based in-home lower body rehabilitation system," in *2020 Int. Joint Conf. on Neural Networks (IJCNN)*, pp. 1–8, Ieee, 2020.

[3] S. Majumder and N. Kehtarnavaz, "Vision and inertial sensing fusion for human action recognition: A review," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 2454–2467, 2020.

[4] Z. Zhang, C. Wang, W. Qin, and W. Zeng, "Fusing wearable imus with multi-view images for human pose estimation: A geometric approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2200–2209, 2020.

[5] P. Picerno, M. Iosa, C. D'Souza, M. G. Benedetti, S. Paolucci, and G. Morone, "Wearable inertial sensors for human movement analysis: A five-year update," *Expert review of medical devices*, vol. 18, no. sup1, pp. 79–94, 2021.

[6] F. Malawski and B. Jankowski, "Depth-based vs. color-based pose estimation in human action recognition," in *Advances in Visual Computing: 17th Int. Symposium, ISVC 2022, San Diego, CA, USA, October 3–5, 2022, Proceedings, Part I*, pp. 336–346, Springer, 2022.

[7] R. V. Schulte, E. C. Prinsen, L. Schaake, and J. H. Buurke, "Synchronization of wearable motion capture and emg measurement systems," in *2022 Int. Conf. on Rehab. Robotics (ICORR)*, pp. 1–6, IEEE, 2022.

[8] F. Malawski and B. Kwolek, "Recognition of action dynamics in fencing using multimodal cues," *Image and Vision Computing*, vol. 75, pp. 1–10, 2018.

[9] H. Lee, W. Yu, and Y. Kwon, "Efficient rbs in sensor networks," in *Third Int. Conf. on Information Technology: New Generations (ITNG'06)*, pp. 279–284, IEEE, 2006.

[10] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd Int. Conf. on Embedded networked sensor systems*, pp. 39–49, 2004.

[11] T. Steinmetzer, S. Wilberg, I. Bönninger, and C. M. Travieso, "Analyzing gait symmetry with automatically synchronized wearable sensors in daily life," *Microprocessors and Microsystems*, vol. 77, p. 103118, 2020.

[12] P. Shrestha, H. Weda, M. Barbieri, and D. Sekulovski, "Synchronization of multiple video recordings based on still camera flashes," in *Proceedings of the 14th ACM Int. Conf. on Multimedia*, pp. 137–140, 2006.

[13] Y. Yang, R. Cheng, J. He, C. Li, X. Qiao, X. Hou, J. Xiong, and X. Chou, "Time synchronization algorithm for the skiing monitoring system," *IEEE Trans. on Instrum. and Meas.*, vol. 71, pp. 1–9, 2022.

[14] C. Wang, H. Zhang, S. H. Ng, X. Zhu, and K. K. Ang, "Wireless multi-sensor physio-motion measurement and synchronization system and method for hri research," in *2021 43rd Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 7328–7331, IEEE, 2021.

[15] S. Shabani, A. K. Bourke, A. Muaremi, J. Praestgaard, K. O'Keeffe, R. Argent, M. Brom, C. Scotti, B. Caulfield, and L. C. Walsh, "An automatic foot and shank imu synchronization algorithm: Proof-of-concept," in *2022 44th Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4210–4213, IEEE, 2022.

[16] Z. Feng, J. Li, and T. Dai, "Sensor synchronization for android phone tightly-coupled visual-inertial slam," in *China Satellite Navigation Conf. (CSNC) 2018 Proceedings: Volume III*, pp. 601–612, Springer, 2018.

[17] M. Faizullin, A. Kornilova, A. Akhmetyanov, K. Pakulev, A. Sadkov, and G. Ferrer, "Smartdepthsync: Open source synchronized video recording system of smartphone rgb and depth camera range image frames with sub-millisecond precision," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 7043–7052, 2022.

[18] S. Yuan, Y. Li, Y. Chen, and C. Wang, "Time synchronization accuracy verification for multi-sensor system," in *2021 Int. Conf. on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, pp. 21–24, IEEE, 2021.

[19] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.