Time Synchronization and Data Fusion for RGB-Depth Cameras and Inertial Sensors in AAL Applications

Conference Paper · June 2015

DOI: 10.1109/ICCW.2015.7247189

7 authors, including:

E. Gambi
Università Politecnica delle Marche

Susanna Spinsante
Università Politecnica delle Marche

Ibrahim Orhan
KTH Royal Institute of Technology

Thomas Lindh
KTH Royal Institute of Technology

Time Synchronization and Data Fusion for RGB-Depth Cameras and Inertial Sensors in AAL Applications

Enea Cippitelli*, Samuele Gasparrini*, Ennio Gambi*, Susanna Spinsante*, Jonas Wåhslén†, Ibrahim Orhan† and Thomas Lindh†

*Dipartimento di Ingegneria dell'Informazione, Universita' Politecnica delle Marche, Ancona, Italy I-60131

Email: {e.cippitelli, s.gasparrini, e.gambi, s.spinsante}@univpm.it

†School of Technology and Health, KTH, Royal Institute of Technology, Stockholm, Sweden SE-100 44

Email: {jonas.wahslen, ibrahim.orhan, thomas.lindh}@sth.kth.se

***Abstract*—Ambient Assisted Living applications often need to integrate data from multiple sensors, to provide consistent information on the observed phenomena. Data fusion based on samples from several sensors requires accurate time synchronization with sufficient resolution, depending on the sensor sampling frequency. This work presents a technical platform for the efficient and accurate synchronization of the data captured from RGB-Depth cameras and wearable inertial sensors, that can be integrated in AAL solutions. A case study of sensor data fusion for Timed Up and Go test is also presented and discussed.**

***Keywords*—depth camera, inertial sensor, data fusion, synchronization, timed up and go**

I. Introduction

Ambient Assisted Living (AAL) aims to provide technical solutions to support elderly and/or disabled people in their home environment. One of the components enabling the definition of Enhanced Living Environments is related to sensing and processing of the data collected from the users or the environment where they live.

AAL applications often need to integrate data from multiple sensors. Data fusion based on samples from several sensors requires accurate time synchronization with sufficient resolution, depending on the sampling frequency ([1], [2]) of the sensors used. Gait and balance analysis, fall detection and inactivity alarm functions are examples of AAL services that can benefit from accurate data fusion of e.g. wireless body-worn inertial sensors, and video and depth images (RGB-Depth, or RGB-D). The techniques used to process depth maps represent an alternative to the traditional RGB ones. The availability of low cost depth cameras, appeared in the mass market in the last years, contributes a lot to the development of this specific image processing branch, which makes it possible to collect distance values without the need of a stereoscopic solution. Depth cameras are particularly suitable in AAL ([3], [4]) because they allow to preserve the privacy of the observed subjects.

Nowadays, wearable and vision-based sensors are alternately used in AAL solutions, but their joint use can overcome the limitations of both the technologies. The main drawback of using wearable devices is the strong incidence of false positives; on the other hand, the use of vision-based sensors usually requires complex algorithms to extract significant features from raw data. Data fusion of wearable and vision-based information may improve the performance of the global system.

The use of heterogeneous sensors makes it necessary to synchronize the received signals, in order to provide consistent information on the observed phenomenon. In the literature, this aspect is addressed by means of different approaches. In [5], the NIST Internet Time Service is used to synchronize, at high level, the depth frames and the data packets sent from the wearable sensor to a smartphone, over a Bluetooth connection. Liu *et al.* [6] implement a hand gesture recognition system using a similar setup, provided as input to an hidden Markov model. The proposed synchronization approach correlates in time the closest inertial sample to the depth frame. It does not take into account possible propagation delays and variable intervals needed to generate the samples. In [7], the authors use physical trigger events to synchronize the devices, such as five vertical jumps to generate a peak in each acceleration stream.

The purpose of this work is to develop and implement a time synchronization strategy to enable fusion of a wearable sensor data and RGB-D data. The considered RGB-D sensor is the Microsoft Kinect, as a good compromise between cost, performance and usability. The inertial wearable platform selected in the proposed study is manufactured by Shimmer Research [8] and integrates a 3-axis accelerometer. The main contributions are, firstly, an efficient and accurate method for time synchronization of video/depth camera output and wireless wearable sensor signal; and secondly, a case study of sensor data fusion applied to Timed Up and Go test (TUG). TUG test represents a well known tool in the rehabilitation field, and it is used to evaluate gait and balance capabilities of people, especially older ones ([9], [10]). In the literature, the TUG is implemented using several wearable sensors [11], or a single depth camera [12]. In this work, the gyroscopes and accelerometers adopted in [11] are replaced by a single sensor placed on the subject's chest, thus requiring a much simpler setup; the not perfect estimation of the skeleton provided by Kinect while the subject walks back to the chair [12], is here

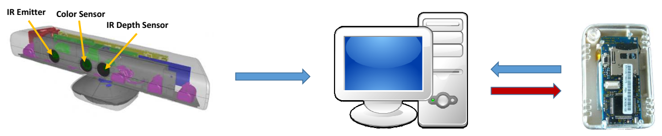


Fig. 1. The testbed consists of a Kinect RGB-D camera connected to a PC, and an IMU (inertial measurement unit) from Shimmer Research connected via Bluetooth to the same PC.

improved, thanks to the use of the acceleration data. Time synchronization between wired and wireless devices has been studied extensively (e.g. in [13], [14], [1], [2]). This paper focuses on the specific problems in time synchronization and data fusion between a video-depth camera and wireless inertial sensors. In previous work, synchronization has not been solved for sensor sampling rates that exceed the camera frame rate and for multiple sensors.

The paper is organized as follows: Section II describes the proposed synchronization technique, whereas its application to TUG and related discussion are provided in Section III. Finally, Section IV concludes the paper.

II. Time Synchronization

The purpose of this activity is to synchronize data samples from multiple sensors in time, i.e. to associate samples from each sensor that are closest to each other in time. Reliable time synchronization is a requirement for data fusion. This section describes the time synchronization method, implementation, testbed and results. The time synchronization should work without having access to the video-depth camera's internal system clock. Hence, new estimation methods have to be developed.

A. Testbed

To implement a synchronization testbed, a PC is connected to a video-depth camera (Kinect) via cable and to a wireless inertial measurement unit (IMU) via Bluetooth, as shown in Fig. 1. The wireless sensor node, from Shimmer Research [8], has an inbuilt three-axis accelerometer. The operating system is TinyOS and the programming language is NesC. The range camera, Kinect v1 and v2 for Windows, provides RGB, depth and infrared streams at 30 Hz frame rate. The Microsoft SDK for Kinect v2 is able to estimate 25 joints representing the skeleton of a person. This information can be used to estimate positions and to track movements. These estimates can be improved and augmented by combining the camera output with data from the wearable accelerometer. The Shimmer sensor node and the PC are synchronized using a variant of Cristian's algorithm ([2], [15]). However, the video-depth camera's internal system clock is not accessible, and can therefore not be synchronized with the same algorithm. An Arduino board, controlling seven LEDs, is connected to the PC to verify the estimation of exposure and transmission times (see Section II-B). The round-trip time between the PC and the Arduino serial interface has been estimated over 1500 transmissions, thus providing an average value of 3.2 milliseconds. The Arduino board is also synchronized to the PC with the previous algorithm ([2], [15]).

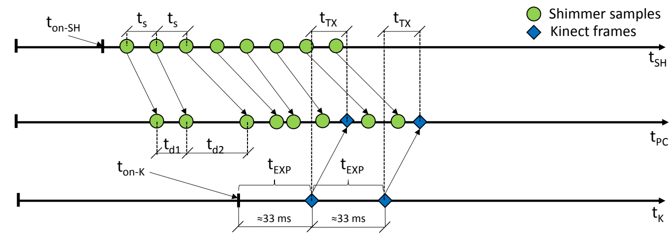


Fig. 2. Time axes and delays between the PC, the Shimmer node (accelerometer), and the Kinect camera.

B. Method

Time synchronization in this case involves two issues: synchronizing the PC and the Shimmer node communicating via Bluetooth, and synchronizing the Kinect sensor and the PC. The different time axes of the Shimmer sensor (t_{SH}), the PC (t_{PC}), and Kinect (t_K) are shown in Fig. 2. The non-trivial problem of synchronizing devices connected via Bluetooth (the PC and the Shimmer node) has been solved previously ([2], [15]), and such a solution is adapted and reused in this work. The main problem is therefore to synchronize the RGB-D camera and the PC. Since it is impossible to access and control the Kinect embedded camera clock, the approach is to estimate the total delay occurring in the link between the camera and the PC, and to exploit appropriate C++ functions on the PC (*QueryPerformanceCounter* and *QueryPerformanceFrequency*) for data timestamping. The total delay between Kinect and the PC can be split into two parts (see Fig. 2): the exposure time t_{EXP}, which is the amount of time the sensing device is *open*; and the transmission time t_{TX}, which is the time needed to encapsulate and transmit the frame from the camera to the PC. The amount of time needed by the Shimmer node to read data samples from the accelerometer and perform A/D conversion (the sample and hold time) is four clock cycles. Shimmer is equipped with a 32.768 kHz crystal, which means a total data acquisition time below 0.13 milliseconds. For a sampling frequency of e.g. 500 Hz, the acquisition time is less than one tenth the sampling time. However, this is not the case when sampling a frame from a camera such as Kinect. The acquisition time (i.e. the exposure time) is much longer compared to the sampling time of the IMU.

The estimation of the transmission time for the RGB frames has been performed by using an Arduino board, connected to the same PC as the Kinect camera, which controls seven LEDs. Fig. 3 shows the implemented procedure to estimate the transmission time of RGB frames of Kinect v1 and v2, according to the steps detailed below:

i. The PC receives frame F0, sets a timestamp (t_{0_PC}) and sends a command to the Arduino board.

ii. The Arduino board receives the command, waits for 20 milliseconds, and sequentially switches on the LEDs with a delay of 3 milliseconds.

iii. When frame F2 arrives at the PC, timestamp (t_{2_PC}) is set and, by counting the number of LEDs that are ON in that frame, it is possible to calculate the time t_{2_K} with an uncertainty of 1.5 milliseconds.

iv. The difference between t_{2_PC} and t_{2_K} gives the transmission time for the frame F2.

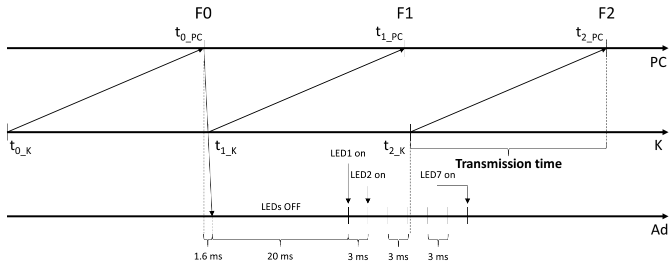


Fig. 3. Evaluation of the RGB transmission time for Kinect v2.

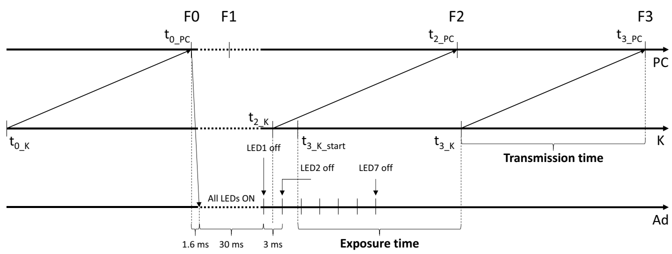


Fig. 4. Evaluation of the RGB exposure time for Kinect v2.

The 20 ms delay allows to center the on/off leds sequence at the t_{2_K} time. If this waiting time is not used, more than seven leds would be required to estimate the transmission time with the same uncertainty. This procedure, for the Kinect v2, has been repeated 75 times, providing an average transmission time of 31.5 milliseconds and a standard deviation of 1.1 milliseconds.

A similar procedure has been implemented to evaluate the transmission time of the IR frame. IR LEDs operating at 830 nm wavelength, that can be detected by the Kinect IR sensor, are used. This approach has been applied to estimate the transmission time of IR and depth frames for Kinect v1, and of IR frames for Kinect v2. The same approach cannot be applied to the Kinect v2 depth stream, as the same LEDs cannot be detected by the depth sensor and are not shown in the depth frame. In this case, a moving object in the Kinect coverage area can be used to estimate additional delays between IR and depth frames, for the same displacement.

The exposure time of the RGB frame of Kinect v2 and RGB-IR-Depth frames of Kinect v1 is estimated according to the steps detailed below (see Fig. 4). The Arduino board controlling seven LEDs is used.

i. The PC receives the frame F0, sets a timestamp (t_{0_PC}) and sends a command to the Arduino board.
ii. The Arduino board receives the command, switches on all the 7 LEDs and waits for 30 milliseconds; then it sequentially switches the LEDs OFF with a delay of 3 milliseconds.
iii. When frame F3 arrives at the PC, the timestamp t_{3_PC} is set and, by counting the number of LEDs which appear ON in that frame, it is possible to calculate the time $t_{3\_K\_start}$ with an uncertainty of 1.5 milliseconds.
iv. It is also possible to calculate the time t_{3_K} as the difference between t_{3_PC} and the transmission time.
v. The exposure time is finally derived from the [$t_{3\_K\_start}$, t_{3_K}] interval.

The average exposure time for RGB frames of Kinect v2, evaluated over 75 frames, is 28.5 milliseconds. The standard deviation is 1.2 millisecond. An additional uncertainty of 1.5 milliseconds, due to the interval of 3 milliseconds between two consecutive LEDs switching off events, should be considered.

A different approach has been applied to evaluate the exposure time of IR frames for Kinect v2, which is much lower than the RGB frames exposure time. The Arduino board controls seven IR LEDs (830 nm), switching between ON and OFF by 1 millisecond intervals. The exposure time can be estimated by counting the number of LEDs that appear ON in a single frame. The evaluated exposure time for IR frames is 3 ± 1.2 milliseconds.

C. Results

Verification tests for the Kinect v1 have been carried out on a laptop Intel core 2 Duo @ 2.53 GHz, 4 GB RAM, Windows 7 operating system and C++ software. The total number of useful frames for each test is 200. Table I shows the resulting exposure and transmission times for each available data stream. The transfer of depth frame requires a higher transmission time than the other two streams. This is explained by the processing time needed to obtain the depth information from the IR frame, by using the structured light technique. The exposure time is approximately the same for all the three streams.

Verification tests for Kinect v2 have been carried out on a desktop PC with Intel i7 @ 3.5 GHz, 16 GB RAM, Windows 8.1 operating system and C++ software. Table II shows that the transmission time for the RGB frames is approximately twice the transmission time for IR and depth frames. The RGB frame resolution is 1920×1080 pixels, whereas the IR frame and depth frame resolution is 512×424 pixels. The IR exposure time is lower than the RGB exposure time.

D. Error Analysis

The synchronization procedure between Kinect and the Shimmer sensor node consists of three steps, detailed below.

i. The synchronization algorithm in [2] is applied to the communication between the PC and the Shimmer node.
ii. Time compensation of the frames arrived at the PC from the Kinect camera is performed.

TABLE I. KINECT V1 EXPOSURE AND TRANSMISSION TIMES (95% CONFIDENCE LEVEL).

Data stream	Exposure time [ms]	Transmission time [ms]
RGB	28.4 ± 2.0	15.0 ± 2.2
IR	31.3 ± 2.2	16.4 ± 2.0
Depth	29.1 ± 6.3	28.2 ± 5.7

TABLE II. KINECT V2 EXPOSURE AND TRANSMISSION TIMES (95% CONFIDENCE LEVEL).

Data stream	Exposure time [ms]	Transmission time [ms]
RGB	28.5 ± 1.2	31.5 ± 1.1
IR	3.0 ± 1.2	16.0 ± 1.0
Depth	3.0 ± 1.2	18.5 ± 1.0

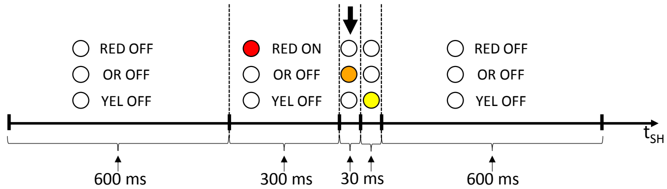


Fig. 5. LED pattern used to verify the synchronization algorithm.

iii. The Shimmer packets closest to each frame are determined and associated, by taking into account the estimated exposure time.

By applying the procedure detailed above, the accelerometer data sampled from the Shimmer node, and the RGB-D frames provided by Kinect can be synchronized.

The accuracy of the transmission and exposure times estimation, and the results obtained from the proposed synchronization approach are verified in the following way. The Shimmer device has three programmable LEDs. The LED ON events, generated by the Shimmer, will be captured by the Kinect camera.

The ON/OFF LED sequences on the Shimmer node are shown in Fig. 5. The node has been programmed to send a dedicated packet when the orange LED only is on. The dedicated packet is received by the PC and the synchronization algorithm is performed, to determine which of the RGB frames received by the PC is closest in time to the received dedicated packet. If the RGB frame is selected properly, the image in such a frame should display the Shimmer device with its orange LED ON. This testing procedure has been repeated 48 times for Kinect v2. The correct frame has been detected in 45 out of the 48 events (93.8%). Furthermore, the maximum estimation error between a sample and a frame is 1 (one) frame only. When using the Kinect v1 sensor, the corresponding results are a 99.4% correct estimation rate, over 180 trials, and a maximum error of one frame.

III. Sample Application: TUG

The TUG test consists in having a subject sitting on a chair, who stands up and starts walking on a straight line; after three meters the subject turns and walks back towards the chair. The test ends when the subject sits. The Kinect v2 sensor and the Shimmer device may be used to compute several parameters to describe the movement. Such parameters may help health care personnel to objectively evaluate the balance capabilities of the person and, most of all, to have an unbiased comparison of different test executions from the same subject, in different times.

A. TUG Parameters Extraction

In order to provide a test that can be performed and repeated in a home environment, the position of the Kinect sensor should be in front of the person, at a reasonable distance and height. According to this requirement, the sensor is positioned at 1.5 meters from the floor, slightly tilted downward. The distance between the sensor and the chair used for the test should be around 3 meters, to avoid large errors in the estimation of the skeleton by Kinect. The Shimmer device

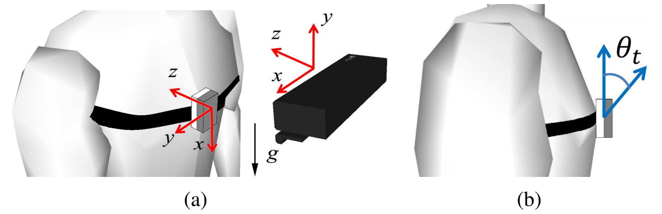


Fig. 6. Setup of sensors: (a) setup of inertial and Kinect v2 sensors, (b) orientation of torso's angle θ_t

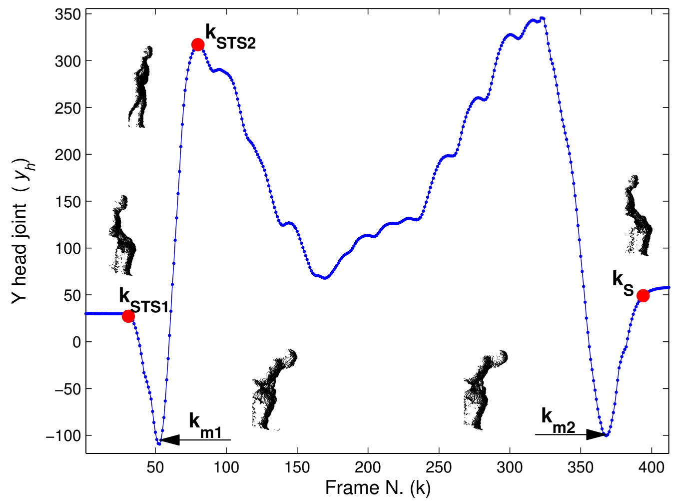


Fig. 7. Head joint trajectory along the y axis during a complete TUG test.

is positioned on the chest of the person, because it can provide information regarding the inclination of the torso during the sit-to-stand phase, which is an important parameter to be estimated ([16], [17]). In particular, the following parameters are evaluated: inclination of the torso (θ_t), time duration of the sit-to-stand phase (t_{STS}), time duration of the turning phase (t_T), time duration of the sit phase (t_S), total time duration of the test (t_{TUG}), length and cadence of the steps (t_{SP}), angular velocity of arm-swing (ω_A).

Fig. 6 shows the setup of inertial and Kinect sensors during the TUG test. The inertial device is constrained to the body of the person by means of a chest strap, and the Kinect v2 is facing the person, with the z-axis oriented in the same direction.

The searched θ_t value corresponds to the maximum inclination of the torso during the sit-to-stand phase. It is derived from the elaboration of the acceleration data provided by the inertial sensor. Starting from the raw data, it is necessary to compensate the bias typical of a real IMU device, identified by testing the 1 g orientation for each axis. Let X, Y and Z be the acceleration data, respectively along x, y and z axis, then θ_t (in degrees) is given by (1):

$$\theta_t = \max\left\{90 - \tan^{-1}\left(\frac{X}{\sqrt{Y^2+Z^2}}\right)\right\} \tag{1}$$

The identification of the different test phases is performed by exploiting the Kinect skeleton frames. The k_{STS1} is the frame in which the person begins to move. The identification of this frame is obtained from the y trajectory of the head skeleton joint. The next frame of interest is the k_{STS2} frame, where the person has finished the sit-to-stand phase and starts walking. Considering a TUG execution, composed by K skeleton frames, and defining as $y_h(k)$ the y-coordinate of the head joint in the k-th frame, the k_{STS2} frame in Fig. 7 must satisfy the relation (2):

$$y_h(k_{STS2}) = \max\{y_h(k)\}, \quad k = 1, 2, \ldots, K/2 \tag{2}$$

The evaluation of the k_{STS1} frame is obtained by the identification of the k_{m1} frame, as stated in (3):

$$y_h(k_{m1}) = \min\{y_h(k)\}, \quad k = 1, 2, \ldots, k_{STS2} \tag{3}$$

and k_{STS1} is defined as the first frame satisfying the following condition:

$$|y_h(k) - y_h(k-1)| > Th_{1\%}, \quad k = 2, \ldots, k_{m1} \tag{4}$$

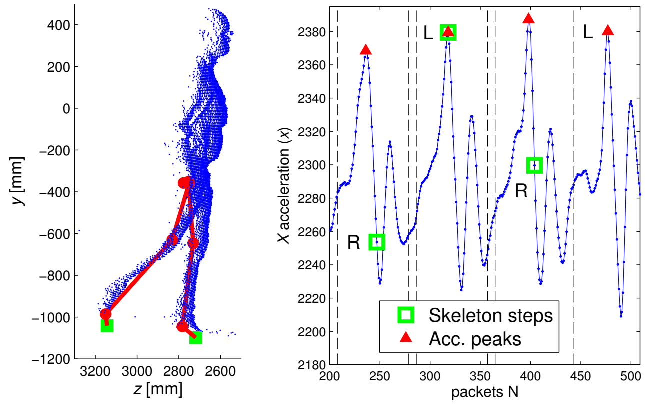
where $Th_{1\%} = |y_h(1) - y_h(k_{m1})|/100$. The time required to complete the sit-to-stand phase is finally computed as:

$$t_{STS} = (k_{STS2} - k_{STS1})\Delta_t \tag{5}$$

where Δ_t represents the time difference between two consecutive skeleton frames.

The turning phase can be identified by exploiting the shoulders and head joints. By combining these three points, it is possible to identify the orientation vector associated to the person. The event of turning is recognized from the angle (θ_s) between the orientation vector and the reference direction, which is parallel to the Kinect z axis in Fig. 6. Another option could be the use of thresholds on the speed of turning [18], but this approach would be affected by differences in young and healthy people, with respect to elderly. Start and stop frames of the turning phase are used to evaluate the t_T, according to (5). The proposed technique to evaluate the turning events, which relies on the θ_s values, is used also in the final part of the test, when the patient arrives in front of the chair and turns to sit down. The turning frame and the k_S frame, extracted through the same approach of k_{STS1}, are used to compute the time duration of the sit phase (t_S). Once the k_{STS1} and k_S frames are known, the total time duration of the TUG test (t_{TUG}) can be computed by applying the same formula in (5) to t_{STS}.

The identification of test phases is exploited to select the intervals to estimate the steps' cadence and the arms' swing velocity. These parameters are both evaluated in the walking phase, which goes from the k_{STS2} frame to the start frame of turning before sitting. During the first walking interval, when the person is moving towards the sensor, the cadence is initially derived from the skeleton data. Then, by exploiting the acceleration data along the x axis, it is possible to improve the results. The steps are first identified through the maximum distance among the feet along the Kinect z coordinate. Fig. 8(a) shows one of these cases: the person walks towards the sensor and the lower limbs are depicted. The squared joints represent the feet, and, in this case, the step length is approximately 400 mm. The skeleton frames featuring the steps are mapped into the acceleration data, as shown in Fig. 8(b). The acceleration values corresponding to the skeleton steps are represented by empty squares, and they are very close in time to the acceleration peaks provided by the inertial sensor. In particular, the second step identified by Kinect matches



(a) (b)

Fig. 8. (a) Side view point cloud of the subject: lower limbs joints are evidenced. (b) Acceleration data along the x axis: skeleton steps and acceleration peaks evidenced.

exactly the acceleration peak detected by the Shimmer device. The final steps correspond to the acceleration peaks within the intervals bounded by vertical dashed lines. The width of the search interval corresponds to the average time distance of the skeleton steps. Regarding the fourth acceleration peak, there is no any equivalent step provided by the skeleton data. In this case the person is very close to the Kinect sensor and the feet are outside the field of view of the device. The algorithm is able to take into account this condition, by scanning the entire x accelerometer trajectory until the turning phase, to find missed steps. The peaks corresponding to steps have to satisfy amplitude and time properties similar to the previously identified ones.

The algorithm performs the cadence analysis also when the person walks towards the chair. During the walking phase, the skeleton data are not reliable and the peaks corresponding to steps are detected by exploiting only the wearable sensor. This operation is very similar to the process applied to identify the fourth step in Fig. 8(b). Finally, each peak must be associated to the corresponding foot. It means that the algorithm exploits the skeleton position to recognize which foot, for each peak, has the closest z component to the sensor. As shown in Fig. 8(b), the first skeleton peak is labeled as **R**, this means that the peak is generated by the right foot. About the steps obtained from the accelerometer data only (4-th triangle in Fig. 8(b)), the algorithm uses the opposite value of the previous peak.

The time indices of the cadence phase are exploited to set the intervals in which the angular velocity of the arm-swing is estimated. It is obtained from the angle composed by the wrist/shoulder joints position at a generic frame k, and the position of the same wrist at frame $k-1$. These three points form an angle centered in the shoulder joint, that changes according to the arm movements. Let $\mathbf{j_s}(k) = [j_{sx}(k), j_{sy}(k), j_{sz}(k)]$ be the vector containing the coordinates of the shoulder joint at k-th skeleton frame, and $\mathbf{j_w}(k)$ the equivalent vector for the wrist joint. It is possible to define two difference vectors:

$$\mathbf{j_{sw}}(k) = \mathbf{j_s}(k) - \mathbf{j_w}(k) \tag{6}$$

$$\mathbf{j_{sw}}(k-1) = \mathbf{j_s}(k) - \mathbf{j_w}(k-1) \tag{7}$$

TABLE III. TUG PARAMETER AVERAGE VALUES AND CONFIDENCE INTERVALS (95% CONFIDENCE LEVEL).

Index		Average value	Minimum	Maximum
θ_t	[deg]	45.3 ± 13.53	18.0	64.0
t_{TUG}	[s]	9.99 ± 1.88	7.00	13.73
t_{STS}	[s]	1.47 ± 0.41	0.96	2.41
t_S	[s]	2.42 ± 0.54	1.42	3.23
t_T	[s]	1.54 ± 0.59	0.66	3.46
t_{SP}	[s]	1.38 ± 0.21	1.12	2.37
$Stride\ length$	[cm]	956 ± 147	743	1295
ω_A	[deg/s]	107 ± 21	67	158

The angular velocity of the arm $\omega_A(k)$ at k-th frame is defined as:

$$\omega_A(k) = \frac{1}{\Delta_t} \cos^{-1}\left(\frac{\mathbf{j_{sw}}(k)}{\|\mathbf{j_{sw}}(k)\|} \cdot \frac{\mathbf{j_{sw}}(k-1)}{\|\mathbf{j_{sw}}(k-1)\|}\right) \tag{8}$$

The output parameter is represented by the average velocity for each gait cycle.

B. Analysis and Discussion

Tests have been performed by 20 healthy subjects, aged between 22 and 39, with different build and height; getting three realizations for each subject, a total number of 60 tests have been collected. Table III summarizes the results obtained over the entire test set. Since all the tests have been performed by healthy people, the obtained values are quite similar, with limited standard deviations. The average time required to complete the TUG test (t_{TUG}) is 10 seconds, while the sit-to-stand phase is completed in about 1.5 seconds. The t_S index, which represents the time to complete the sit-down movement, is higher than in the previous sit-to-stand phase, because it includes also the turning movement. The turning movement alone is completed in 1.54 seconds on average. The indices t_{SP} and $Stride\ length$ refer to the first completed gait cycle, and represent the time required to complete it, and the distance covered, respectively. Finally, parameter ω_A gives the angular velocity of the left arm, during the first gait cycle.

IV. Conclusion

The main contributions of this paper are, firstly; development, implementation and error analysis of efficient and accurate methods for time synchronization of video/depth camera output and external wireless sensors, and secondly; an AAL application case study using the Timed Up and Go test. A video-depth camera output is augmented by data from an accelerometer. The paper shows that accurate results can be obtained, with very low errors, using low-cost consumer products without having access to the video-depth cameras internal system clock. The sensor fusion case study clearly shows the benefit of combining RGB-D camera and accelerometer data. In this work, a single wearable sensor is combined with a camera. If needed, the algorithm can be extended to multiple wearable wireless sensors, which is shown in [2]. This technology can improve the quality of life of elderly people by enabling the TUG assessment at home, thus avoiding the need to reach the doctor's ambulatory. Future work includes testing the system with elders to evaluate the performances in a real case scenario, and extending the applications to e.g. fall detection and balance assessment.

Acknowledgment

This work was supported by STSM Grants from COST Action IC1303 *Algorithms, Architectures and Platforms for Enhanced Living Environments* (AAPELE).

References

[1] J. Wåhslén, I. Orhan, T. Lindh, and M. Eriksson, "A novel approach to multi–sensor data synchronisation using mobile phones," *International Journal of Autonomous and Adaptive Communications Systems (IJAACS)*, vol. 6, no. 3, 2011.

[2] J. Wåhslén, I. Orhan, and T. Lindh, "Local time synchronization in bluetooth piconets for data fusion using mobile phones," in *Body Sensor Networks (BSN), 2011 International Conference on*, May 2011, pp. 133–138.

[3] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi, "A depth-based fall detection system using a kinect® sensor," *Sensors*, vol. 14, no. 2, pp. 2756–2775, 2014.

[4] E. Cippitelli, S. Gasparrini, S. Spinsante, and E. Gambi, "Kinect as a tool for gait analysis: Validation of a real-time joint extraction algorithm working in side view," *Sensors*, vol. 15, no. 1, pp. 1417–1434, 2015.

[5] À. Pardo, A. Clapés, S. Escalera, and O. Pujol, "Actions in context: System for people with dementia," in *Citizen in Sensor Networks*. Springer, 2014, pp. 3–14.

[6] K. Liu, C. Chen, R. Jafari, and N. Kehtarnavaz, "Fusion of inertial and depth sensor data for robust hand gesture recognition," *Sensors Journal, IEEE*, vol. 14, no. 6, pp. 1898–1903, June 2014.

[7] F. Destelle, A. Ahmadi, N. 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 *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, Sept 2014, pp. 371–375.

[8] A. Burns, B. R. Greene, M. J. McGrath, T. J. O'Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, "SHIMMER™– a wireless sensor platform for noninvasive biomedical research," *Sensors Journal, IEEE*, vol. 10, no. 9, pp. 1527–1534, 2010.

[9] B. R. Greene, A. O. Donovan, R. Romero-Ortuno, L. Cogan, C. N. Scanaill, and R. A. Kenny, "Quantitative falls risk assessment using the timed up and go test," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 12, pp. 2918–2926, 2010.

[10] V. Weerdesteyn, M. de Niet, H. J. van Duijnhoven, and A. C. Geurts, "Falls in individuals with stroke," *Journal of rehabilitation research and development*, vol. 45, no. 8, pp. 1195–1214, 2008.

[11] A. Salarian, F. B. Horak, C. Zampieri, P. Carlson-Kuhta, J. G. Nutt, and K. Aminian, "iTUG, a sensitive and reliable measure of mobility," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 3, pp. 303–310, 2010.

[12] B. Kargar, A. Mollahosseini, T. Struemph, W. Pace, R. Nielsen, and M. Mahoor, "Automatic measurement of physical mobility in get-up-and-go test using kinect sensor," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, Aug 2014, pp. 3492–3495.

[13] N. Kaempchen and K. C. Dietmayer, "Data synchronization strategies for multi-sensor fusion," in *Proceeding of ITS 2003, 10-th World Congress of Intelligent Transport Systems*, 2003.

[14] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed systems: concepts and design*, 4th ed. Addison-Wesley, 2006.

[15] F. Cristian, "Probabilistic clock synchronization," *Distributed computing*, vol. 3, no. 3, pp. 146–158, 1989.

[16] E. Papa and A. Cappozzo, "Sit-to-stand motor strategies investigated in able-bodied young and elderly subjects," *Journal of Biomechanics*, vol. 33, no. 9, pp. 1113–1122, 2000.

[17] K. Kerr, J. White, D. Barr, and R. Mollan, "Analysis of the sit-stand-sit movement cycle in normal subjects," *Clinical Biomechanics*, vol. 12, no. 4, pp. 236–245, 1997.

[18] Y. Higashi, K. Yamakoshi, T. Fujimoto, M. Sekine, and T. Tamura, "Quantitative evaluation of movement using the timed up-and-go test," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 27, no. 4, pp. 38–46, 2008.