PresentPostures: A Wrist and Body Capture Approach for Augmenting Presentations

Jochen Kempfle
Ubiquitous Computing
University of Siegen
Siegen, Germany
jochen.kempfle@uni-siegen.de

Kristof van Laerhoven
Ubiquitous Computing
University of Siegen
Siegen, Germany
kvl@eti.uni-siegen.de

***Abstract*—Capturing and digitizing all nuances during presentations is notoriously difficult. At best, digital slides tend to be combined with audio, while video footage of the presenter's body language often turns out to be either too sensitive, occluded, or hard to achieve for common lighting conditions. If presentations require capturing what is written on the whiteboard, more expensive setups are usually needed. In this paper, we present an approach that complements the data from a wrist-worn inertial sensor with depth camera footage, to obtain an accurate posture representation of the presenter. A wearable inertial measurement unit complements the depth footage by providing more accurate arm rotations and wrist postures when the depth images are occluded, whereas the depth images provide an accurate full-body posture for indoor environments. In an experiment with 10 volunteers, we show that posture estimates from depth images and inertial sensors complement each other well, resulting in far less occlusions and tracking of the wrist with an accuracy that supports capturing sketches.**

***Index Terms*—motion capture, inertial measurement, kinect**

I. Introduction

Tracking a person's wrist's position and orientation is a key feature in many applications such as virtual reality, medical applications, computer games, or manual task analysis [1]. In this paper, we present a novel approach that combines a wrist-worn inertial measurement unit (IMU) with depth images of the entire person, to robustly track the human posture in real time, for capturing a presenter's *body language* and *writing*. We argue that the dominant wrist needs to be tracked very accurate for this purpose, and that the two modalities combined will lead to a more accurate system that can cope with common problems that the individual sensors suffer from, in particular occlusions and inertial sensor drift. To this end, we focus here on a study that measures how accurate depth imaging and inertial sensing can track the hand's position while writing on a whiteboard. The contributions of this paper are threefold:

- A software framework is presented that allows, in real-time, to acquire and combine the measurements of body-worn inertial data and depth images.
- We present custom methods to calibrate and synchronize smartwatch data with the depth data for a body model.
- A study evaluates the tracking performance of both body and wrist for the special case of writing on a whiteboard.

In the following, we highlight our approach with relation to related research, before presenting the study and its results.

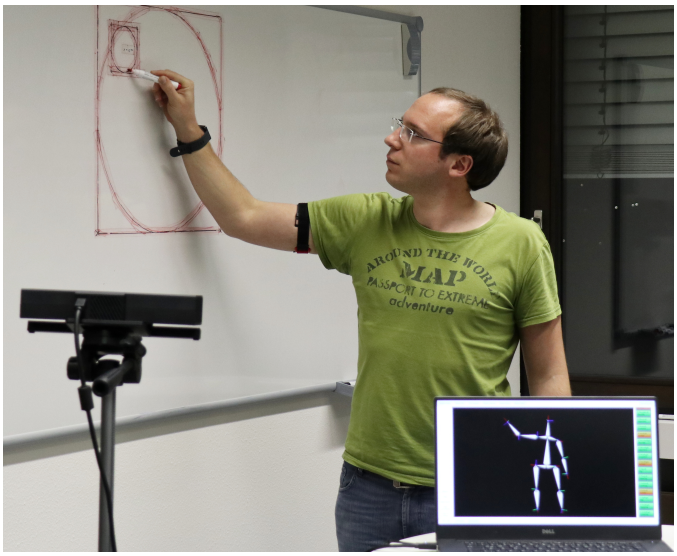

Fig. 1. Our approach combines a depth camera data with wrist-worn 9D inertial (IMU) readings in real-time, to robustly capture a presenter's postures.

II. Related Work

IMU-based posture estimation successfully is applied in many applications and IMU-based full body tracking systems already are deployed industrially [2]. Integrating the IMU sensor data into a biomechanical model and modelling the sensor to bone offset, such as in [3] or [4], increases the overall accuracy [2]. Accessing the various calibration parameters therefore is a vital requirement. For camera-based systems, extensive frameworks exist for so-called RGB-D sensors that use depth information, such as [5], and for highly accurate commercial systems that rely on fiducial markers. Vision-based motion capture systems are known to have their specific weaknesses as well. Self-occlusion by the person under observation and occlusion by nearby structures, as well as adverse lighting conditions tend to hamper an accurate body posture recognition [6]. Additionally, these systems tend to be less flexible to be moved at different locations, and their setup effort and costs tends to be higher than wearable inertial measurement solutions. In recent years, some examples have shown how these weaknesses in one modality can be addressed by another. In [7], for instance, cameras in the environment

are used to improve on wireless localization units worn by humans. With the Kinect, [8] has shown how a combination with wearable inertial measurement units improves motion capture by tracking the body's position and the initial calibration pose with the Kinect and the limb movements with IMUs.

Utilizing RGB-D sensors such as the Microsoft Kinect for estimating the sensor to bone offset leads to superior motion capture results compared to estimating them by hand, relying on the correct sensor placement, or executing specific calibration movements [9]. Furthermore, determining each IMU's sensor drift using standard system identification methods with the respective Kinect's joint data output increases the overall long-time accuracy even when the captured body no longer is tracked by the Kinect [10]. On the other hand, [11] and [8] show how IMU-based tracking can improve the Kinect data, especially under occlusion. Other works, such as [12] and [13], show that IMU-based systems can be cost-effective and dynamically deployable, yet face calibration and 'floating' artifacts for hip-joint rooted methods. Indoor magnetic disturbances are also known to affect the IMU-based units' accuracy, leading to a variety of research efforts to characterize and compensate for this (as for example summarized in [14] in a recent survey and collection of methods).

For the capturing of presentation-related gestures, RGB-D sensors have been successfully used to control a PC with gestures. This includes taking control over a power point application and navigating the slides with certain gestures with the arm movement. In many presentations, for example a lecture at university, the presenter often writes something on the whiteboard. In such a case, the RGB-D sensor not only has to deal with occlusion, but also with some specific weaknesses such as being unable to correctly track a human from the side or from the back (see [15] and [16] for surveys on the Kinect abilities compared to a gold standard Vicon 3D motion capture system). Thus, even though gestures and body postures facing the audience can be captured relatively well by a single RGB-D sensor, performance tends to deteriorate as soon as the presenter faces the blackboard.

This paper presents an approach that allows capturing the presenter's body postures and sketches on the blackboard or whiteboard throughout a presentation, by complementing the data from a depth sensor with wrist-worn IMU readings. This type of tracking creates less overhead than an additional video recording of the presenter, which often includes the need to adjust the camera's direction and zoom level, and tends to work less well in darker presentation environments.

III. Approach Overview

Sensors and Data Acquisition. The most prominent sensor types for motion capturing are marker-based or marker-less optical motion capture systems and systems based on body-worn sensors that are able to estimate the orientation of the limb they are attached to. Optical systems are very precise in both tracking the position and orientation of a body and its movable parts, but typically suffer from occlusion and limited working space. In many cases, such systems also are expensive and have low mobility. The body-worn devices on the other hand do not know occlusion, are not limited to a certain workspace, have high mobility, and are typically less costly. Their main disadvantages are that they suffer from precision and sensor drift, and often are susceptible to magnetic disturbances. Interestingly, the optical and the sensor-based approaches cover each others' disadvantages well. In this paper, therefore, the combination of both types is applied to motion capturing, utilizing an optical RGB-D sensor and body-worn Inertial Measurement Units (IMUs). To obtain a limb's orientation from RGB-D data, the data first has to be processed for example by using [17]. An IMU either directly provides an orientation quaternion or its acceleration, gyroscope, and magnetometer sensor readings have to be fused by well-known filtering algorithms such as [18].

The RGB-D data is captured by the Microsoft Kinect v2, which has the advantage of being an optical motion capture system that is cheap and mobile. Furthermore, it already provides precomputed joint positions and orientations that are accessible through the software interface of the Kinect for Windows SDK 2.0. Although the wrist joint orientations typically are not trustworthy, as due to ambiguity the actual forearm orientation around its direction axis cannot be determined easily, these already precomputed joint orientation data is forwarded to the motion capturing process. The Kinect receiver implementation simulates for each Kinect joint an own virtual IMU sensor. Distributing the Kinect data to single simulated sensors serves the purposes (1) that it can be mapped to arbitrary skeletons with different bone setup, (2) that the data is treated like other sensor data and existing motion capture routines can be reused, and (3) that it can be interchanged, compared or fused with ordinary IMU sensor data.

The body-worn sensors are represented by one or multiple smartwatches. They contain an IMU for sensing their orientation and also have the necessary communication interfaces to send the data immediately to a connected PC. In our current setup, the smartwatches relay their data via Bluetooth to a nearby smartphone, which in turn forwards all data directly to the PC. The smartwatches, in contrast to the Kinect, do not come with a data acquisition tool for motion capture. Data acquisition is performed with a custom App that lets the Android operation system estimate the orientation of the device and sends the data first via Bluetooth to the smartphone and then via a UDP broadcast to the PC. The advantage of a UDP broadcast is that it allows the connected PC to receive data from multiple devices simultaneously on the same network address. However, this comes with a cost: UDP broadcast is not a reliable connection, meaning that single data packets can get lost, interchanged, or be arbitrarily delayed.

Sensor Model and Bone Mapping. The smart-watches in use will send their orientation with respect to their earth global coordinate system - for example the attitude heading reference system (AHRS) - as a unit quaternion. This reference frame has to be mapped to OpenGL screen coordinates that define the global reference frame for motion capturing. Here,

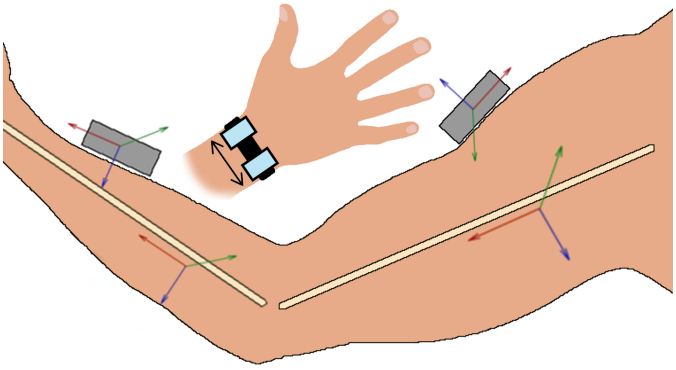

Fig. 2. The sensor to bone offset challenge: Each sensor can arbitrarily be placed on and around the arm, facing in any direction. The body surface is not flat and the sensor orientation differs from the bone orientation.

instead of the z-axis, the y-axis points up. In screen coordinates furthermore the x-axis points to the right and the z-axis to the front of the screen. Each sensor takes its measurements in its own local coordinate system, which it assumes to be global. This means that two different sensors may face the same direction, but do not necessarily output the same orientation quaternion. This leads to a visible offset of both sensors in the screen reference frame. Furthermore, the screen coordinate system with respect to the earth global coordinate system can be rotated arbitrarily. So even if all sensors have the same notion of the global coordinate system of the earth, it is not desired to use it for motion capturing, as the captured object always would have to move in the same compass direction in order to obtain the same movement of the skeleton. For each sensor, therefore its offset to the screen coordinate system has to be modeled. This rotational offset is called the coordinate offset.

A sensor can arbitrarily be attached on the body's surface. The body surface neither is flat nor is a sensor always placed on the same location or in the same direction as shown in Figure 2. So simply taking a sensor's orientation estimate as bone orientation is not sufficient. To compute a bone orientation correctly, the sensor's rotational offset with respect to the bone has to be modeled for each sensor. This offset is called the bone offset.

Both rotation offsets are represented by a unit quaternion, describing how the sensor has to be rotated to match a bone's local coordinate system or the screen coordinate system respectively. Their correct acquisition is part of the calibration and is described in detail in the next chapter. To map a sensor orientation q_{sensor} to a correct global bone orientation $q_{bone,global}$ in screen coordinates, both calibration values, the coordinate offset $q_{coordinate\,offset}$ and the bone offset $q_{bone\,offset}$ are required. The bone orientation $q_{bone,global}$ then can be computed with (1).

$$q_{bone,global} = q_{coordinate\,offset} \cdot q_{sensor} \cdot q_{bone\,offset} \quad (1)$$

In this case, it is assumed that the sensor data is perfect and the rotational bone offset stays constant over time. In reality however, these assumptions are not necessarily correct as the sensors may loosely be attached or are applied on soft tissue, invalidating the constant bone offset.

Calibration. The calibration serves the purpose to find a sensor's coordinate and bone offset and, in contrast to the Kinect, is necessary for the IMUs. As a sensor's bone offset depends on its coordinate offset, in a first step the sensors have to be aligned to the screen coordinate system. This is done by arranging all sensors such that their positive x-axes point to the desired screen's positive x-axis, their positive z-axes point upwards or their positive y-axes point to the desired front direction respectively. The sensor alignment defines the screen's coordinate system projected into the room and now is called the working coordinate system. Moving with respect to these coordinates directly corresponds to moving within the screen coordinates. After the sensor alignment, the coordinate offset $q_{coordinate\,offset}$ simply becomes the inverse of the current sensor orientation: q_{sensor}^{-1}. This working coordinate system has to be remembered for the next step where the captured object is aligned with respect to these coordinates. In the second step the sensors are placed on the single movable parts of interest and the respective sensor node is assigned to the respective bone in the software toolkit. The object then has to imitate the default pose of the skeleton as seen on screen with respect to the previously defined working coordinates. Each sensor's bone offset $q_{bone\,offset}$ now can be determined by recording its orientation q_{sensor}, mapping it to the skeletal coordinate system using the sensor's coordinate offset $q_{coordinate\,offset}$ and finally computing the rotation quaternion from the sensor node to the default bone orientation $q_{bone,def}$ as stated in (2).

$$q_{bone\,offset} = (q_{bone,def}^{-1} \cdot q_{coordinate\,offset} \cdot q_{sensor})^{-1} \quad (2)$$

Synchronization. To allow the reconstruction of a skeleton pose from sensor data belonging to the same pose at a given time, it is necessary to know the time point at which each measurement was taken. Unfortunately, the received data packets, especially on the UDP broadcast, may arbitrarily be delayed and the packet's receive time does not reflect the time point at which a measurement was taken. Along with the data, each sensor therefore has to provide a so-called time stamp that tells the relative time in milliseconds of the current measurement with respect to the time point of its very first measurement. To find the time point of each measurement on the system's common time scale, each sensor's start time has to be estimated. As a packet can only be delayed, or in other words can only be sent after it was taken, the estimation of a sensor's start time always is a bit too late compared to its real start time. Each newly estimated start time that is smaller (earlier) than the smallest start time so far therefore is a better approximation because the receive time was closer to the measurement time than before. The formula for estimating the start time t_{start} for all n received packets is stated in (3):

$$t_{start} = \min(t_{receive,n} - t_{timestamp,n}) \quad n \in \mathbb{Z} \quad (3)$$

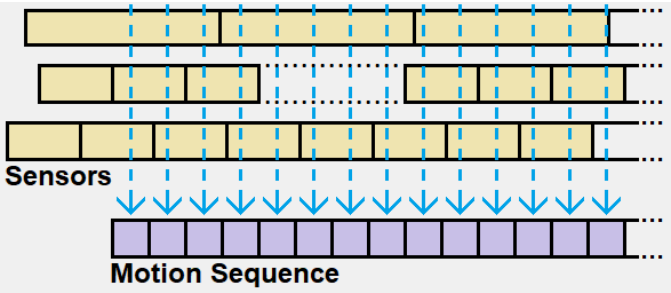

Fig. 3. The sensor data (yellow) has different start times, different sampling times, and in some cases comes from an unreliable connection with missing data. The data is recorded with a fixed sampling time.

Each sensor's start time now is defined on the common time scale provided by the system the software is running on.

Data Processing. Once received, the motion data is forwarded to a session logger and the motion recording routine. The session logger directly dumps all received data along with calibration values and other session specific settings to a log file. The log file allows simulating the overall sensory input from a certain motion capture session at any time, making it possible to restore the whole motion capture session at a later time point as if it was the original session. The single sensors then can be recalibrated or arbitrarily be reassigned to different bones. The motion recording routine on the other hand applies the sensor model to the sensor data and forwards the resulting bone orientations to the motion recorder as described below.

Recording with Time Domain Optimization. In order to record a motion correctly, the recording routine has to take into consideration that each sensor will have a different start time, possibly a different sampling rate, and additionally some data may be missing. This is illustrated in Figure 3. A sensor's start time t_{start} is estimated for each sensor during synchronization. Each data packet n then can be associated with a certain time point $t_{data,n}$ by the sum of t_{start} and its time stamp $t_{timestamp,n}$ as stated in (4):

$$t_{data,\,n} = t_{start} + t_{time\,stamp,\,n} \quad n \in \mathbb{Z} \tag{4}$$

The common time scale resolves the start time of a motion capture session, each sensor's start time, and the respective time points at which the measurements were taken. This enables recording the motion data frame-wise to a so-called motion sequence with a predefined frame time t_{frame}. In terms of the common time scale the time point t_k of each single frame k can be computed with (5):

$$t_k = t_{start\,recording} + k \cdot t_{frame} \quad k \in \mathbb{Z} \tag{5}$$

A motion sequence comprises different channels, each affiliated with a certain bone. The bone orientations at the time points t_k are computed from the sensor data as described above and are inserted into the respective channel into the frame k given by the time point t_k. The data to frame association is shown in Figure 3. When some data packets are lost or when some sensors with a low sampling rate compared to the frame time are used in combination with fast sensors, the resulting motion sequence may have visible artifacts. These issues are tackled by trying to estimate the correct bone orientation for each frame k with a spherical linear interpolation between two successively received data packets $n-1$ and n. Using (4) to obtain t_n and (5) to obtain t_k, the respective interpolated orientation q_k can be computed with (6).

$$q_k = slerp(q_{n-1},\, q_n,\, \frac{t_k - t_{n-1}}{t_n - t_{n-1}}) \qquad t_{n-1} \leq t_k < t_n \quad k, n \in \mathbb{Z} \tag{6}$$

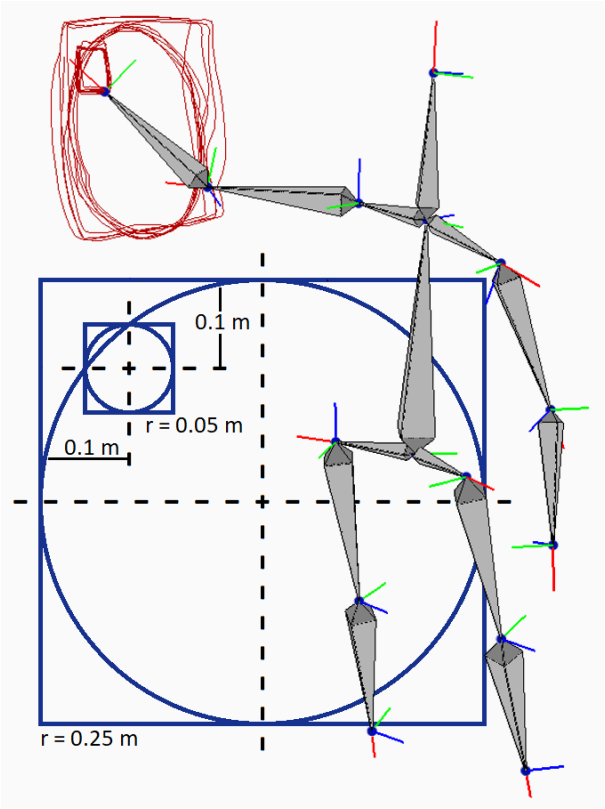

Fig. 4. The pattern to be traced and the visualization of a human tracing this pattern. The large rectangle and circle have a diameter of 0.5 m and the small ones have diameter of 0.1 m. The real-time trace and captured body visualization, on top of the pattern, provide direct feedback during recording.

IV. Evaluation

Method and experiment setup. For our experiments, 10 study participants, all with moderate to extensive presentation experience at a whiteboard, were recruited within our university. Their body heights were between 1.79 m and 1.98 m tall, and all were right-handed.

The evaluation setup consists of the Kinect and two smartwatches, worn on the dominant upper arm and wrist each. On a whiteboard, a pattern was drawn that consists of a large and a small rectangle and a large and a small circle (see Figure 4). The large rectangle and circle have a diameter of 0.5 m and the small ones a diameter of 0.1 m, respectively. During the motion capture session, the study participants are asked to trace the described pattern with their dominant hand at least five times at a pace they could determine. In setting 1, the Kinect was placed to the side of the whiteboard and both, the

current test candidate and the Kinect are oriented such that they face each other during the motion capture (see Figure 1). The captured person's front in this setting is fully visible to the Kinect sensor. In setting 2, the participant was asked to draw the pattern with a natural, self-chosen orientation towards the whiteboard, leaving the Kinect on its previous position to the side. In this setting self-occlusions are not prevented and it can be studied in which extent the Kinect faces problems tracking the arm movements in a more realistic setup. All participants performed setting 1 and four out of the ten participants additionally performed setting 2.

During the study, the participants' whole body is captured with the Kinect and, as described above, the dominant arm and wrist additionally are captured with two smartwatches. The overall raw sensor input during each motion capture session is recorded in a log file such that the session can be restored at any time, equaling a simulation with the same conditions and body movements. The simulated session then is utilized to test three different settings: (1) The whole body is captured only using the Kinect (K), (2) the Kinect's wrist capture is replaced by the respective wrist-worn smartwatch (K W), and (3) in addition to the wrist, also the upper arm is captured by a second smartwatch (K W A).

Evaluation results, qualitative analysis. Study participants were recorded in different settings such that the effect from the level of occlusion could be investigated as a parameter. One of the observations from the first visual inspection of the Kinect's capture data is the detrimental effects that occlusions have on the tracking of the wrist. Only in the very careful placement of the Kinect toward the side of the presenter (i.e., from the viewpoint of Figure 1), the wrist can be tracked at most times with the Kinect alone. Even in such a best-case setup, self-occlusions regularly happen and have led to deviations, as can be seen in Figure 6. This figure also illustrates some effects of the smartwatches' IMU drift, where especially in the X-axis toward the whiteboard, accumulated errors build up in the tracking performance. Due to the aforementioned difficulties to track the wrist with the Kinect alone when major parts of the dominant arm are occluded as can be seen in Figure 5 from setting 2, we focused the qualitative analysis on the Kinect's optimal position and a more occlusion-prone sample. This will allow a comparison of the Kinect's best-case performance to track the wrist position, compared to when the upper and lower arm is tracked with a smartwatch.

Evaluation results, quantitative analysis. The quantitative analysis again is focused on setting 1 to compare the IMU performance to the Kinect in a best-case scenario. The euclidean distances between the great rectangle's and circle's trace points and the quantized ground truth points are calculated. The measure of the overall shape fit is based on the distance from each single sample point to its respective nearest ground truth point by calculating the mean of these distances. It tells how well the trace points are aligned to the ground truth, but missing trace points or a hole in the trace point pattern (see Fig. 6) can not be detected due to the distance-to-nearest-point calculation. To have a measure on how well the shape is covered by the sample points, the mean is computed as before, but this time the distance from each single ground truth point to its respective nearest sample point is considered. The shape coverage measure reflects a hole in the sample pattern, but as only the nearest sample points to the shape are considered, it only reflects how well the best individual trace points are aligned to the shape, not how well the overall trace fits the ground truth pattern. Both measures with mean and standard deviation are listed in table I for the overall data set, and for the best and worst single recordings respectively. The

TABLE I
DISTANCE OF GREAT RECTANGLE AND CIRCLE TRACE TO GROUND TRUTH

Setup	Shape Fit [cm]				Shape Coverage [cm]			
	Rectangle		Circle		Rectangle		Circle	
	μ	σ	μ	σ	μ	σ	μ	σ
Total								
K	4.89	3.36	5.05	3.49	4.53	3.23	4.43	3.17
K W	4.93	3.43	5.15	3.76	3.74	2.90	4.65	3.51
K W A	6.86	3.03	7.40	4.40	5.99	2.77	7.21	4.15
Best Recording								
K	3.87	3.94	4.01	3.61	2.95	3.27	2.33	1.82
K W	4.01	3.78	3.41	2.80	2.25	2.31	2.39	2.25
K W A	5.49	3.86	4.62	3.54	2.91	1.90	4.47	3.33
Worst Recording								
K	4.48	3.67	5.56	3.83	4.49	3.72	4.70	3.17
K W	7.36	4.26	8.37	4.66	7.02	4.51	8.09	4.29
K W A	8.04	4.63	9.39	4.93	8.91	4.62	9.64	4.44

Kinect (K) and Kinect + Wrist (K W) setup perform equally well in the shape fit measure and thus the wrist sensing can easily be replaced by an IMU sensor. The shape coverage measure indicates that at least for the rectangle a better shape coverage can be achieved by using a wrist-worn IMU. Using an additional IMU worn on the upper arm (K W A) introduces errors in the kinematic chain that add up and lead to larger errors on the wrist. In setting 1, where the whole body is seen by the Kinect, the arm mounted IMU therefore does not bring any benefits. Comparing the best with the worst recording, it can be seen how important a good calibration is, as the system is very sensitive to it.

V. CONCLUSIONS

Wrist-worn inertial measurement units (IMUs) are embedded in most smartwatches and can be used to track the wrist's orientation and motion. In this paper, we have shown how IMU data can improve the capturing of a presenter's body, where the tracking accuracy of the dominant hand is especially important. We have shown in a focused experiment with 10 participants that a combination of depth imaging and a wrist-worn smartwatch delivers more robust data: The Kinect suffered severely from self-occlusions of the arm when facing the board, and results from where the arm's segments were replaced with IMU data were significantly better, despite minor sensor drift.

The presented approach allows to digitize presentations in much more detail, by including the presenter's body posture and sketches on large surfaces such as whiteboards, without requiring extra video streams or other tracking devices.

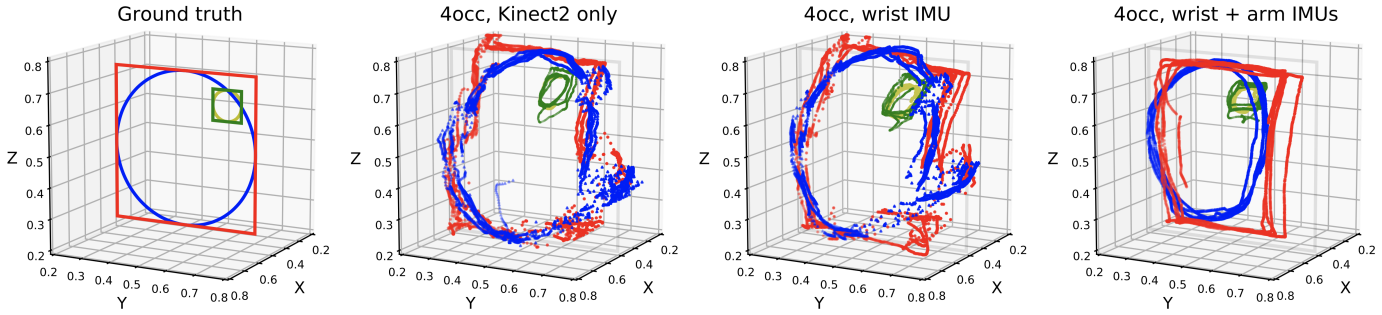

Fig. 5. The performance of setting 2 for the three different approaches in case of the presenter's self-occlusion: The leftmost plot shows the ground truth. The second plot from the left shows the wrist tracking results from just the Kinect's estimates. The third plot shows the Kinect results, with the lower arm segment replaced with the wristwatch's IMU data. The right plot shows the tracking results when both arm segments are replaced by smartwatches.

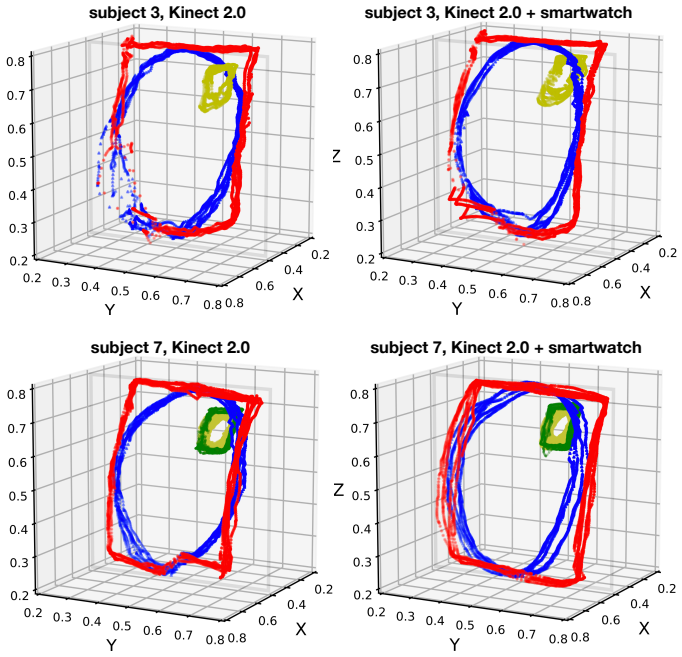

Fig. 6. Two examples illustrating how occlusion distorts the depth imaging's performance (artifacts in the lower area for the left plots). For these recordings, the Kinect was positioned in a *best-case scenario*, i.e., without occlusion from others in the lecture room and tracked from the participant's side to reduce self-occlusion. Tracking performance is improved by replacing the quaternion for the lower arm with the IMU's data, although the latter contains IMU drift.

References

[1] A. Zinnen, K. van Laerhoven, and B. Schiele, *Toward Recognition of Short and Non-repetitive Activities from Wearable Sensors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 142–158.

[2] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: full 6dof human motion tracking using miniature inertial sensors," *Xsens M T BV, Tech. Rep*, 2009.

[3] M. Kok, J. D. Hol, and T. B. Schön, "An optimization-based approach to human body motion capture using inertial sensors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 79–85, 2014.

[4] X. Wu, Y. Wang, C. Chien, and G. Pottie, "Self-calibration of sensor misplacement based on motion signatures," in *2013 IEEE International Conference on Body Sensor Networks*, May 2013, pp. 1–5.

[5] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, oct 2011.

[6] R. Poppe, "Vision-based human motion analysis: An overview," *Comput. Vis. Image Underst.*, vol. 108, no. 1-2, pp. 4–18, 2007.

[7] C. Xu, M. Gao, B. Firner, Y. Zhang, R. Howard, and J. Li, "Towards robust device-free passive localization through automatic camera-assisted recalibration," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys '12*. ACM Press, 2012.

[8] F. Destelle, A. Ahmadi, N. E. O'Connor, K. Moran, A. Chatzitofis, D. Zarpalas, and P. Daras, "Low-cost accurate skeleton tracking based on fusion of kinect and wearable inertial sensors," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, Sept 2014, pp. 371–375.

[9] H.-I. Chang, V. Desai, O. Santana, M. Dempsey, A. Su, J. Goodlad, F. Aghazadeh, and G. Pottie, "Opportunistic calibration of sensor orientation using the kinect and inertial measurement unit sensor fusion," in *Proceedings of the Conference on Wireless Health*, ser. WH '15. New York, NY, USA: ACM, 2015, pp. 2:1–2:8. [Online]. Available: http://doi.acm.org/10.1145/2811780.2811927

[10] A. P. L. B, M. Hayashibe, and P. Poignet, "Joint angle estimation in rehabilitation with inertial sensors and its integration with kinect," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug 2011, pp. 3479–3483.

[11] P. Jatesiktat and W. T. Ang, "Recovery of forearm occluded trajectory in kinect using a wrist-mounted inertial measurement unit," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2017, pp. 807–812.

[12] A. D. Young, M. J. Ling, and D. K. Arvind, "Distributed estimation of linear acceleration for improved accuracy in wireless inertial motion capture," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*. ACM Press, 2010.

[13] Y. Zheng, K. Chan, and C. C. L. Wang, "Pedalvatar: An imu-based real-time body motion capture system using foot rooted kinematic model," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, 2014, pp. 4130–4135.

[14] G. Ligorio and A. Sabatini, "Dealing with magnetic disturbances in human motion capture: A survey of techniques," *Micromachines*, vol. 7, no. 3, p. 43, 2016.

[15] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of microsoft kinect and vicon 3d motion capture for gait analysis," *Journal of Medical Engineering & Technology*, vol. 38, no. 5, pp. 274–280, 2014. [Online]. Available: http://dx.doi.org/10.3109/03091902.2014.909540

[16] K. Otte, B. Kayser, S. Mansow-Model, J. Verrel, F. Paul, A. U. Brandt, and T. Schmitz-Hbsch, "Accuracy and reliability of the kinect version 2 for clinical measurement of motor function," *PLOS ONE*, vol. 11, no. 11, pp. 1–17, 11 2016.

[17] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, Jan. 2013.

[18] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," 2010.