sensors

Article

Hybrid Orientation Based Human Limbs Motion Tracking Method

Grzegorz Glonek †,* and Adam Wojciechowski †

Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology, 215 Wolczanska street, 90-924 Lodz, Poland; adam.wojciechowski@p.lodz.pl

* Correspondence: grzegorz@glonek.net.pl; Tel.: +48-660-758-666

† These authors contributed equally to this work.

Received: 10 November 2017; Accepted: 7 December 2017; Published: 9 December 2017

Abstract: One of the key technologies that lays behind the human–machine interaction and human motion diagnosis is the limbs motion tracking. To make the limbs tracking efficient, it must be able to estimate a precise and unambiguous position of each tracked human joint and resulting body part pose. In recent years, body pose estimation became very popular and broadly available for home users because of easy access to cheap tracking devices. Their robustness can be improved by different tracking modes data fusion. The paper defines the novel approach—orientation based data fusion—instead of dominating in literature position based approach, for two classes of tracking devices: depth sensors (i.e., Microsoft Kinect) and inertial measurement units (IMU). The detailed analysis of their working characteristics allowed to elaborate a new method that let fuse more precisely limbs orientation data from both devices and compensates their imprecisions. The paper presents the series of performed experiments that verified the method's accuracy. This novel approach allowed to outperform the precision of position-based joints tracking, the methods dominating in the literature, of up to 18%.

Keywords: data fusion; depth sensor; Microsoft Kinect; IMU; motion tracking

1. Introduction

Human limbs motion tracking can be defined as a process of unambiguous limbs joints spatial positioning, devoid of significant delays. Nowadays, it might be exploited within several areas, such as entertainment issues, virtual environments interaction or medical rehabilitation treatment [1]. However, it is the system precision that determines its possible application.

Since 1973, when Gunnar Johanson invented his motion capture system [2], such techniques were available mainly for professionals. However, since 2010, when the Microsoft Company (Redmond, WA, USA) released the Kinect v.1 controller, motion capture became available to almost everyone. The Microsoft Kinect controller is an exemplary implementation of the optical, markerless depth sensor with the most remarkable human motion capture functionality. Though it was designed for, and mainly used in the field of home physical activity games, Kinect became a popular subject for many researchers, whose goal was to find other, more advanced scenarios, where this controller might be applied e.g., Lange et al. [3] and Chang et al. [4] designed Kinect based systems to support physical rehabilitation for people with motor disabilities.

Microsoft Kinect represents a broad group i.e. PrimeSense Carmine (PrimeSense, TelAviv, Israel), ASUS Xtion (AsusTek Computer Inc., Taipei, Taiwan) of mono-view markerless depth sensors that exploit structural light patterns to reconstruct the depth of the scene and track perceived objects, with special attention put to human limbs and body parts tracking (i.e., face tracking). This group reveals similar constraints and limits that were usually compensated by sophisticated algorithms of other modes controllers (i.e., inertial, magnetic, acceleration, etc.) [5].

An easy access to the motion capture has been additionally supplemented and magnified by the fact that inertial devices such as gyroscopes and accelerometers (IMU – inertial measurement units) have become an integral and mandatory part of current electronic devices. These units are built in almost every smartphone device and they are used in various applications like pedometers [6,7], screen rotators [8] or digital camera orientation detectors.

Inertial measurement units (IMU) are also used as measuring devices in non-optical, inertial human limbs motion tracking systems i.e., XSense [9,10]. Despite IMU sensors being used in products available for home users, and are being improved continuously, their measurements precision and work limitations can be recognized as blockers for motion capture scenarios that require high accuracy of joint positioning.

The broad and easy access to mentioned types of devices became a trigger for many researchers around the world to work on these controllers' data fusion methods that would increase the accuracy of human skeleton joint positioning. Bo et al. [11] focused on the knee joint angle estimation. They proposed accelerometer and gyroscope data fusion, with linear Kalman filter, to calculate the angle, and then aligned it to the Kinect's estimation. In this approach, Kinect's measurements were treated as reference data so that means authors treated them as always reliable. This also means that the method can only work while Kinect measurements are available. In the case of occlusions or any other disorders, the proposed method will give incorrect results. Destelle et al. [12] have proposed the method based mainly on IMU devices. In this method, human skeleton bones' orientations and overall pose of the body model have been estimated basing on inertial data only, while sizes of skeleton bones were set basing on Kinect estimation. In addition, the skeleton placement was defined by the *Spine* joint position of the Kinect's originated skeleton model. The data from the accelerometer, gyroscope and magnetometer were fused with Madgwick's filter [13]. In Destelle's method, the role of the Kinect was reduced to only two functions: measuring the initial length of bones and tracking the location of the user (its selected joint position) on the scene. The pose of the user was estimated only by IMU. Unfortunately, authors didn't publish results that would allow for defining the accuracy of that method. Kalkbrenner et al. [14] proposed combining together Madgwick's filter with the linear Kalman filter. In their approach, they fused accelerometer and gyroscope data with Madgwick's filter to estimate the selected bone rotation. Basing on this information and on the bone length, estimated with Kinect, joint positions were calculated. In Kalkbrenner's approach, bone lengths were taken from Kinect's data in every measurement frame. This means that this value was dynamic and may differ even a few centimeters between two consecutive frames. In the last step, joint positions, estimated by Kinect, were fused by means of the linear Kalman filter with joint positions, estimated simultaneously with IMU sensors. Kalkbrenner's method was the method with the highest declared precision of the joint position estimation. Published results declare the position estimation error of about 2.2 cm; however, the method was tested on sensors' privileged body gestures. In the method proposed by Feng and Murray-Smith [15], the accelerometer data has been fused with Kinect measurements with the linear Kalman filter, modified by authors to work with frequency unaligned signals. Publicly available and described results of the Murray-Smith method show that the proposed approach allows for stabilizing joint position estimation, at the beginning or at the end of the movement, much faster than the classical Kalman filter implementation. On the other hand, published diagrams show results in a short time period (no longer than 5 s), so it is difficult to retrieve the joint positioning accuracy for this method. Tannous et al. [16] proposed the extended Kalman filter to fuse bone rotations estimated both by Kinect and IMU devices. Their experiments focused on the knees flexion in fully visible position without any occlusion. The elaborated error for knee angle estimation was 6.79° for the left knee and 8.98° for the right knee. The last considered method that addresses a similar problem of upper limb motion tracking with IMU and Kinect devices is the one described in the Tian et al. [17] article. Authors of this method proposed the usage of Unscented Kalman Filter (UKF) for both accelerometer and gyroscope data fusion as well as IMU and Kinect fusion. The uniqueness of the proposed method was taking into consideration geometrical constraints of the human pose. If, according to mentioned constraints, the estimated pose was impossible to achieve, the estimation was aligned to the closest, valid value. Thanks to that approach, authors achieved deviation of rotation angle estimation lower than 20°.

The analysis of the available literature shows that the majority of known data fusion methods is based on different variants of Kalman filter (linear, extended or unscented), to fuse accelerometer with gyroscope data as well as IMU data with these provided by the Kinect device. Only Kalkbrenner et al. decided to fuse accelerometer and gyroscope data with Madgwick's filter, which is designed especially for this kind of data and, regarding the Madgwick's report [13], it has better accuracy and performance than the linear Kalman filter. It is also noticeable that authors of these methods are selectively taking into consideration characteristics and limitations of measurement devices. As an example, Bo et al. [11] and Destelle et al. [12] focused mainly on IMU devices and their intrinsic data fusion, treating Kinect device data as a reliable reference for their methods, despite the known inaccuracy of this device. Kalkbrenner et al. [14] as well as Feng and Murray-Smith [15] tried to compensate some Kinect's measurement problems using proper parameters in the Kalman filter. However, they focused only on joint position estimation fluctuations, e.g., Kalkbrenner recalculated fusion parameters only when the difference in the selected joint position estimation value, between two consecutive measurement frames, was greater than 15 cm. None of the authors included in their methods any compensation for IMU and Kinect's measurement inaccuracy due to the environment conditions (the device operational temperature) or the context of the performed motion (the body rotation angle to the camera, the distance from the measurement device). The fact that these kinds of factors influence noticeably the quality and the accuracy of data measured by both devices may have an impact on the further data processing and on the final joint's position estimation. Authors of the discussed methods fuse the signals of both measuring devices basing on joint positions and they don't take bone orientations into consideration during that process. This fact has an impact on the final results, as joint positions used during the fusion include an additional, significant error caused by the non-fixed bones model length. The data fusion based on bone oation is free of this kind of error and it uses data that are measurable directly by both considered devices.

This paper presents the novel approach to depth sensor (Microsoft Kinect) and IMU data fusion that systemically makes up for limitations of both measurement devices and compensates their imperfections. The novel orientation based, human limbs' joint positions tracking method outperforms state-of-the-art hybrid systems' accuracy by 11–18%.

2. Materials and Methods

2.1. Microsoft Kinect Controller Characteristics

The Microsoft Kinect controller might be described as an RGB-D (Red, Green, Blue and Depth) camera. The first version of this device, originally designed for the Xbox 360 video game console, was built from two CMOS (Complementary metal–oxide–semiconductor) cameras and an integrated infrared (IR) projector. One of these CMOS cameras was responsible for an RGB signal and the second one was calibrated to record IR beam's distribution. Considering the limbs' motion tracking and the body gesture recognition, the most crucial part of Microsoft Kinect controller is the chip made by the PrimeSense company (PrimeSense, TelAviv, Israel). All recognition algorithms that stand behind Kinect's motion tracking were implemented as a firmware for this processing unit. The Microsoft Company has not revealed any detailed description of tracking algorithms and key characteristics, so their retrieval became a subject for scientists and hobbyists [18–20]. The good understanding of the Kinect controller constraints seems to be indispensable to create an accurate method that would fuse its data with data collected from any other source.

In the official device specification, it is claimed that the operating range of Kinect varies between 0.8 m and 4 m with the field of view 57° horizontally (Figure 1a) and 43° vertically (Figure 1b). At the same time, the specification does not reveal the possible heterogenity of the measurement's accuracy. However, the experiments conducted by the authors showed that there are distance measurement nonlinearities within the Kinect's working area. Moreover, researchers [21] and hobbyists [22] reported that the operating range is different among device series.

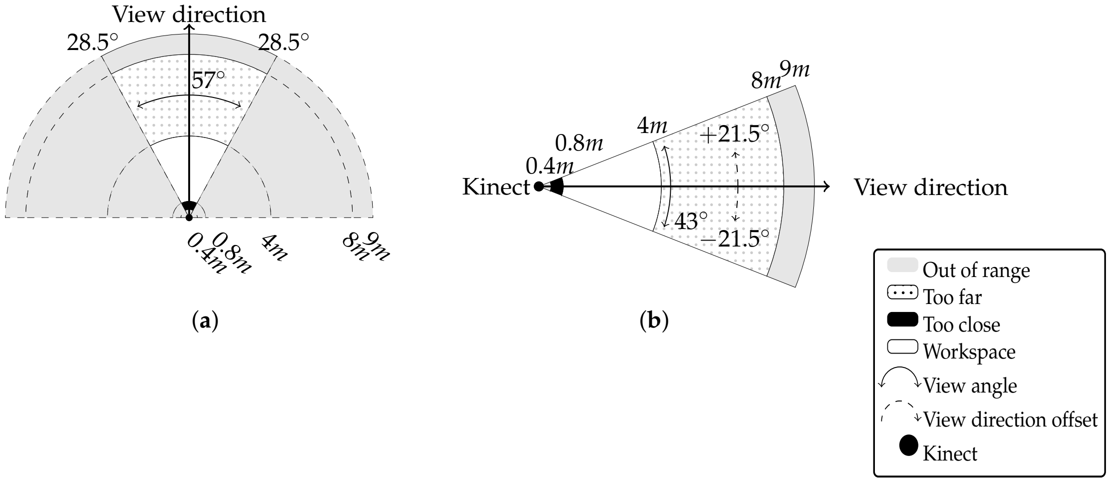


Figure 1. Microsoft Kinect v.1 operating range diagram in horizontal (**a**) and vertical (**b**) directions.

A comparison of fluently changing distances between Kinect and limbs' joints, collected simultaneously from Microsoft Kinect controller and Vicon (Vicon Motion Systems, Inc., Los Angeles, CA, USA) (Vicon—high accuracy, reliable, optical marker-based motion capture system) motion capture system, showed that Kinect has tendencies to underestimate the distance in the close range and to overestimate it in the far range. Collected values (squares on Figure 2) allowed to elaborate a distance estimation error function in a form of 3rd order polynomial, as it is presented in Equation (1), calculated based on Equation (2), where X is the matrix of sample points, Y is the matrix of values in these sample points and A is a coefficients matrix used in Equation (1):

$$f(z) = 0.02z^3 - 0.11z^2 + 0.27z - 0.25, \tag{1}$$

$$X^T X A = X^T Y => \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & x_1^3 \\ x_2^0 & x_2^1 & x_2^2 & x_2^3 \\ x_3^0 & x_3^1 & x_3^2 & x_3^3 \\ \dots & & & \\ x_n^0 & x_n^1 & x_n^2 & x_n^3 \end{bmatrix}^T \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & x_1^3 \\ x_2^0 & x_2^1 & x_2^2 & x_2^3 \\ x_3^0 & x_3^1 & x_3^2 & x_3^3 \\ \dots & & & \\ x_n^0 & x_n^1 & x_n^2 & x_n^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_1^0 & x_1^1 & x_1^2 & x_1^3 \\ x_2^0 & x_2^1 & x_2^2 & x_2^3 \\ x_3^0 & x_3^1 & x_3^2 & x_3^3 \\ \dots & & & \\ x_n^0 & x_n^1 & x_n^2 & x_n^3 \end{bmatrix}^T \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}. \tag{2}$$

Figure 2 presents the plot of function defined with Equation (1). The analysis of this chart shows that Kinect's distance estimation is optimal when the user stands about 2 m from the device (2–2.3 m). Outside this distance range, when Kinect works with a noticeable error, the systematic sensor depth correction should be taken into consideration.

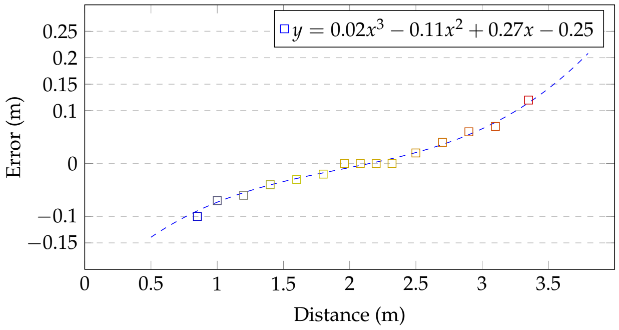


Figure 2. Microsoft Kinect depth measurement accuracy as a function of Kinect–joints distance.

Such inaccuracy and its tendencies are probably caused by the algorithm implemented in the Kinect controller. Though Microsoft did not reveal any precise technical description of their method for measuring distance between the camera and objects at the scene, original patent forms [23–25] owned by PrimeSense, combined with independent research results [26], provide some general overview of how this recognition algorithm works. It is known that the analyzed scene, in front of the Kinect controller, is enlightened with IR light dots structured pattern, which is skew symmetric, so the Kinect can work upright or upside down. Then, Kinect analyses the IR pattern's dots distortion, and, basing on that, it estimates the subject distance. For the depth map reconstruction, Kinect uses two techniques in parallel: dots blurriness analysis [27] and stereo-vision based on a single IR camera and a projector [28]. The achieved depth map is a foundation of a human skeleton estimation algorithm. Depth-retrieved human body parts estimations are subsequently classified into body poses by means of the machine learning approach: random decision forest algorithms as well as object detection algorithms, such as the one proposed by Viola-Jones [29,30].

As the only source of data for that estimation process is an IR camera (RGB camera is not used at all), the observed scene must be free of any external source of the IR light. This requirement restricts the usage of Kinect in outdoor scenarios. A need for the scene isolation from any IR sources other than the Kinect itself is the main reason why Microsoft does not recommend using two or more controllers simultaneously. Nevertheless, scientists invented and published methods on combining signals from several Kinect devices [31–33]. It is noticeable that data gathered from the Kinect is incomplete, due to the lack of information about joint rotation along the bone. This is the result of the skeleton model design, where each of 20 tracked joints is described as a single point.

The last major flaw of Microsoft Kinect controller is an occlusion, which occurs when a part of a user's body is covered by another object or is hidden behind any other body part (self-occlusion). The occlusion by an external object seems obvious and does not require any additional explanation; however, self-occlusions are less intuitive. They are connected with Kinect's sensitivity to user's rotation to the camera (α angle in Figure 3), as Microsoft recommends working with Kinect in the face off pose, which is not precisely defined. Self-performed experiments allowed observing α angle changes while the user rotated in front of the camera. Assuming that $P_{Sh_L} = [p^K_{Sh_L,X}, p^K_{Sh_L,Z}]$ is the position of the left shoulder and $P_{Sh_R} = [p^K_{Sh_R,X}, p^K_{Sh_R,Z}]$ is the position of the right shoulder, both defined in a Kinect controller coordinating system (limited to X and Z axes in Figure 3), then α can be calculated according to Equation (3).

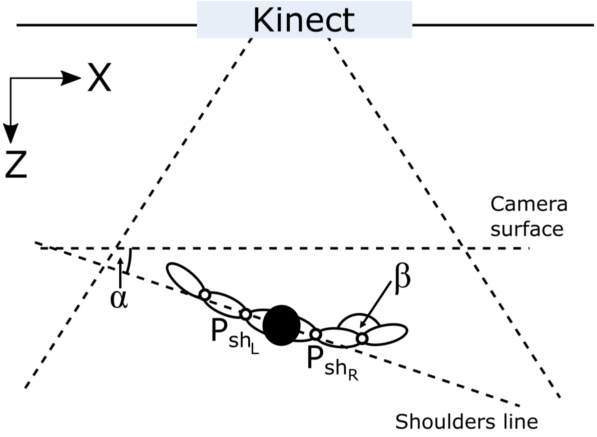


Figure 3. Rotation angle α between the user and Kinect.

$$\alpha = \begin{cases} atan(\frac{|p^K_{Sh_R,Z} - p^K_{Sh_L,Z}|}{|p^K_{Sh_R,X} - p^K_{Sh_L,X}|} & , |p^K_{Sh_R,X} - p^K_{Sh_L,X}| \neq 0, \\ \frac{\Pi}{2} & , |p^K_{Sh_R,X} - p^K_{Sh_L,X}| = 0. \end{cases} \tag{3}$$

There are two possible strategies that Kinect uses when the occlusion happens. Either the device tries to estimate the location of the covered joint or, when it is not able to perform even rough estimation of the joint position, it stops tracking of such a joint. The results of self-performed experiments show that considerable occlusions occur when the user is rotated by more than 50° ($\alpha > 50°$).

Charts presented in Figures 4 and 5 show measured joints tracking states and the elbow angle (angle β in Figure 3) changes during the rotation, respectively. In Figure 4, it is visible that tracking the states of both shoulders joints fluctuate between *Tracked* and *Interferred* states, when the subject was rotated more than 50° ($\alpha > 50°$) in relation to Kinect's camera. It is worth mentioning that even an easily visible right shoulder joint lost tracking state during that rotation. In Figure 5, it can be noticed that the estimation of the right elbow angle β was unstable for considerable body rotation ($\alpha > 50°$), though the hand was fully visible all the time.

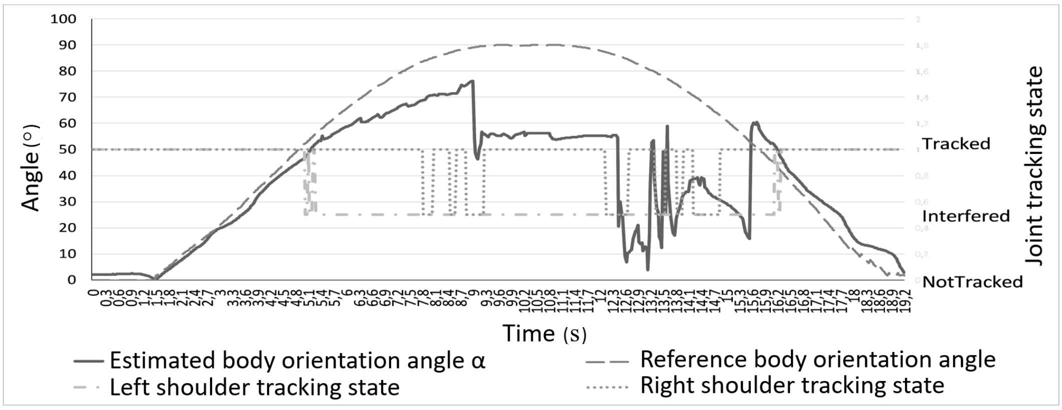


Figure 4. Shoulders joints tracking state in relation to body rotation angle α.

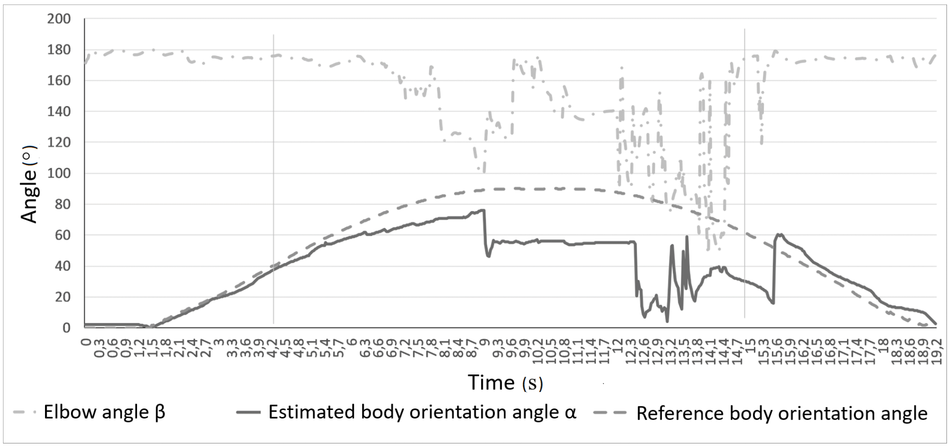


Figure 5. Elbow angle β estimation in relation to body rotation angle α.

2.2. IMU Characteristics

IMU devices have been in professional usage from decades. Within the scope of the current study, there are two types of sensors that measure inertial forces affecting them: accelerometers and gyroscopes. All experiments presented in this paper were performed with the usage of IvenSense MPU-6050 module (San Jose, CA, USA), which integrates both of these types of inertial devices and a thermometer. Even though the accelerometer and the gyroscope were integrated on a single PCB (Printed Circuit Board), their data must be processed individually, as they are affected by the different set of external noises with various frequencies that must be filtered out to make them usable. The IMU device used in our research can be classified according to [34] as an industrial device. Characteristics described in the article are thus representative for other devices belonging to this class.

An accelerometer is an inertial sensor that measures linear forces, affecting it along the device coordinating system axes. Usually, the measurements are defined in relation to the gravity force (1 g $\approx$ 9.81 m/s^2), so it is possible to estimate its linear acceleration during the motion. When the device rests, the theoretically correct measurement should be 1 g in the upward direction and 0 g in two other directions. However, due to external noises (mainly high frequency), the resting device measurements are oscillating around theoretical values. The analysis of these oscillations, during calibration, allows for designing low pass filter [35,36] that would be able to filter out at least major part of these high frequency noises. Accelerometers are also sensitive to operating temperature changes and its influence on sensor's accuracy was the subject of some researches [37,38]. The temperature sensitivity is caused by the architecture of the sensor, which is built with several capacitors, hence the operating temperature has influence on their capacity. During the self conducted experiment, the measurement of the g-force was observed when the resting device was heated and then cooled down multiple times in the temperature range 10–50 °C. The results of this experiment are presented in Figure 6. This chart shows that the measurement tends to change from under to overestimation with the temperature change. It is also worth noticing that the measured value of the gravity is up to 4% different than the expected, what may influence the further results. According to the MPU-6050 specification and results of self-conducted experiment, the neutral operating temperature for the considered module is 25 °C. Due to the fact that the operating temperature (T) of the IMU device, placed on a human body, rises and stabilizes at approx. 30 °C, some sort of compensation (*f(T)*) is required there. Equation (4) presents the formula of the measured error:

$$f(T) = 7 \cdot 10^{-}7T^3 - 5 \cdot 10^{-}5T^2 + 0.0022T + 0.9648. \quad (4)$$

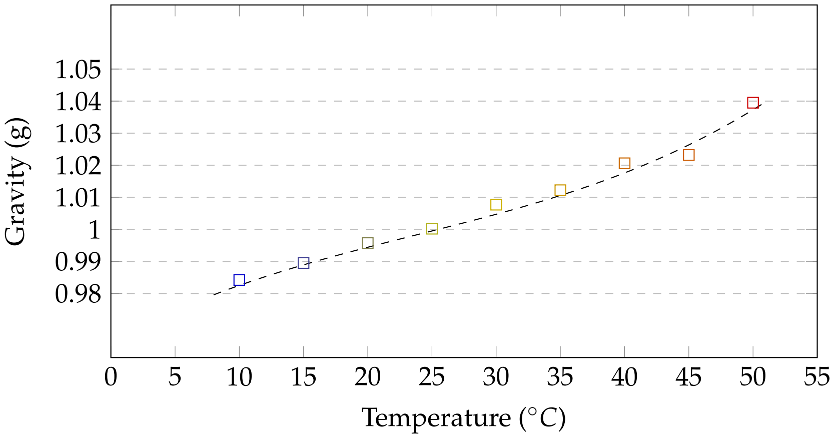


Figure 6. Accelerometer gravity measurements in temperature range 10–50 °C.

A gyroscope—the second type of inertial sensors—measures its angular velocity in deg/s units. If the sensor is in the resting state, all measurements should equal 0 for each of three axes. However, the device suffers from the long-term bias that should be limited (ideally–removed). The analyzed noise has low frequency characteristics, thus the appropriate high pass filter [39], limiting its influence, should be applied. As a MEMS (MEMS—microelectromechanical system) based gyroscope is also built from multiple capacitors, their bias is influenced by the temperature as well. However, as the distortions related to the temperature have low frequency nature, their influence on the data is reduced during signal filtration by a high-pass filter.

The last problem, considered by the authors, related to IMU devices is the incompleteness of the data provided by these units. To calculate the accurate orientation of the IMU module, data from the accelerometer and the gyroscope (orientation around corresponding axes) should be fused together. The accelerometer allows for calculating orientation around two axes, even though it measures forces along all three orthogonal directions. The orientation (or the rotation) around the gravity vector is unmeasurable for such device. Theoretically, the gyroscope should retrieve orientation measures around all three directions; however, even the filtered signal contains some errors. Its numerical integration over time results in the significant data drift that makes such measurements useless. An example of the data drift is presented in Figure 7. Despite the sensor remaining motionless during the experiment, the estimations based on the numerically integrated data showed the sensor theoretical rotation about 60° within 2 min (about 0.5°/s).

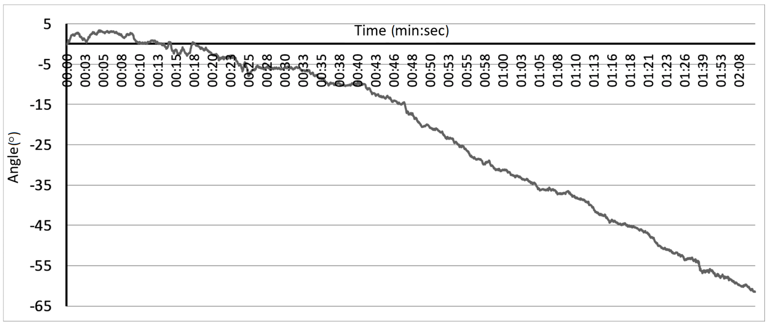


Figure 7. Gyroscope drift for non-moving devices.

2.3. *Hybrid, Orientation Based, Human Limbs Motion Tracking Method*

As an input to the method, two (Kinect and IMU), independently registered, streams of data representing tracked human skeleton bone oations, were considered. The problem that the motion tracking method needs to face was estimating human skeleton selected joint (j) fused (F) from considered data sources, position ($P^F_{j,t} = [p^F_{j,x}, p^F_{j,y}, p^F_{j,z}]_t$) within the referenced three-dimensional coordinating system, in a particular moment of time (t). To solve this problem, the authors propose a novel, hybrid human limbs motion tracking method, based on the continuous linear fusion of skeleton bone oations (contrary to state-of-the-art skeleton joint positions fusion) with respect to the current motion context. It takes into consideration the controllers' reliability and compensates measurement devices' imperfections described in previous chapters. The data fusion method has been split into several, sequential phases presented on a diagram in Figure 8 and described in the further part of this article.

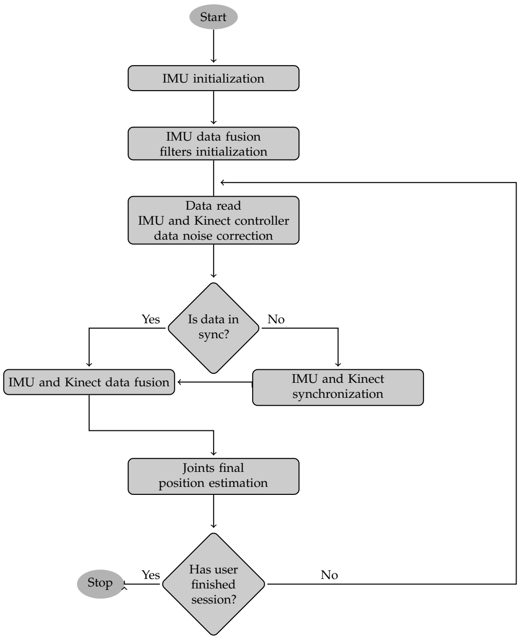

Start
IMU initialization
IMU data fusion filters initialization
Data read IMU and Kinect controller data noise correction
Is data in sync?
Yes
No
IMU and Kinect data fusion
IMU and Kinect synchronization
Joints final position estimation
Has user finished session?
Yes
No
Stop

Figure 8. Orientation based data fusion method schema.

The discussed novel data fusion method can be presented as a general formula with Equation (5):

$$P^F_{j,t} = f(A, G, T, P^K_{j-1,t}, P^K_{j,t}, \Delta t, P^F_{j-1,t}, P^K_{sh_L,t}, P^K_{sh_R,t}, l). \tag{5}$$

The estimation of the selected joint j position ($P^F_{j,t}$) in particular time t is based on the measurements of the accelerometer ($A = [a_x, a_y, a_z]$), the gyroscope ($G = [g_x, g_y, g_z]$) and the positions of the selected joint and its parent, measured by the Kinect controller ($P^K_{j,t} = [p^K_{j,x}, p^K_{j,y}, p^K_{j,z}]_t$, $P^K_{j-1,t} = [p^K_{j,x}, p^K_{j,y}, p^K_{j,z}]_t$). Additionally, to compensate for imperfections of IMU measurements, the current operating temperature T is taken into consideration. In addition, positions of both limb (i.e., shoulder) joints ($P^K_{sh_L,t}, P^K_{sh_R,t}$) are used to determine the reliability of the data and choose the right data fusion formula. The discussed method estimates the joint position with some time interval Δt aligned to the Microsoft Kinect update interval. IMU are stuck to the body surface, on the bone of length l, between the tracked joint j and its parent joint $j - 1$. To define the fused position $P^F_{j,t}$ in superior coordination space, it has to be aligned to the fused position of its parent joint $P^F_{j-1,t}$. The simplified hierarchical model of the limb (described on a hand example), as well as the placement of IMUs, are presented in Figure 9.

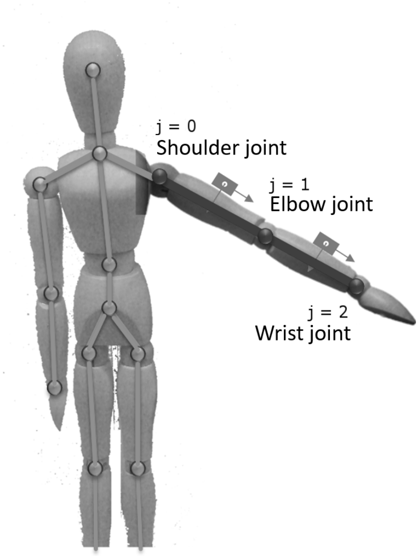


Figure 9. The hand joints simplified hierarchical model.

2.3.1. Coordination Space

Both measurement devices have different local coordination spaces that presented in Figure 10.

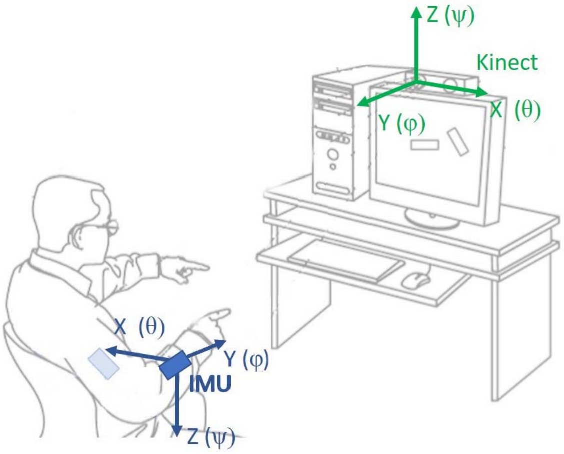


Figure 10. Measurement devices' coordination spaces.

As the majority of the data processed in the method are gathered from the Kinect controller, the global coordination space equals the Kinect's coordination system. This allows for reducing the number of required transformations and to improve the overall performance of the system. To unify both coordination spaces, frame of reference needs to be defined according to which any further rotations are estimated. During the calibration process, when a user stands without motion in T-pose in front of a Kinect camera, such frame of reference is defined. Later, during motion, each IMU estimation ($[x, y, z]_I^T$) needs to be transformed into the Kinect's coordination space ($[x, y, z]_K^T$). As an IMU coordination system is anchored to the gravity vector, and retains its orientation regardless of device rotations, such transformation is defined as a sequence of a rotation by 180° around the y-axis ($R_Y(180)$) and scale (S) transformations according to Equation (6). A definition of one common coordination space for both measurement devices is necessary to fuse their signals:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_K = R_Y(180) S \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I = \begin{bmatrix} \cos(180) & 0 & \sin(180) \\ 0 & 1 & 0 \\ -\sin(180) & 0 & \cos(180) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_I . \quad (6)$$

2.3.2. IMU Initialization

The goal of the IMU initialization procedure, presented in Figure 11, is to calculate the data (G, A) correction factors matrix $(cor = [cor_A \quad cor_G]^T = [[c_a x, c_a y, c_a z] \quad [c_g x, c_g y, c_g z]]^T)$ for each IMU individually.

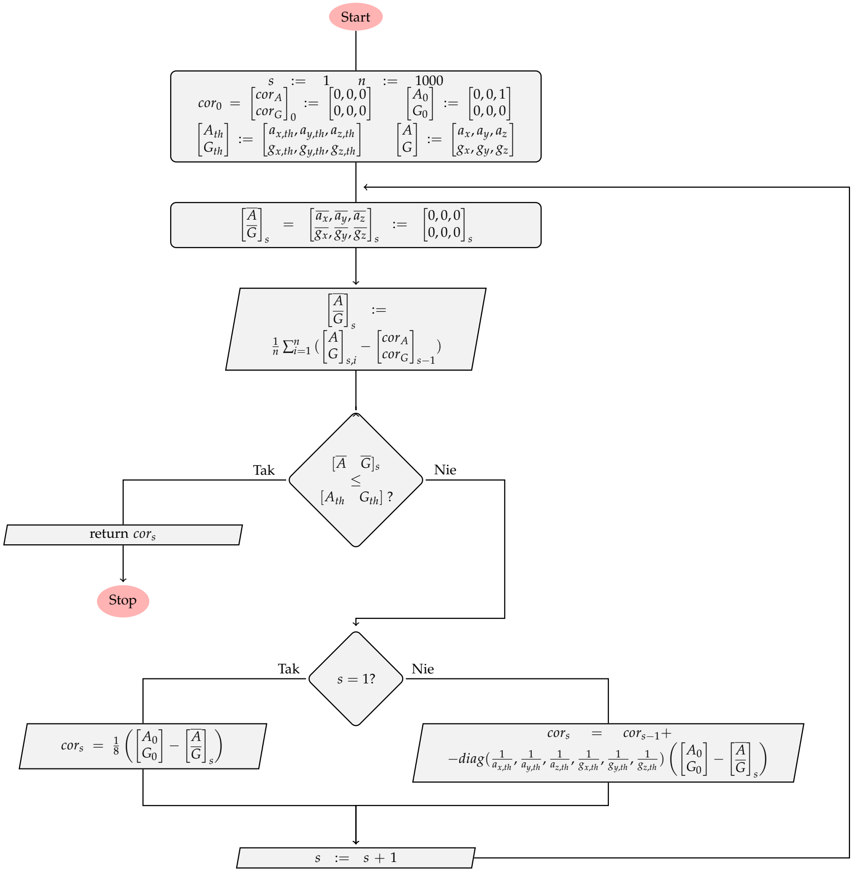

Start
$s := 1 \quad n := 1000$
$cor_0 = \begin{bmatrix} cor_A \\ cor_G \end{bmatrix}_0 := \begin{bmatrix} 0,0,0 \\ 0,0,0 \end{bmatrix}$
$\begin{bmatrix} A_0 \\ G_0 \end{bmatrix} := \begin{bmatrix} 0,0,1 \\ 0,0,0 \end{bmatrix}$
$\begin{bmatrix} A_{th} \\ G_{th} \end{bmatrix} := \begin{bmatrix} a_{x,th}, a_{y,th}, a_{z,th} \\ g_{x,th}, g_{y,th}, g_{z,th} \end{bmatrix}$
$\begin{bmatrix} A \\ G \end{bmatrix} := \begin{bmatrix} a_x, a_y, a_z \\ g_x, g_y, g_z \end{bmatrix}$
$\begin{bmatrix} \overline{A} \\ \overline{G} \end{bmatrix}_s = \begin{bmatrix} \overline{a_x}, \overline{a_y}, \overline{a_z} \\ \overline{g_x}, \overline{g_y}, \overline{g_z} \end{bmatrix}_s := \begin{bmatrix} 0,0,0 \\ 0,0,0 \end{bmatrix}_s$
$\begin{bmatrix} \overline{A} \\ \overline{G} \end{bmatrix}_s :=$
$\frac{1}{n} \sum_{i=1}^{n} (\begin{bmatrix} A \\ G \end{bmatrix}_{s,i} - \begin{bmatrix} cor_A \\ cor_G \end{bmatrix}_{s-1})$
$[\overline{A} \quad \overline{G}]_s \leq [A_{th} \quad G_{th}]$?
Tak
Nie
return cor_s
Stop
$s = 1?$
Tak
Nie
$cor_s = \frac{1}{8} \left(\begin{bmatrix} A_0 \\ G_0 \end{bmatrix} - \begin{bmatrix} \overline{A} \\ \overline{G} \end{bmatrix}_s \right)$
$cor_s = cor_{s-1} +$
$-diag(\frac{1}{a_{x,th}}, \frac{1}{a_{y,th}}, \frac{1}{a_{z,th}}, \frac{1}{g_{x,th}}, \frac{1}{g_{y,th}}, \frac{1}{g_{z,th}}) \left(\begin{bmatrix} A_0 \\ G_0 \end{bmatrix} - \begin{bmatrix} \overline{A} \\ \overline{G} \end{bmatrix}_s \right)$
$s := s + 1$

Figure 11. Diagram of the inertial measurement unit (IMU) initialization routine.

During this initialization, IMU modules must lay down without any motion, in the face-up position, as this is the position, in which we know exactly the expected data ($A_0 = [0, 0, 1]$ for the accelerometer and $G_0 = [0, 0, 0]$ for the gyroscope). However, the ideal values are impossible to achieve due to the noise that cannot be completely removed. Because of this, the calibration routine works iteratively (s—iteration index) as long as the average IMU measurement errors ($[\overline{A} \quad \overline{G}]^T = [[\overline{a_x}, \overline{a_y}, \overline{a_z}] \quad [\overline{g_x}, \overline{g_y}, \overline{g_z}]]^T$) are lower than the defined threshold ($[A_{th} \quad G_{th}]^T = [[a_{x,th}, a_{y,th}, a_{z,th}] \quad [g_{x,th}, g_{y,th}, g_{z,th}]]^T$). In this case, a relation of $[\overline{A} \quad \overline{G}]^T \leq [A_{th} \quad G_{th}]^T$ is true when each element of the first matrix is lower than the corresponding element of the second matrix. As a result of this calibration method, the matrix *cor* is calculated, which elements added to the current IMU data, allow to use data with the desired accuracy.

If $[A \quad G]_{s,i}$ is an i–th out of the n consecutive IMU device measurements for the iteration s, then the average error $[\overline{A} \quad \overline{G}]^T$ is calculated according to Equation (7):

$$\begin{bmatrix} \bar{A}^T \\ \bar{G}^T \end{bmatrix}_s = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} \begin{bmatrix} A^T \\ G^T \end{bmatrix}_{s,i}, & s = 1, \\ \frac{1}{n}\sum_{i=1}^{n} \begin{bmatrix} A^T \\ G^T \end{bmatrix}_{s,i} - \begin{bmatrix} cor_A^T \\ cor_G^T \end{bmatrix}_{s-1}, & s > 1. \end{cases} \tag{7}$$

Then, the data correction factors matrix *cor*, for single measurement, is defined by Equation (8):

$$cor_s = \begin{bmatrix} cor_A^T \\ cor_G^T \end{bmatrix}_s = \begin{cases} \frac{1}{8}\left(\begin{bmatrix} A_0^T \\ G_0^T \end{bmatrix} - \begin{bmatrix} \bar{A}^T \\ \bar{G}^T \end{bmatrix}_1\right), & s = 1, \\ cor_{s-1} - diag(1/a_{x,th}, 1/a_{y,th}, 1/a_{z,th}, 1/g_{x,th}, 1/g_{y,th}, 1/g_{z,th})\left(\begin{bmatrix} A_0^T \\ G_0^T \end{bmatrix} - \begin{bmatrix} \bar{A}^T \\ \bar{G}^T \end{bmatrix}_s\right), & s > 1. \end{cases} \tag{8}$$

2.3.3. IMU Data Fusion Filters Initialization

The Madgwick filter [13], defined with the formula presented in Equation (9), has been used as an IMU module inherent accelerometer and gyroscope data fusion method:

$$Q^I = m(A, G, f_m, \Delta t). \tag{9}$$

This filter estimates the IMU module orientation in the quaternion form (Q^I), using accelerometer data (A), gyroscope data (G) and the filtration factor (f_m) based on the gyroscope drift characteristics. Madgwick filter processes the IMU data with the time interval Δt. Before the filter can run on the real data, it needs to be initialized to set up its internal parameters correctly. Madgwick filter initialization requires running the filter multiple times for the measurements of non-moving accelerometer and gyroscope sensors. During this initialization the filtration factor f_m needs to be set to the relatively high value e.g., $f_m = 2$. After the completion of the initialization process, f_m should be defined according to the average Angle Random Walk (ARW) noise value $\widetilde{\omega}$ that can be calculated with the Allan's variance [40–42]. In case of exploited MPU-6050, IMU modules filtration factor f_m is calculated according to the original article [13] equation: $f_m = \sqrt{\frac{3}{4}}\widetilde{\omega}$, and equals 0.082.

2.3.4. IMU and Kinect Controller Data Noise Correction

The data noise correction is a crucial phase for the data fusion. Its goal is to remove as much noise as possible from the raw data. All methods used in this phase are related to measurement devices characteristics described earlier in this article.

The first step to improve the IMU module data quality is the accelerometer measurements (A) error compensation due to the device operating temperature (T). For the neutral temperature (T_0 = 25 °C) and the correction factor $f_T = 0.0011$, the correction is defined with Equation (10). The value of f_T has been estimated in self-conducted experiments:

$$A' = A/(1 + f_T(T - T_0)). \tag{10}$$

The low and high pass filtration of accelerometer and gyroscope data is a part of Madgwick filter implementation, so there is no need to repeat this process. The second measurement device—Kinect controller—requires two data correction steps in order to improve the measurement quality: the correction of the distance estimation and smoothing the joint position estimations. The distance estimation correction is done by the function presented in Equation (11), which is the opposite of the distance estimation error model (Equation (1)). The argument of this function (z) is directly taken from the Kinect's distance estimation API:

$$f(z) = z' = -0.02z^3 + 0.11z^2 - 0.27z + 0.25. \tag{11}$$

The smoothing of the Kinect's joint positions ($P^K_{j,t}$) is necessary as coordinates' fluctuations may happen, especially when occlusions occur. While there is an occlusion, the position of the same joint in two consecutive frames (measurements) can differ few centimeters, which is physically and anatomically impossible, while refresh rate of the Kinect is 30 Hz. The smoothing can be done by the Kinect's firmware, but this approach gives a significant delay of the signal interpretation. The alternative approach to positions smoothing, which was exploited in the method, is a simple low pass filter. The method uses the 1st order exponential filter to remove unreliable positions estimations defined with Equation (12) and the filtration factor $f_{LPF} = 0.065$:

$$P'^K_{j,t} = f_{LPF} \cdot P^K_{j,t} + (1 - f_{LPF}) \cdot P'^K_{j,t-1}. \tag{12}$$

The value of f_{LPF} has been chosen as a compromise between error of estimation and smoothed signal delay. Figure 12 presents charts of estimation error and signal delay related to f_{LPF} factor of proposed smoothing function.

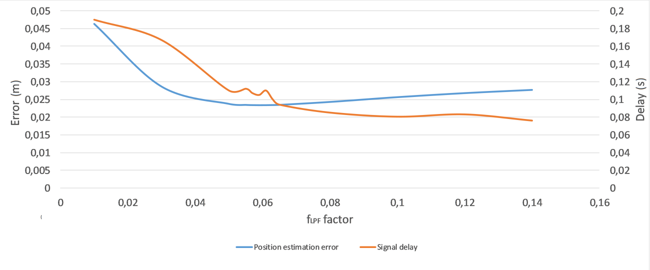


Figure 12. Estimation error and signal delay of smoothing function charts.

2.3.5. IMU and Kinect Synchronization

The data fusion of any two signals requires signal alignment in the time domain. This guarantees fusion of samples that were collected in the same time synchronously. The goal of IMU and Kinect signals synchronization is to find the time offset τ between two data streams. To synchronize both signals, they must have the same frequency, so the IMU signal has to be downsampled from 100 Hz to 30 Hz, which is a nominal frequency of the Kinect controller. The downsampling has been implemented in the form of decimation, where Kinect's new sample availability defines the time of IMU sample picking. Then, the time offset τ between the IMU signal I and the Kinect signal K was defined as a time offset argument of the cross-correlation algorithm that gives the maximum value of correlation between these two signals (variable τ_{max}) (Equations (13a) and (13b)). The value of τ_{max} is added to the timestamp of IMU samples to align it to the Kinect's signal:

$$(I * K)(\tau) = \int_{-\infty}^{+\infty} I(t)K(t+\tau)dt, \tag{13a}$$

$$\tau_{max} = \underset{\tau}{argmax}((I * K)(\tau)). \tag{13b}$$

2.3.6. IMU and Kinect Data Fusion

The data fusion method processes time aligned filtered data and takes into consideration their reliability. The most serious concerns about the quality of data used within the fusion were related to the data provided by Microsoft Kinect controller. The decision about Kinect's data quality was based on the several information gathered during the motion. The first one is the combination of the joint tracking state and the noise level of its position estimation. A fully visible joint has its state set to *Tracked* value and its position can be considered as reliable. Otherwise, if the tracking state value is set to *Interferred*, the position estimation noise level must be calculated to check if the value can be treated as reliable or not. In the *Interferred* state, the position of the joint is estimated basing more on the predication than direct measurements, so its accuracy might be low and values may differ significantly between consecutive frames (or in a short time period). Other parameters taken into consideration are: the value of the angle α between the user and the camera, as well as the information about the tracking state of both shoulder joints. Microsoft Kinect has been designed to track the motion of the human who stands frontally to the camera. If the user rotates too much, Kinect is not able to track the motion correctly anymore. During self experiments, it turned out that the maximum reliable angle between the human and the Kinect is 50°. Exceeding this rotation angle results in unreliable, often random, values of all joint positions. As the value of the α angle is calculated using position values of both shoulder joints, both of them should be fully visible (tracking state set to *Tracked*), and their position values estimations should be stable in time. This means that the angle measurement standard deviation should be lower than 1.5°, so there should be no rapid changes in shoulders position estimations, which are characteristic for the significant error in the provided data.

If all of the mentioned conditions are satisfied, signals are fused according to Equation (14). The novelty of the presented method, in comparison with literature approaches, lies in the fact that the conditionally fused data represents skeleton bone oations in the form of Euler angles $E = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$, instead of joints resulting positions. Microsoft Kinect controller provides natively information about selected bone oation in the form of quaternion, calculated using current bone's joint positions. Used in the presented method, Madgwick's filter also provides IMU orientation as single quaternion due to the fact that orientation fusion, described in this article, was defined with Euler angles. Quaternions from both sources need to be converted to this form before they are fused. To prevent the Euler angle's singularity problem, during the conversion from quaternion, an additional check is performed. This step is supposed to detect if the sensor is oriented vertically up or down and, if it is, convert data to fixed values. The singularity is detected by the sum of multiplications of quaternion's values:

$$c_s = q_X q_Y + q_Z q_W.$$

If the value of c_s is greater that 0.4999, it means that the sensor is oriented vertically up, and, if it is lower than -0.4999, then it is oriented down. In case the singularity appears, the ϕ, θ, ψ are defined as:

- $\phi = 0$,
- $\theta = \pm 2 \tan^{-} 1(q_X / q_W)$,
- $\psi = \pm \Pi / 2$.

The ± is related to the sign of the c_s value. The proposed data convertion method is based on the assumption that we need to align estimations to some fixed value around the singularity point. This means that we lose real measurements. On the other hand, ±0.4999 threshold value cuts off the orientation angle with a value exceeding ±87°. Details about the conversion from the quaternion to the Euler angles can be found in [43]:

$$E_t^F = \begin{bmatrix} \phi^F \\ \theta^F \\ \psi^F \end{bmatrix}_t = diag(w_\phi, w_\theta, w_\psi) \begin{bmatrix} \phi^I \\ \theta^I \\ \psi^I \end{bmatrix}_t + diag(1 - w_\phi, 1 - w_\theta, 1 - w_\psi) \begin{bmatrix} \phi^K \\ \theta^K \\ \psi^K \end{bmatrix}_t . \tag{14}$$

The originally elaborated weights $[w_\phi, w_\theta, w_\psi]$ describe the importance level of IMU measurements. They were based both on the device precision and the data completeness, and were defined as $[0.98, 0.05, 0.65]$, respectively. The very high and very low first and second values are the result of measurement devices' characteristics described in the beginning of this article. As Kinect is not able to detect Roll rotation, the estimation of IMU is promoted (the first value). Similarly, the second value that relates to the Yaw rotation, for which IMU estimation is considerably less accurate than Kinect's estimation, has been depreciated with a low second coefficient value. The Pitch rotation can be estimated by both devices; this is why the third value is defined close to 0.5. Exact values are the results of self-made research about estimating arm and forearm bone oations by both measurement devices while the hand motion was performed only in one, selected degree of freedom. Such constraint guaranteed that possible, accidental rotations around other axes than the considered one have no impact on gathered results. Figures 13–15 present charts of the influence of w factor's value to the estimation error of each rotation during mentioned experiments.

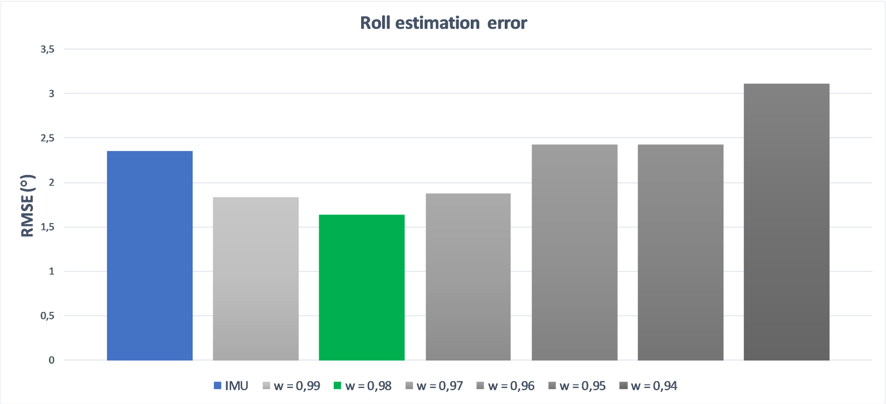


Figure 13. Influence of w_{ϕ} to the roll rotation estimation error–roll for Kinect is not available.

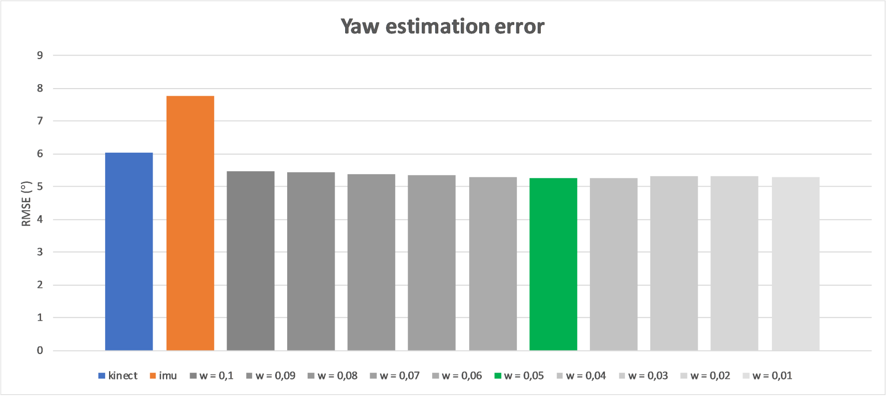


Figure 14. Influence of w_{θ} to the yaw rotation estimation error.

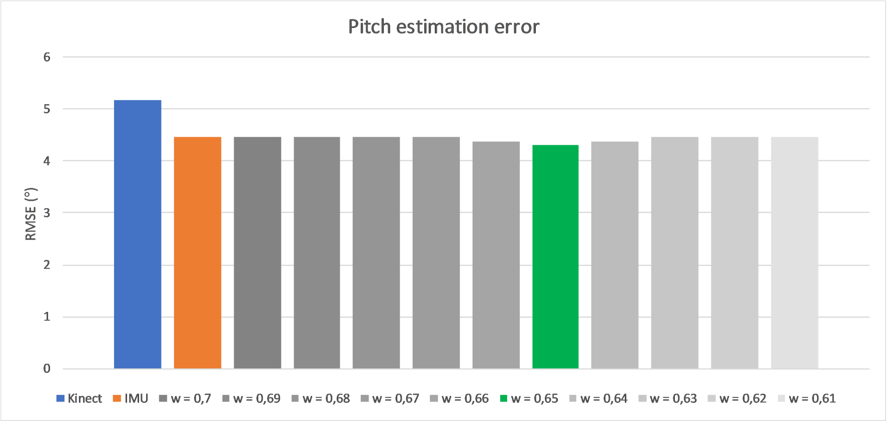


Figure 15. Influence of w_{ψ} to the pitch rotation estimation error.

If Kinect's data turn out to be unreliable, they cannot be fused with the IMU data. Averaged bone oations are then estimated basing on the previously ($t-1$) fused estimation and the current change of IMU orientation, according to Equation (15):

$$E_t^F = \begin{bmatrix}\phi^F \\ \theta^F \\ \psi^F\end{bmatrix}_t = \begin{bmatrix}\phi^F \\ \theta^F \\ \psi^F\end{bmatrix}_{t-1} + diag(w_\phi, w_\theta, w_\psi)\ (\begin{bmatrix}\phi^I \\ \theta^I \\ \psi^I\end{bmatrix}_t - \begin{bmatrix}\phi^I \\ \theta^I \\ \psi^I\end{bmatrix}_{t-1}). \tag{15}$$

However, in the case of Kinect data unreliability, weights $[w_\phi, w_\theta, w_\psi]$ are defined differently. In Equation (15), these weights were defined as

$$[0.98, (1 - t_{noise}/10) \cdot 0.65, (1 - t_{noise}/10) \cdot 0.65],$$

where t_{noise} coefficient represents the time in seconds, when Kinect stayed unreliable and its maximum accepted value is set to 10s. After this time, the estimation of the new bone orientation (only around two axes) stops, until Kinect is available again, and the method privileges the IMU signal. In this scenario, as the IMU is in fact the only source of up to date data, the initial coeficient of Yaw rotation (w_θ) has been increased compared to the value in the scenario when both Kinect and IMU data are available to fuse. Leaving this value on the low level caused the rotation to not be detectable.

2.3.7. Joints Final Position Estimation

The final joint position ($P_{j,t}^F$) estimation requires the information regarding the position of the joint's parent in the hierarchical skeleton model ($P_{j-1,t}^F$), the estimated while calibration (and fixed) length of the bone between the current joint and its parent (l) and the bone orientation estimation ($E_t^F = [\phi^F, \theta^F, \psi^F]_t$). As a result of the data fusion, estimated orientations were presented in the form of Euler angles. However, this representation of the orientation may complicate further calculations. To simplify them, the opposite conversion from Euler angles to the quaternion Q_t^F is recommended, according to the formula described in [43]. Since the skeletal model is defined as a hierarchical structure, the newly estimated joint position must be aligned to the position of its parent joint. Taking that into consideration, the final joint position calculation procedure is described with Equation (16):

$$P_{j,t}^F = P_{j-1,t}^F + Q_t^F \cdot [l, 0, 0] \cdot Q_t^{F^{-1}}. \tag{16}$$

3. Results

In order to verify the accuracy of the elaborated method, several experiments were performed. They were conducted with the Vicon motion capture system, as a source of ground truth reference data. Hand movements of three subjects were monitored, being tracked simultaneously with Kinect controller, two IMUs attached to the hand's arm and forearm bones on the skin surface, as well as the Vicon tracking system. In current implementation of the proposed method, we have made the assumption that each sensor is statically assigned to the particular limb. This means that dedicated (to arm and forearm) sensors are assigned to arm and forearm, respectively, and can not be misplaced. Automatic sensors identification and their relative hierarchical assignment leave room for further method improvements.

Each user had to perform four exercises, each of them twice with five repetitions per single try. Each try started with the devices' synchronization in the T-Pose, and then five repetitions of the motion began. The initial T-pose has been used as a frame of reference and joint positions were updated according to this position. Exercises focused on the horizontal and vertical hand flexion in the elbow joint, horizontal hand motion and keeping a straight arm in the T-pose with no motion for at least 60 s. All of these exercise schemes are presented in Figure 16.

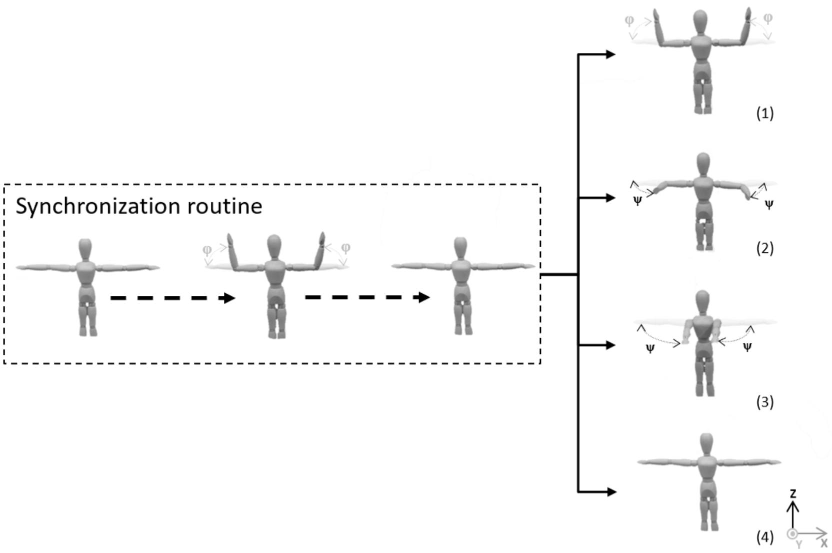


Figure 16. Movement sequences performed during tests.

Three out of four motion trajectories used in the verification experiments can be considered as difficult according to the measurement devices' characteristics.The first motion is relatively easy to track by both devices as no joints occlusion was observed there, and it was fully measurable by the accelerometer as well as the gyroscope. The second and the third trajectories define motions around the axis, where the IMU device was not able to correctly measure current orientation and Kinect controller loses visibility of the joint due to the self occlusion. The important difference between these two motions was the number of occluded joints: the elbow only in the second motion and the elbow and the shoulder in the third. This difference had an impact on Kinect's estimations reliability, which was worse when the shoulder was not fully visible to the controller's camera. The last motion focused on IMU's constant measurement drift around the axis perpendicular to the gravity vector. Mentioned physical body movements let us claim that selected exercises represent the most challenging trajectories that the motion tracking system can face, so it should be able to track other trajectories with similar or better accuracy. The pace of the movements was characteristic for the movements of rehabilitation's physical exercises and the subjects could perform the motions according to individual preferences, without any specific constraints. The only formal pace limitation is the Microsoft Kinect tracking performance (Kinect has 30 Hz slower operating frequency than the exploited IMU controller).

For the self made measurement device used in my studies, both IMU work with different addresses in I2C communication channel and, in order to simplify the code and processing in the current version of the method, sensors are statically assigned to the limbs/bones. It means that the sensor with the channel address *addr1* should always be attached to the upper arm and *addr2* to the forearm (or generally *addr1* to the parent bone and *addr2* to its child).

The accuracy of three parameter estimations has been observed: the position of the elbow joint, the position of the wrist joint and the angle between the arm and the forearm measured in the elbow joint. Those values have been compared with the corresponding parameters estimated with Kalkbrenner's method [14], which, on the contrary, fused positions instead of orientations of selected joints. The position estimation error has been defined as the root mean squared error ($RMSE$) of the Euclidean distance (d_e) between joint reference position, measured by the Vicon motion capture system (P_j^V), and joint position estimated by two competitive data fusion methods. The first one is the joint position-based Kalkbrenner data fusion method (P_j^{FP}) and the second: the bone rotation-based fusion method (P_j^{FO}) proposed by the authors of this article. The $RMSE$ value has been calculated with Equation (17) [44]. The error for the elbow joint angle β estimation has been calculated in a similar way:

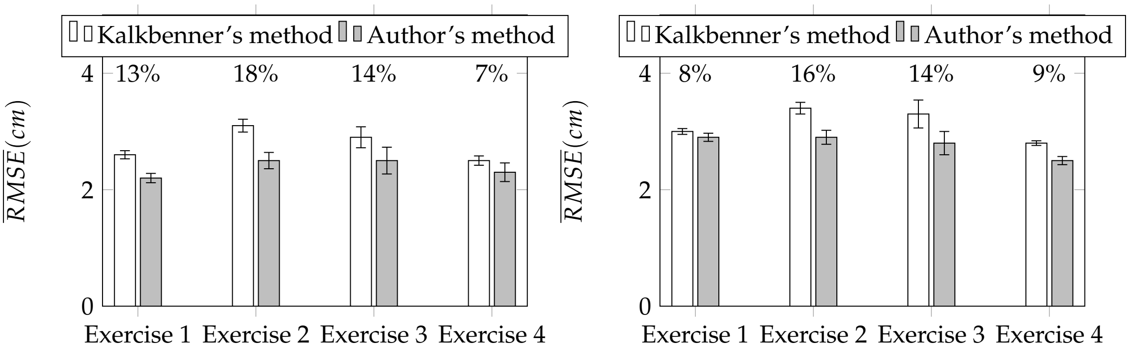
$$RMSE_j^F = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_e(P_{j,i}^V, P_{j,i}^{Fs})^2} \quad ,s = \{O,P\}. \tag{17}$$

The overall error of both methods, for each test exercise, has been calculated as a mean of $RMSE$ defined with eqation

$$\overline{RMSE} = \frac{\sum_{i=1}^{n} RMSE}{n}$$

of each motion tracking session. To compare Kalkbrenner's method with the authors' novel method, the ratio r between the difference of $\overline{RMSE}$ of both methods to the $\overline{RMSE}$ of Kalkbrenner's method has been calculated according to Equation (18) and expressed in the form of percents. Figure 17a,b show the summary of $\overline{RMSE}$ for tracked upper limb joints: the elbow and the wrist. Above each pair of charts' bars, the value of r ratio has been presented. Based on the results presented in Figure 17a,b, the improvement of the proposed orientation-based fusion method, in reference to the best in the literature reference—Kalkbrenner's method, has been noticed. However, the improvement ratio varies between joints. The error of the elbow joint position estimation has been reduced up to 18% (from about 3.1 cm to 2.6 cm) and the wrist joint position estimation error up to 16% (from about 3.4 cm to 2.9 cm). The difference in the improvement ratio between these two joints is caused by the fact that the elbow joint is closer to the skeleton model root than the wrist joint, so it accumulates less ascending joint position estimation errors. Figure 18 presents the chart of $\overline{RMSE}$ of the elbow angle β estimation (calculated similarly to Equation (17)). In this value estimation, the improvement has been also noticed and its ratio is up to 11% (from about 4.6° to 4.1°):

$$r = \frac{\overline{RMSE_j^P} - \overline{RMSE_j^O}}{\overline{RMSE_j^P}}. \tag{18}$$



(**a**) (**b**)

Figure 17. Average root square mean error $\overline{RMSE}$ of the elbow (**a**) and the wrist (**b**) joint position estimation.

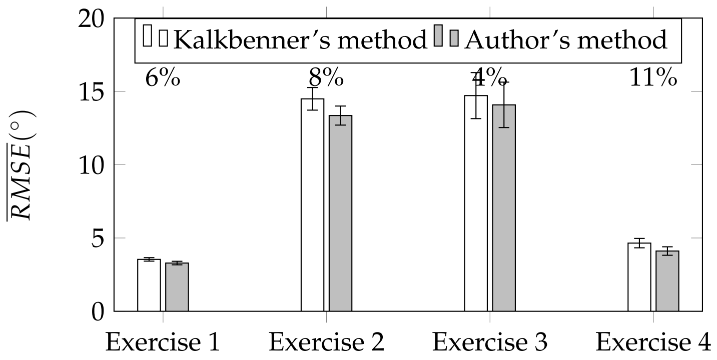


Figure 18. Average root square mean error $\overline{RMSE}$ of the elbow flexion angle β.

4. Discussion and Conclusions

The authors presented a new, orientation-based method for skeleton joint positioning, which compensates devices' imperfections and considers the context of the motion. It was tested on a set of right hand movements, which appeared to be demanding for measurement devices. Results were compared with those gathered from the position-based fusion method and the reference, professional, ground truth Vicon tracking system. Presented results prove that the newly elaborated, complex approach reduces joint position estimation error. The principle reason of such estimation improvement, identified by the authors, is related to the better stability of bone rotation estimation in reference to the position-based method, as well as the diminished inaccuracy of bone length measurements and the lack of the fixed skeleton model definition support. Moreover, the authors have found that the introduced complementary fusion of the signals, measured by both devices, reduces the influence of source signals' imperfections on the final result. In the case of IMU devices, the information about their 3D space orientation (for IMU devices stuck to the body, this orientation also represents the orientation of the bone) was the default and required less complex data processing. The estimation of the joints' positions, based on IMU measurements, required additional measurement of the selected bone length and additional geometric transformations (geometric transformations required to estimate the joint position based on IMU measurement has been described in detail in [14]), which have an impact on the final joint position estimation error. Considering the Microsoft Kinect controller, the human body skeletal model includes information about each bone length and its orientation. However, these length values are updated continuously and may differ significantly between two consecutive measurement frames. The difference might be even a few centimeters. During experiments, the measured differences in bone length were up to 5 cm for *Tracked* state joints. Moreover, even if the occlusion doesn't occur, the estimation of angles between selected bones turns to be more stable in time than the estimation of joint positions. This means that the estimation of bone rotation is more reliable than joint position estimation. Furthermore, the bone rotation is the information available from both signal sources with minimal raw data processing, which has an impact on the quality of data used in the fusion.

What differentiates the proposed method from the already known methods described in the literature is the fact that it takes into consideration characteristics of the classes of chosen measurement devices, and provides the complex compensation of detectable errors that exist in their data. It also allows for improving the quality of the source data before their fusion. It is important that many of the characteristics described in this article adaptively respect the context of the performed motion. As an example, the variable accuracy of the distance measured between Kinect and the user, or the human rotation angle relative to the surface of Kinect, has an impact on the stability and reliability of Kinect's data.

The goal of the conducted research was to find the method of human limbs motion tracking, which would take into consideration elaborated characteristics and would reduce joint position estimation error significantly. The proposed method, tested on the example of the upper limbs motion, has met this goal. The results obtained by the authors' method have been compared with the results of the position-based data fusion method with the highest declared accuracy. Three parameters estimated by both methods have been compared: the elbow joint position error, the wrist joint position error and the elbow angle β error. The set of motions (Figure 16) used to test these methods have been chosen to check how methods work when source data is of poor quality due to the limitations and imperfections of measurement devices.

The results of the comparison of the elbow joint position estimation error are presented in the chart in Figure 17a. It shows that the achieved improvement in this parameter estimation is up to 18%. The comparison of the wrist joint position estimation error is presented on the chart in Figure 17b and the improvement of this parameter estimation is slightly lower than the improvement of the elbow joint position estimation error and its value is up to 16%. The difference in these values is caused by the distance between each of these joints and the root of the skeletal model. The joint that is further from the root accumulates existing errors of all joints that are defined between the root and itself. This explains why the improvement of the estimation error is lower for the wrist than for the elbow. The chart presented in Figure 18 shows the results of the comparison of the third parameter, which is the elbow angle estimation error during the motion. The best improvement in this parameter estimation was close to 11%. However, in exercises 2 and 3, which might be considered as the most difficult from the data quality detection point of view, the improvement is noticeably lower.

Obtained results prove that the novel data fusion approach, based on skeletal bone rotations, might be considered as an improved alternative to the well-known, joint position-based methods.

Author Contributions: Grzegorz Glonek designed the method and experiments and analyzed the data; Adam Wojciechowski contributed and supervised the work; Grzegorz Glonek and Adam Wojciechowski wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RMSE	Root Mean Squared Error
IMU	Inertial Measurement Units
LPF	Low Pass Filter

References

1. Glonek, G.; Pietruszka, M. Natural user interfaces (NUI): Review. *J. Appl. Comput. Sci.* **2012**, *20*, 27–45.
2. Johansson, G. Visual perception of biological motion and a model for its analysis. *Percept. Psychophys.* **1973**, *14*, 201–211.
3. Lange, B.; Koenig, S.; McConnell, E.; Chang, C.Y.; Juang, R.; Suma, E.; Bolas, M.; Rizzo, A. Interactive game-based rehabilitation using the Microsoft Kinect. In Proceedings of the 2012 IEEE Virtual Reality Short Papers and Posters (VRW), Costa Mesa, CA, USA, 4–8 March 2012; pp. 171–172.
4. Chang, Y.J.; Chen, S.F.; Huang, J.D. A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. *Res. Dev. Disabil.* **2011**, *32*, 2566–2570.
5. Helten, T.; Müller, M.; Seidel, H.-P.; Theobalt, C. Real-time body tracking with one depth camera and inertial sensors. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1105–1112.
6. Huang, B.; Wu, X. Pedometer algorithm research based-matlab. *Adv. Comput. Sci. Inf. Eng.* **2012**, *1*, 81–86.
7. Jayalath, S.; Abhayasinghe, N. A gyroscopic data based pedometer algorithm. In Proceedings of the 2013 8th International Conference on Computer Science and Education, Colombo, Sri Lanka, 26–28 April 2013; pp. 551–555.
8. Pedley, M. Tilt sensing using a three-axis accelerometer. *Freescale Semicond. Appl. Notes* **2013**, *3*, 1–22.
9. Roetenberg, D.; Luinge, H.; Slycke, P. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors. *Xsens Motion Technol. BV* **2009**, *1*, 1–9.

10. Xsens Motion Technologies. Products—Xsens 3D motion tracking. Available online: https://www.xsens.com/products/ (accessed on 8 December 2017).
11. Bo, A.P.L.; Hayashibe, M.; Poignet, P. Joint angle estimation in rehabilitation with inertial sensors and its integration with Kinect. In Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, 30 August–3 September 2011; pp. 3479–3483.
12. Destelle, F.; Ahmadi, A.; O'Connor, N.E.; Moran, K.; Chatzitofis, A.; Zarpalas, D.; Daras, P. Low-cost accurate skeleton tracking based on fusion of kinect and wearable inertial sensors. In Proceedings of the European Signal Processing Conference, Lisbon, Portugal, 1–5 September 2014; pp. 371–375.
13. Madgwick, S.O.H.; Harrison, A.J.L.; Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. In Proceedings of the IEEE International Conference on Rehabilitation Robotics, Zurich, Switzerland, 29 June–1 July 2011; pp. 179–185.
14. Kalkbrenner, C.; Hacker, S.; Algorri, M.E.; Blechschmidt-trapp, R. Motion capturing with inertial measurement units and Kinect - tracking of limb movement using optical and orientation information. In Proceedings of the International Conference on Biomedical Electronics and Devices, Angers, France, 3–6 March 2014; pp. 120–126.
15. Feng, S.; Murray-Smith, R. Fusing Kinect sensor and inertial sensors with multi-rate Kalman filter. In Proceedings of the IET Conference on Data Fusion and Target Tracking 2014 Algorithms and Applications, Liverpool, UK, 30 April 2014; pp. 1–9.
16. Tannous, H.; Istrate, D.; Benlarbi-Delai, A.; Sarrazin, J.; Gamet, D.; Ho Ba Tho, M.; Dao, T. A new multi-sensor fusion scheme to improve the accuracy of knee fexion kinematics for functional rehabilitation movements. *Sensors* **2016**,*16*, 1914–1931. doi:10.3390/s16111914.
17. Tian, Y.; Meng, X.; Tao, D.; Liu, D.; Feng, C. Upper limb motion tracking with the integration of IMU and Kinect. *Neurocomputing* **2015**, *159*, 207–218.
18. Skalski, A.; Machura, B. Metrological analysis of microsoft kinect in the context of object localization. *Metrol. Meas. Syst.* **2015**, 469–478, doi:10.1515/mms-2015-0050.
19. Gonzalez-Jorge, H.; Riveiro, B.; Vazquez-Fernandez, E.; Martínez-Sánchez, J.; Arias, P. Metrological evaluation of microsoft kinect and asus xtion sensors. *Measurement* **2013**, *46*, 1800–1806.
20. Khoshelham, K.; Elberink, S.O. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* **2012**, *12*, 1437–1454.
21. DiFilippo, N.M.; Jouaneh, M.K. Characterization of different microsoft kinect sensor models. *IEEE Sens. J.* **2015**, *15*, 4554–4564.
22. Stackoverflow Community. Precision of the Kinect Depth Camera. Available online: https://goo.gl/x4jWoX (accessed on 8 December 2017).
23. Freedman, B.; Shpunt, A.; Machline, M.; Yoel, A. Depth Mapping Using Projected Patterns. U.S. Patent 20100118123, 13 May 2010.
24. Shpunt, A. Depth Mapping Using Multi-Beam Illumination. U.S. Patent 20100020078, 28 January 2010.
25. Shpunt, A.; Petah, T.; Zalevsky, Z.; Rosh, H. Depth-Varying Light Fields for Three Dimensional Sensing. U.S. Patent 20080106746, 8 May 2008.
26. Reichinger, A. Kinect Pattern Uncovered. Available online: https://goo.gl/DOqRxZ (accessed on 8 December 2017).
27. Fofi, D.; Sliwa, T.; Voisin, Y. A comparative survey on invisible structured light. *SPIE Mach. Vis. Appl. Ind. Insp. XII* **2004**, 1–90, doi:10.1117/12.525369.
28. Rzeszotarski, D.; Strumiłło, P.; Pełczyński, P.; Więcek, B.; Lorenc, A. System obrazowania stereoskopowego sekwencji scen trójwymiarowych. *Elektronika* **2005**, *10*, 165–184.
29. Shotton, J.; Johnson, M.; Cipolla, R. Semantic texton forests for image categorization and segmentation. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
30. Shotton, J.; Fitzgibbon, A.W.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A. Real-time human pose recognition in parts from single depth images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1297–1304.

31. Asteriadis, S.; Chatzitofis, A.; Zarpalas, D.; Alexiadis, D.S.; Daras, P. Estimating human motion from multiple Kinect sensors. In Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications - MIRAGE '13, Berlin, Germany, 6–7 June 2013; pp. 1–6.
32. Kitsikidis, A.; Dimitropoulos, K.; Douka, S.; Grammalidis, N. Dance analysis using multiple Kinect sensors. In Proceedings of the 2014 International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, 5–8 January 2014; pp. 789–795.
33. Schroder, Y.; Scholz, A.; Berger, K.; Ruhl, K.; Guthe, S.; Magnor, M. Multiple kinect studies. *Comput. Graph. Lab* **2011**, *09-15*, 1–30.
34. Alexiev, K.; Nikolova, I. An algorithm for error reducing in IMU. In Proceedings of the 2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications, Albena, Bulgaria, 19–21 June 2013.
35. The Center for Geotechnical Modeling (CGM) at UC Davis. Signal Processing and Filtering of Raw Accelerometer Records. Available online: https://goo.gl/jQt49h (accessed on 8 December 2017).
36. Wang, W.Z.; Guo, Y.W.; Huang, B.Y.; Zhao, G.R.; Liu, B.Q.; Wang, L. Analysis of filtering methods for 3D acceleration signals in body sensor network. In Proceedings of the 2011 International Symposium on Bioelectronics and Bioinformatics, Suzhou, China, 3–5 November 2011; pp. 263–266.
37. Gebhardt, S.; Scheinert, G.; Uhlmann, F.H. Temperature influence on capacitive sensor structures. In Proceedings of 51st International Scientific Colloquium, Ilmenau, Germany, 11–15 September 2006; pp. 11–15.
38. Grigorie, M.; de Raad, C.; Krummenacher, F.; Enz, C. Analog temperature compensation for capacitive sensor interfaces. In Proceedings of the 22nd European Solid-State Circuits Conference, Neuchatel, Switzerland, 17–19 September 1996; pp. 388–391.
39. Cheung, P. Impulse Response & Digital Filters. Available online: https://goo.gl/GkbGHg (accessed on 8 December 2017).
40. Freescale Semiconductor, Inc. *Allan Variance: Noise Analysis for Gyroscopes*; Freescale Semiconductor, Inc.: Austin, TX, USA, 2015; pp. 1–9.
41. Allan, D.W. Statistics of atomic frequency standards. *Proc. IEEE* **1966**, *54*, 221–230.
42. Allan, D.W. Should the classical variance be used as a basic measure in standards metrology? *IEEE Trans. Instrum. Meas.* **1987**, *IM-36*, 646–654, doi:10.1109/TIM.1987.6312761.
43. Dunn, F.; Parberry, I. Rotation in three dimensions. In *3D Math Primer for Graphics and Game Development*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2011; pp. 217–294.
44. Armstrong, J.S.; Collopy, F. Error measures for generalizing about forecasting methods: Empirical comparisons. *Int. J. Forecast.* **1992**, *8*, 69–80, doi:10.1016/0169-2070(92)90008-W.