

Article

# Successive Collaborative SLAM: Towards Reliable Inertial Pedestrian Navigation

Susanna Kaiser

Institute for Communications and Navigation, German Aerospace Center (DLR), Oberpfaffenhofen, D-82234 Wessling, Germany; susanna.kaiser@dlr.de; Tel.: +49-8153-282862

Received: 21 August 2020; Accepted: 28 September 2020; Published: 30 September 2020



**Abstract:** In emergency scenarios, such as a terrorist attack or a building on fire, it is desirable to track first responders in order to coordinate the operation. Pedestrian tracking methods solely based on inertial measurement units in indoor environments are candidates for such operations since they do not depend on pre-installed infrastructure. A very powerful indoor navigation method represents collaborative simultaneous localization and mapping (collaborative SLAM), where the learned maps of several users can be combined in order to help indoor positioning. In this paper, maps are estimated from several similar trajectories (multiple users) or one user wearing multiple sensors. They are combined successively in order to obtain a precise map and positioning. For reducing complexity, the trajectories are divided into small portions (sliding window technique) and are partly successively applied to the collaborative SLAM algorithm. We investigate successive combinations of the map portions of several pedestrians and analyze the resulting position accuracy. The results depend on several parameters, e.g., the number of users or sensors, the sensor drifts, the amount of revisited area, the number of iterations, and the windows size. We provide a discussion about the choice of the parameters. The results show that the mean position error can be reduced to  $\approx 0.5$  m when applying partly successive collaborative SLAM.

**Keywords:** indoor navigation; pedestrian dead reckoning; simultaneous localization and mapping; FootSLAM; collaborative mapping; professional use cases

## 1. Introduction

In indoor emergency scenarios, the first responders usually arrive by emergency vehicles and small teams of them enter a building without knowing the floorplan. At the same time, the commander of the first responders on sight needs to know the positions of the first responders inside the building to coordinate their actions. Examples for such scenarios are a building on fire or policemen operations in the case of a hostage-taken, a rampage, or a terror attack. In addition to the coordination of the team members, the knowledge of the positions and the activities of the first responders are additionally desired for the case that a first responder needs help or needs to be rescued. For instance, in the case of a fire, this information would help if an injured firefighter in operation needs assistance. With the knowledge of the position, the injured firefighter can easily and quickly be found. For the above-mentioned applications in professional use cases, an infrastructure-less, ad-hoc applicable indoor navigation system has to be developed.

Indoor pedestrian navigation that cannot rely on signals of opportunity like wireless fidelity (WiFi), iBeacons, ultra-wideband (UWB), or radio frequency identification (RFID) is still challenging [1–4]. Inertial navigation is a promising solution; but, it suffers from a remaining drift due to integration of sensor measurement errors. In widespread scenarios like navigation in shopping malls and airports, the above-mentioned infrastructure for indoor navigation might be available; but, this surely will not be the case consistently in every building and especially not in tunnels or mines. In addition,

depending on the operator of the building different infrastructure solutions in different buildings will not ease the indoor navigation problem. Therefore, techniques that are not dependent on pre-installed infrastructure and which are able to provide accurate positioning in all situations are preferable.

In emergency situations, after arriving, several emergency forces enter the building usually in groups. These groups follow a special formation like it is the case for special police forces who enter the building in groups of 6–8 persons in different (mostly confident) formations depending on the situation in the crime. In this special case, the group usually stays together when entering the building and they walk a similar path until they get another order. The focus of the paper is collaborative simultaneous localization and mapping (SLAM) which makes use of the fact that several pedestrians follow similar paths for learning a map of the environment and for simultaneous positioning. In our collaborative SLAM approach, the maps are collected, combined iteratively, and used as prior information for better positioning.

Instead of collecting the maps from different users, we can also mount multiple sensors on one pedestrian (or emergency force) and calculate the (different) trajectories of multiple sensors. The trajectories differ due to the varying drifts of the sensors depending on the actual measurement errors. It was observed that the drifts of the sensors vary in a similar way as it is the case for different walks. From these trajectories, we can estimate individual maps and combine them with collaborative SLAM. The advantage of this is that we surely know that the trajectories should be the same.

In this paper, we investigate collaborative SLAM solely based on inertial data either resulting from a number of pedestrians (or rescue personnel) that are entering an unknown building and that are following similar paths—or resulting from one pedestrian (or emergency force) wearing multiple sensors. The collaborative SLAM applied in this paper is the so called FeetSLAM algorithm [5,6]. The FeetSLAM algorithm is applied in a new partly successive way which enables a fast combination of small portions of the walks, and reduces the complexity of the successive FeetSLAM algorithm [7]. We analyze the position accuracy depending on different parameters like the number of users, the number of iterations, and the window size. We show that the position accuracy is enhanced from the beginning on and we do not necessarily need a loop closure. Without a loop closure or a revisited area, the position accuracy is usually poor in SLAM algorithms. It should be noted that the communication part and the starting position determination is not a focus of the paper and is foreseen for future research.

The paper is organized as follows: After describing a review of the state-of-the-art of collaborative SLAM in Section 2, the pedestrian navigation system is introduced in Section 3. In this section, first the single-user pedestrian dead reckoning (PDR) and the cascaded SLAM systems are shortly presented followed by the description of the collaborative, multi-user FeetSLAM. In Section 4, the new partly successive collaborative FeetSLAM algorithm is explained. The conducted experiments and their results are described in Section 5. Finally, Section 6 is devoted to conclusions and presents an overview of future work.

## 2. Collaborative Mapping: State-of-the-Art

The varying solutions for collaborative SLAM that can be found in the literature mainly differ depending on the type of sensors used and on the application foreseen. In the field of robotics, the idea of collaborative mapping already came up in the last decade of the last century [8]. In the meantime, the SLAM algorithm was developed [9], and with it the possibility of merging the maps from multi-robots/agents, that is a straightforward continuation but is not yet fully investigated. In [10], five different multi-robot SLAM solutions are examined and compared. Different SLAM solutions are presented in [11], where a comprehensive review of the state-of-the-art on multi-robot SLAM is given. Here, the problems of multi-robot SLAM are highlighted considering also communication, representation, and map merging. A first collaborative SLAM with multiple cameras moving independently in a dynamic environment is presented in [12] in the field of visual sensor localization [13]. In this work, the map includes 3D positions of static background points and

trajectories of moving foreground points. The position uncertainty of each map point is stored in the map. The authors of [14] extended this visual SLAM system with inertial data. In addition, a client–server structure for collaborative SLAM with visual-inertial data and monocular data are examined.

In [15], a collaborative system for unmanned flight vehicles is proposed. Here, the authors apply an extended Kalman filter SLAM (EKF-SLAM) algorithm using different kinds of sensors such as inertial, global positioning system (GPS), vision, and radar/laser sensors, and exchange the map in a decentralized way. In a decentralized system, an efficient map representation is a main requirement because usually the communication bandwidth is limited. In [16], the RFID memory is exploited for exchanging information in a decentralized SLAM for pedestrians using PDR and RFID techniques.

A more general survey on the current state, different aspects, and the future directions of SLAM techniques can be found in [17]. In addition to a comprehensive review of the state-of-the-art in SLAM methods, it provides also a review of the state-of-the-art in collaborative mapping. The authors claim that the SLAM method chosen is dependent on the application and that besides pure place recognition or landmark detection, additional metric information makes the system more robust. This makes the SLAM algorithm itself reasonable in unknown areas.

In this work, we take into consideration a collaborative SLAM technique that is designed for pedestrians and especially useful for emergency forces. This system works also in global navigation satellite system (GNSS) denied, infrastructure-less indoor areas. The references in the literature mainly deal with collaborative mapping for robots, vehicles, or unmanned aerial vehicles, but not for pedestrian navigation in professional use cases. The only reference that deals directly with pedestrians is the already mentioned decentralized collaborative SLAM proposed in [16], which is not infrastructure-less. Camera solutions are not appropriate in the case of fire because of the dusty environment, and lidar and radar solutions need to scan the whole environment in advance and usually represent a higher computational effort. Therefore, we apply in this paper a system where the pedestrian only needs to wear one or several inertial measurement unit (IMU) devices for determining the pedestrian's position. This system does not exclude the use of other sensors like e.g., lidar or radar, or signals of opportunity, because it is already a sensor fusion approach.

Since the positioning with inertial data results in a relative position, the starting conditions (SCs) need to be known. For instance, they can be obtained via GNSS data or a UWB system outside the building, e.g., mounted stationary on the truck. Emergency forces arrive usually by emergency vehicles; therefore, the assumption of entering the building from outside holds for them in any case.

In this paper, we additionally investigate the results for multiple sensors worn by a single user instead of combining the trajectories of multiple users. Many studies on using sensor arrays for getting better performance can be found in the literature. In [18,19], a review of the state-of-the-art in the field of sensor arrays and a maximum likelihood based measurement fusion method including performance evaluation via the Cramér–Rao bound is elaborated. In contrast to the works on sensor arrays, where usually raw data fusion is considered, the sensor errors are analyzed, a better calibration is achieved, and the sensor constellation is optimized. We combine the already calculated trajectories (the outcome of the PDR) in an efficient, collaborative way in order to reduce the remaining drift. This does not exclude enhancements of the underneath PDR—which surely enhances also the upper SLAM. The aim of the work was reducing the remaining drift (that exists also after enhancing the underneath PDR) in a collaborative way in order to assure the final position estimates.

Different from SLAM methods given in the literature, the FeetSLAM-algorithm, which is used in this paper and which builds upon FootSLAM [20], is based on a hexagonal probabilistic transition map. The advantage of using a hexagonal map is that the map representation is very effective. In [21], it has been estimated that, when using a FootSLAM transition map representation, a building with a floor surface of  $10^4 \text{ m}^2$  needs only 40 kB memory and the indoor areas of the world would need 200 gigabytes of memory. Efficient map representation especially for collaborative visual SLAM is an

issue elaborated in the literature [22] and should be considered when designing a collaborative SLAM system having in mind the data amount to be transmitted.

Another advantage of FootSLAM is the possibility of handling dynamic objects. In [23], the authors distinguish between static and dynamic objects in order to aid their system. In FootSLAM, the map is based on a probabilistic map, where the hexagon transitions are counted. Whenever a pedestrian is hindering the movement of a pedestrian only temporarily, another—but possible—transition is counted. When using the walks of many pedestrians inside the building together or measuring a long time for building the map, the transitions in the temporarily occupied area are nevertheless counted and probably counted more often so that the dynamic object—which is the hindering pedestrian—does not have a strong influence on the map. For example, when a group of pedestrians enters the building, having to sidestep at different locations for instance because of an escaping person, FootSLAM will nevertheless learn the main path through the corridor, if the remaining pedestrians pass the main path. The learned map can then be used for localizing further pedestrians like another group of emergency forces entering the building. FootSLAM also observes changes in the room structure in a building. The algorithm learns the transitions, and whenever a transition is new or not counted anymore, the algorithm will learn the new situation. Contrary to using available floorplans for restricting the movement of the pedestrian, that should always be up-to-date, FootSLAM will learn the map even when there are changes in the map.

### 3. Overview of the System

#### 3.1. Pedestrian Dead Reckoning (PDR)

The trajectory of a single user is estimated from inertial data via a zero velocity update (ZUPT) aided unscented Kalman filter (UKF) [24]. The measurement data coming from the IMU, namely 3D accelerometer and 3D gyroscope data, are used as input to the UKF, which estimates 15 states: orientation, position, velocity, and gyroscope biases and accelerometer biases (each of them in three dimensions). In order to obtain the orientation in navigation frame, a strapdown algorithm is applied. The attitude estimation from the UKF is fed back to it for orientation correction. The PDR filter can also be exchanged by another PDR system; for instance, a pocket navigation system [25]. The 100 Hz measurement data (3D accelerometer and 3D gyroscope data) can be reduced to 1 Hz position and orientation data which are enough for the following SLAM algorithm to estimate a map of walkable areas and to enhance positioning. In our approach, we send the data when a certain distance (e.g., 1 m is usually enough for FootSLAM) is reached, but also update the positions every second during standing. Please refer to [24] for a comprehensive description of the PDR applied in this paper.

#### 3.2. Simultaneous Localization and Mapping (SLAM)

The positions and orientations from the PDR are fed to the FootSLAM algorithm. FootSLAM estimates a hexagonal transition map during the walk indicating walkable areas. The FootSLAM algorithm is not restricted to sensors mounted on the foot. It can handle a drifted trajectory from a sensor at any location of the body. In [26], it has been shown that the algorithm is also applicable to sensors mounted in the pocket.

In FootSLAM, each edge of a hexagon of the hexagonal grid map represents a transition counter. During the walk, the edge transitions are counted and the transition counts are used for weighting the particles according to the estimated transition map. FootSLAM is based on a Rao blackwellized particle filter (RBPF) which leaves open the opportunity to fuse other sensors, if available. A comprehensive description of FootSLAM can be found in [20].

#### 3.3. Collaborative SLAM: FeetSLAM

In FeetSLAM, the outputs of several PDRs are collected and combined based on a collaborative map learning process in order to obtain a better map for the next walk through the building. This can

be done in an iterative way [5,6], which is shortly recapitulated next. Comprehensive information on FeetSLAM can be found in [5,6,27].

In the following, the trajectory of a pedestrian is called a data set. Different data sets, i.e., several trajectories, are applied to FeetSLAM in a first iteration without any prior knowledge for estimating a first set of environmental maps (for each data set we obtain an individual map). A general assumption in FeetSLAM is that the maps are rotational invariant. “Rotational invariant” means that depending on the sigma used for the starting heading—a higher sigma, i.e., diversity, helps reach convergence—the final SLAM might converge to a map that is rotated to the assumed SCs, but better in performance. The estimated maps can then be combined so that they fit best either with a geometric transformation [5,6] or a Hugh transform [27] reducing the complexity from quadratic to linear. The same transformation that is used for merging the maps is applied to the SCs. In [27], it has been shown that the map combination is successful with individual converged maps of a whole area or reasonable parts of that area.

In the next iteration of FeetSLAM, the merged posterior maps are used as prior maps. It is worth noting that the prior map for a specific data set in the next iteration is the combination of the resulting posterior maps of every data set except the specific data set over which FeetSLAM is actually performed. The map of the specific data set is excluded intentionally when calculating the prior map to prevent favoring the specific data set. FeetSLAM iterates in this way several times until a good estimation is found. When taking into account the crossed wall ratio, it has been shown that this ratio does not change much after 10 iterations [6].

The transformation part of FeetSLAM is very computationally complex. Therefore, it is omitted in this paper which facilitates real-time operation. All other estimated maps, except the specific data set, are merged and directly applied as prior map without transformation. Contrary to the assumption that the maps are rotational invariant, we assume that the initial direction and position are known from the SCs and that FeetSLAM estimates the direction of the maps precisely so that the transformation of the map is unnecessary.

## 4. Partly Successive Collaborative FeetSLAM

### 4.1. Partly Successive Collaborative FeetSLAM Algorithm

When starting to explore the unknown building, it is very unreliable that emergency forces revisit areas inside the building, which is usually necessary for correcting the map provided by SLAM. Therefore, the estimated map is uncertain and the estimated position of the best particle is not very accurate due to the fact that the SLAM algorithm was not yet able to efficiently learn the environment for restricting the drift (see also [7]). Therefore, in this paper, the collaborative SLAM approach is applied partly successively with a sliding window technique. The idea of successive FeetSLAM has been already presented in [7]. Based on this idea, a new approach is developed in this paper which is called partly successive FeetSLAM. The advantage of this new approach is that it is less computationally complex and, therefore, more applicable for real-time operations.

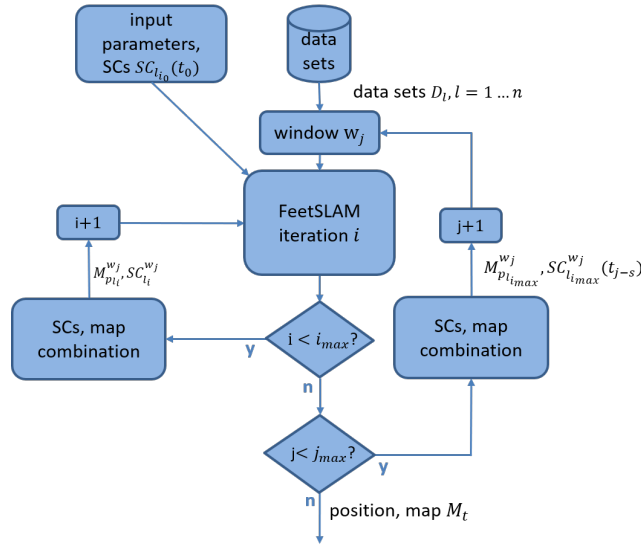
In the original FeetSLAM algorithm, the converged maps of whole data sets are combined. In contrast to this and also to successive FeetSLAM, we apply FeetSLAM only partly successively without having to run the algorithm over the whole trajectory again. For this, we divide the walks into overlapping windows and apply FeetSLAM only on small portions of the walks. By applying FeetSLAM partly successively, the drift can be reduced because the information of several sensors is used together. The drift is small after a short distance of walking, and correcting the drift directly after short distances leads to a better overall performance. Since we use delta step and heading measurements as inputs to FeetSLAM and no absolute positions, we can directly apply them to partly successive FeetSLAM.

A schematic illustration of partly successive collaborative FeetSLAM is given in Figure 1. Before starting FeetSLAM, the data sets are divided into overlapping windows. In our implementation,

we divided the odometry data depending on the distance traveled: After a certain distance (5 m in our case), the time of reaching that distance is stored in a  $m$ -dimensional time vector  $t_d = \{t_0, \dots, t_j, \dots, t_m\}$ , where  $t_j$  is the time of reaching  $j \cdot d$  [m]. The windows over the  $n$  data sets  $D_l, l = 1 \dots n$  are generated in the following way:

$$w_{(j-s:j)} = w_j = \begin{cases} D_l(t_0:t_j) & \forall j < s \\ D_l(t_{j-w}:t_j) & \forall j \geq s \end{cases} \quad (1)$$

where  $s$  is the window size and  $s \geq 2$ . The last window ends with the maximum duration  $t_{max}$  of the data set. For each window  $w_j$  of data sets  $D_l, l = 1 \dots n$  partly successive FeetSLAM iterates  $i_{max}$  times.



**Figure 1.** Schematic illustration of partly successive FeetSLAM.

The SCs  $SC_{l_{i_0}}$  for data set  $l$  at iteration  $i_0$  that are used as input parameters to the original FeetSLAM algorithm are the following:

$$SC_{l_{i_0}}(t_0) = (p_{l_x}(t_0), p_{l_y}(t_0), p_{l_z}(t_0), \alpha_l(t_0), f_{s_l}(t_0), \sigma_{p_{l_x}}(t_0), \sigma_{p_{l_y}}(t_0), \sigma_{p_{l_z}}(t_0), \sigma_{\alpha_l}(t_0), \sigma_{f_{s_l}}(t_0)), \quad (2)$$

where  $p_{l_x}(t_0), p_{l_y}(t_0), p_{l_z}(t_0)$  are the initial  $x, y, z$ -positions,  $\alpha_l(t_0)$  is the heading, and  $f_{s_l}(t_0)$  is the scale factor for data set  $l$  at starting time  $t_0$ , respectively. The respective uncertainties of the SC-values are described via the associated sigma values  $\sigma_{p_{l_x}}(t_0), \sigma_{p_{l_y}}(t_0), \sigma_{p_{l_z}}(t_0), \sigma_{\alpha_l}(t_0)$ , and  $\sigma_{f_{s_l}}(t_0)$ . The distributions for the starting values are assumed to be Gaussian distributions with respective sigma values.

In this paper, we assume to know the SCs at  $t_0$  for all data sets. Therefore, for the first window(s)  $w_j, j < s$  the SCs are known, because they start at  $t_0$ . Since with  $j \geq s$  the SCs are not known, the SCs at different starting times of the windows are estimated and stored during the run over the previous window. More specifically, the SCs  $SC_{l_{i_0}}^{w_j}$  for window  $w_j, j \geq s$  of data set  $l$  at iteration  $i_0$  are the  $x, y, z$ -positions, heading, and scale factor from the results of FeetSLAM over the previous window  $w_{j-1}$  at times  $t_{j-s}$ :

$$SC_{l_{i_0}}^{w_j}(t_{j-s}) = SC_{l_{i_{max}}}^{w_{j-1}}(t_{j-s}) \quad (3)$$

In the following, for simplicity, we omit the index  $l$ . With this, the SCs for a new window  $w_j$  at iteration  $i_0$  that are used in partly successive FeetSLAM can be written as:

$$\begin{aligned} SC_{i_0}^{w_j}(t_{j-s}) = & (p_{x,i_{max}}^{w_{j-1}}(t_{j-s}), p_{y,i_{max}}^{w_{j-1}}(t_{j-s}), p_{z,i_{max}}^{w_{j-1}}(t_{j-s}), \alpha_{i_{max}}^{w_{j-1}}(t_{j-s}), f_{s_{i_{max}}}^{w_{j-1}}(t_{j-s}), \\ & \sigma_{p_{x,i_{max}}}^{w_{j-1}}(t_{j-s}), \sigma_{p_{y,i_{max}}}^{w_{j-1}}(t_{j-s}), \sigma_{p_{z,i_{max}}}^{w_{j-1}}(t_{j-s}), \sigma_{\alpha_{i_{max}}}^{w_{j-1}}(t_{j-s}), \sigma_{f_{s_{i_{max}}}}^{w_{j-1}}(t_{j-s})) \quad . \end{aligned} \quad (4)$$

In our implementation, we store the position and heading of the best particle at times  $t_{j-s}$  during the last iteration  $i_{max}$  over window  $w_{j-1}$  for using them as SCs for the next window  $w_j$  at iteration  $i_0$ . The SCs for the subsequent iterations are determined in the same way as in the original FeetSLAM algorithm described in [6].

FeetSLAM itself iterates  $i_{max}$  times over the respective window of all data sets. In addition to the SCs and the windowed data sets, a prior map is used as input to FeetSLAM if available. In the first iteration, no prior map is available. In partly successive FeetSLAM, similarly no prior map is available for the first window  $w_1$ . For all successive iterations we are able to use a combination of the estimated posterior maps from the previous iteration as in the original FeetSLAM algorithm. More specifically, at each iteration  $i$  of FeetSLAM applied to the windows  $w_j$  of all  $n$  data sets, the resulting posterior maps  $(M_{1_i}^{w_j}, \dots, M_{n_i}^{w_j})$  are combined in the same way as in the original FeetSLAM algorithm to a prior map  $M_{p_{l_i}}^{w_j}$ ,  $l = 1 \dots n$ , where the map of the respective specific data set is intentionally excluded in the combination process. The prior map  $M_{p_{l_i}}^{w_j}$  is then applied to data set  $l$  in the next iteration.

When calculating partly successive FeetSLAM over a new window  $w_j$ , the calculated prior map  $M_{p_{l_{i_{max}}}}^{w_{j-1}}$  of the last iteration of the previous window is used for the first iteration  $i_0$  of window  $w_j$ . In addition to that, after the first iteration  $i_0$  the total combined posterior map at iteration  $i_{max}$  of the previous window  $w_{j-1}$  is additionally combined with the respective estimated posterior maps of the current window  $w_j$  for generating the prior maps for the next iteration. This assures that the already walked area from the previous window is included in the following prior maps. Since this is done for every window, the resulting prior map contains walked areas from all previous windows. Without adding the map of the previous window, the resulting prior/posterior maps represent only walked area of the current window.

#### 4.2. Adjustment of the First Heading Angle

It has been observed that the estimated trajectories based on the measurements of the six sensors slightly differ in the first heading angle. This is because the sensors provide only a relative position and the sensors may be mounted with a small heading error—in our experiments the sensors might be mounted with different small errors with regard to zero heading—so that the starting heading differs. Whereas the z-direction can be estimated via the gravitational force, there is no possibility to adjust the heading of the sensor in terms of the first heading. For suppressing this effect, the first angle is adjusted to the known starting direction. This is done by the following algorithm: The first 2–3 steps are assumed to be on a straight line in the direction given by the SCs. If the estimated heading differs from that straight line, the angle offset is calculated and subtracted from the estimated angle in the following. With this, the starting headings of all trajectories of the different sensors are corrected. For correcting the starting heading, we used the following formula:

$$\Delta\alpha_l(t_0) = \arctan(\delta_{y_l}/\delta_{x_l}) \quad , \quad (5)$$

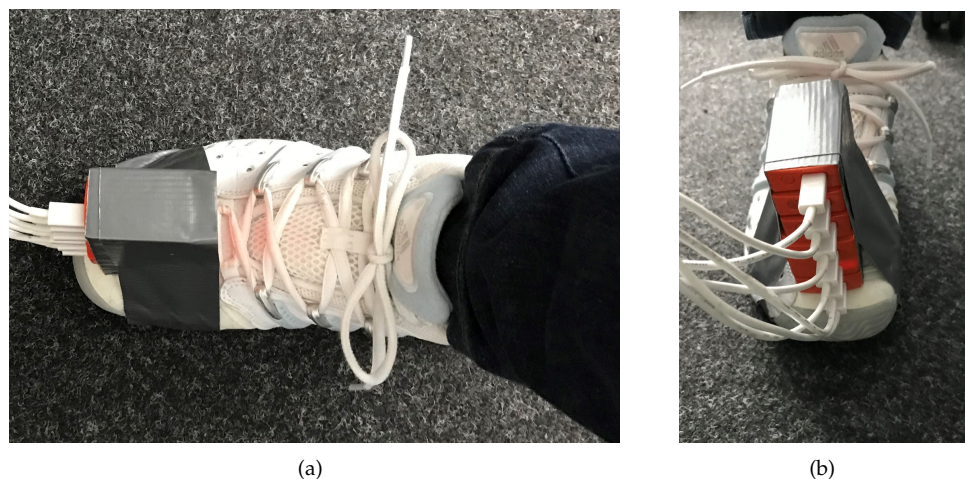
where  $\Delta\alpha_l(t_0)$  is the heading angle correction for data set  $l$ , assuming a starting heading of  $\alpha_l(t_0) = 0^\circ$ , and  $\delta_{x_l}, \delta_{y_l}$  are the differences between the  $x_{l_0}, y_{l_0}$ -values at the starting point and  $x_{l_d}, y_{l_d}$ -values at a certain distance  $d$ , respectively. In our simulations, we used a distance of 1.5 m, because, later in the real experiments, we crossed at the beginning two ground truth points (GTPs) that were in the starting direction with a distance of 1.6 m.

## 5. Experimental Results

### 5.1. General Settings

In our experiments, we collected data of different walks with a total of 6 sensors mounted on the foot of one pedestrian as depicted in Figure 2. It has been shown that the ZUPT-phase lasts the longest above the toes of the foot; therefore we used this placement. Our vision is to integrate the sensor in the sole of the shoe, but other placements like the ankle of the foot are also possible. The placement of the sensors should be carefully chosen due to the fact that sensor misplacement influences the ground reaction force estimation as shown in [28] and with it the results of the PDR. Due to the fact that the sensor measurement errors vary and the ZUPT detection varies with the measurement errors, too, we observed varying drifts similarly to drifts of trajectories from different pedestrians like in [29]. Therefore, we are convinced that the results are similar when using trajectories of different pedestrians. It should be noted that we use general parameters for a sensor type inside the UKF and do not need to specifically calibrate.

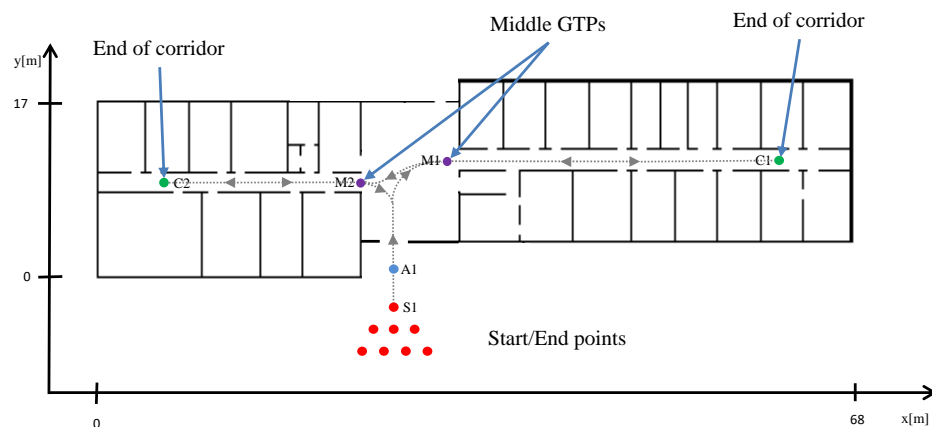
During the walk, we crossed several GTPs for providing a reference. For the indoor GTPs, the positions of the GTPs were measured with a laser distance measurer (LDM, millimeter accuracy) and a tape measurer (sub centimeter range accuracy), whereas outdoors we used a Leica tachymeter as described in [30].



**Figure 2.** Six Xsens sensors mounted on the foot of the pedestrian; (a) top view and (b) front view of the 6 sensors.

The GTPs used in the ground floor are depicted in Figure 3. The pedestrian started at different starting points in front of the building to emulate a group of emergency forces entering a building in triangle formation with a distance of 0.8 m between the forces. This is assumed as a sample formation. Depending on the situation, special forces are grouped in different formations, e.g., triangle or line formations. We chose the triangle formation because it is more challenging due to the different starting headings. When entering the building, the forces are assumed to be behind each other because the door and corridors leave no space for entering together at the same time in that formation. When not otherwise stated, the GTPs depicted in Figure 3 are used in the following order: S1, A1, M1, C1, M1, M2, C2, M2, A1, S1. When we started from another starting point than S1, we crossed additionally the respective starting point at the beginning and end of the walk. The points S1 and A1 are used for the adjustment of the first heading angle. As mentioned before, the distance between S1 and A1 was 1.6 m. The path was intentionally chosen without entering rooms, because entering and exiting rooms result in loop closures that help the SLAM to perform better. We wanted to test the longest way inside the building without loop closures. The pedestrian stopped at the GTPs for 2–3 s and the time when passing the GTP is logged.

The total number of walks was 10 which resulted in a total of 60 trajectories. The pedestrians walked 3.8–7.4 min with estimated distances of 137–226 m. In the first two walks, the trajectories inside the building are repeated. Additionally, to test the algorithm for the motion mode “running”, we performed 3 running walks with 6 sensors mounted on the foot following the same trajectory resulting in 18 different estimated trajectories. The duration of the walks was 1.9 min and the estimated distances 136–139 m. The six sensors used are one Xsens MTi-G sensor, three new Xsens MTw sensors, and two elder Xsens MTw sensors. The raw IMU data are processed by a lower UKF [24] and the following parameters are applied to the upper FeetSLAM-algorithm: 10.000 particles and 0.5 m for the hexagon radius.



**Figure 3.** Building layout of the ground floor of our office building. The ground truth points (GTP) used are marked in different colors. Several starting points—marked in red—are used to emulate a group of pedestrians or emergency forces entering the building. The dotted arrows/lines represent the starting/walking direction.

For the SCs, we set the following parameter values: Depending on the window, the mean  $x, y$ -positions and heading are either the known starting position or the respective positions of the best particle. The only parameter that we manually set was the mean scale: We set it to 1.0 for the sensors Xsens MTi-G and MTw, old generation, because they are properly calibrated by the factory and already provide a good estimation in  $x, y$ -distances. The new generation Xsens MTw-sensors, which are cheaper, are not factory calibrated as of our knowledge. We observed that we had to adjust the mean scale to 1.02 for all of them in order to get comparable results.

The sigma values of the SCs used in the experiments are given in Table 1. For windows starting at  $t_0$ , we set the sigma values to be zero to force the particles at the beginning to that position, direction, and scale. This does not mean that the particles have same position, heading, and scale at the beginning, because after initialization, we sample from the FootSLAM error model and add the drawn error vector to the initial position, heading, and scale. We refer the reader to [27], page 37, where a comprehensive description of the error model is provided. For the windows starting after  $t_0$ , we set the sigma values to fixed values as given in Table 1. Due to the fact that the sigma values of the best particles are sometimes very high and lead to disturbed results, we set most of the sigma values to fixed values in order to provide a reasonable position and heading of the particles. The sigma values for the  $x, y$ -positions are set with a small sigma because the position of the best particle might be inaccurate. The same holds for the heading-angle. We set the sigma-scale value to zero because of the good estimations of the UKF regarding the delta step measurements. Please note that we used these sigma values only for the first iteration of window  $w_j$ . After that, we set all sigma values to zero.

**Table 1.** Sigma values of the starting conditions (SC) for the first iteration of window  $w_j$ .

	$\sigma_{p_x}$	$\sigma_{p_y}$	$\sigma_{\alpha}$	$\sigma_{f_s}$
$SC_{i_0}^{w_j}(t_0) \forall j < s$	0.0	0.0	0.0	0.0
$SC_{i_0}^{w_j}(t_{j-s}) \forall j \geq s$	0.1	0.1	2.0	0.0

In the following, we explain the different error metrics that are used in the experimental results sections. The 2D-error  $e_l$  of data set  $l$  at the  $h$ th GTP  $g_h$ ,  $h = 1, \dots, n_g$ , where  $n_g$  is the number of GTPs, is calculated as the Euclidean distance to the known GTP:

$$e_l(g_h) = \sqrt{(p_{x_{g_h}} - \hat{p}_{x_{g_h}})^2 + (p_{y_{g_h}} - \hat{p}_{y_{g_h}})^2} \quad , \quad (6)$$

where  $p_{x/y_{g_h}}$  are the known and  $\hat{p}_{x/y_{g_h}}$  are the estimated  $x, y$ -positions at GTP  $g_h$ , respectively.

The mean error value  $\bar{e}$  is the mean over all  $n$  mean error results  $\bar{e}_l = \sum_{h=1}^{n_g} e_l(g_h)$  of the  $n$  data sets:

$$\bar{e} = \sum_{l=1}^n \bar{e}_l / n \quad . \quad (7)$$

The maximum error  $e_{\max}$  is the maximum over all maximum errors of the  $n$  error curves of the different data set results:

$$e_{\max} = \max_l(\max(e_l)) \quad . \quad (8)$$

It reflects the maximum error that may occur in one of the results for one data set.

It should be noted that we used in this paper 2D-FeetSLAM. Whereas it has been shown that the system also runs for 3D-environments, we concentrated in this paper on the 2D-case to show that it works. We are convinced that the algorithm performs similar in 3D-environments. In future work, we will extend it to 3D environments.

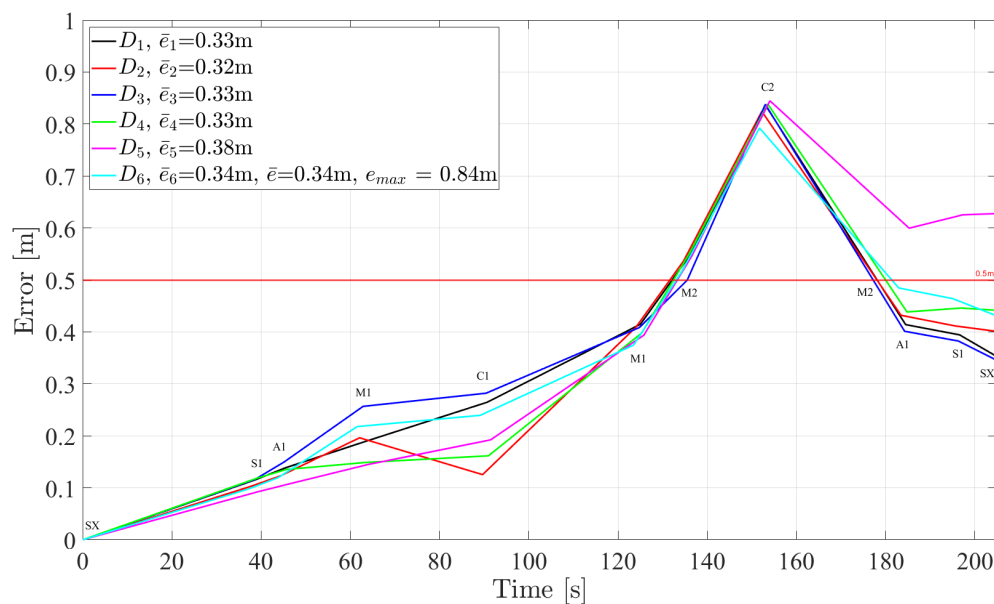
## 5.2. Experimental Results for Multiple Sensors

Figures 4 and 5 show the error performance over time and resulting tracks for partly successive FeetSLAM, respectively, for one of the 10 outdoor to indoor walks. The times for windowing were taken when passing 5 m of traveled distance measured with the first sensor. One can see that the error remains stable between 0 m and 1 m, and the mean error is below 0.5 m. The highest values are at the end of corridor C2. The heading drift causes sometimes higher errors at the end of the corridors. This is especially the case when the varying drifts of the six sensors do not compensate each other. Note that the errors at the end of the walk tend to be smaller due to the fact that the area is walked again and we return to the starting point at the end of the walk. This is because the accuracy of the estimated path is better at the beginning of the walk and FeetSLAM converges to it.

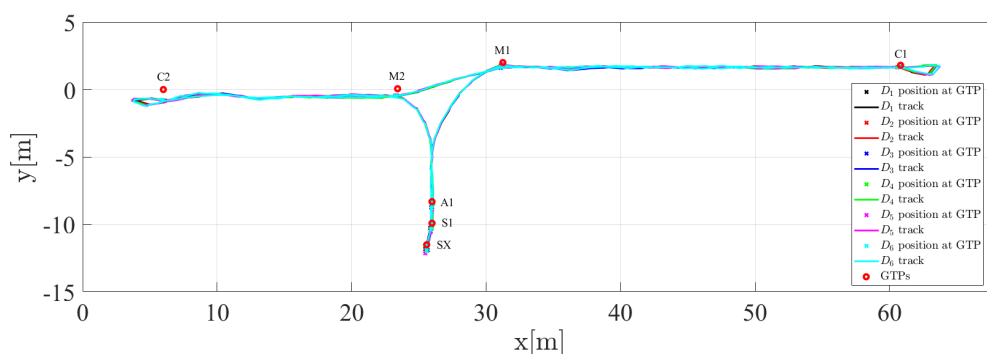
The mean position error and the maximum error values of all 10 multi-sensor walks are given in Table 2 for different window sizes. The results show that it is better to choose a larger window lengths  $s > 2$ . The window length  $s = 2$  seems sometimes to be too small in order to correct the drift. This is especially the case for walk 7 where we realized a high maximum error that might cause problems afterwards—for instance when using the estimated map as a prior map. The SCs of the best particles are sometimes not reliable especially regarding the heading angle. With a higher window length, the particles are forced to be on the previous paths for a longer time and, therefore, it is more probable that the path is followed. This holds especially for the difference between  $s = 2$  and  $s = 3$ . Compared to that, the difference between window lengths  $s = 3$  and  $s = 4$  is only small. In the following, we have chosen a window length of  $s = 3$ , which also means lower computational complexity than a window length of  $s = 4$ .

Table 2 shows the results for FeetSLAM computed over the whole walk and successive FeetSLAM without windowing. One can observe that FeetSLAM provides mean values that are always below

0.5 m and the maximum error is only once slightly over 1 m. The advantage of original FeetSLAM is that for long walks with loop closures, the estimated path is calculated considering the whole path history. Therefore, better convergence can be found in the case of loop closures. These are not real-time results: We obtain the results after performing FeetSLAM over the whole walk(s). FeetSLAM in the original form cannot be applied in real-time, whereas FootSLAM and partly successive FeetSLAM can be applied in real-time. If we apply FootSLAM in real-time, we obtain much worse results especially when exploring a new area (see also [31], Figure 6.10, results for using no prior information). With partly successive FeetSLAM, we obtained on average slightly worse results than FeetSLAM, but we obtain them from the beginning on. One main goal of this paper was to provide accurate real-time results in indoor environments.



**Figure 4.** Error performance of partly successive FeetSLAM for one walk with 6 sensors mounted on the foot. The window size was 3 and the number of iteration was 3. The results are always below one meter and the average result over all six data sets was 0.34 m.



**Figure 5.** Resulting tracks after performing partly successive FeetSLAM for one walk with 6 sensors mounted on the foot. The window size was 3 and the number of iteration was 3.

The results for successive FeetSLAM are worse than for partly successive FeetSLAM ( $s = 3$ ). This is especially the case when the trajectories suffer from similar drifts that can mainly be seen in higher errors  $e_{\max}$ . Then the results are forced to the average drift. Revisiting areas, that helps in the case of collaborative FeetSLAM does not help in this case because we are already forced to that drift (the drifted map is already learned). This effect is strongly reduced with partly successive FeetSLAM

because we do consider only portions of the walk that will be corrected directly and we do not rely on the whole paths. This is the main advantage of the windowing in partly successive FeetSLAM.

**Table 2.** Mean position error  $\bar{e}$  and maximum error  $e_{\max}$  of 10 different walks with 6 sensors mounted on the foot for partly successive FeetSLAM for different windows lengths, successive FeetSLAM, and pure FeetSLAM computed over the whole walk. The number of iterations was 3. The maximum error is depicted in brackets.

Walk Number	Window Length			Successive FeetSLAM	
	$s = 2$	$s = 3$	$s = 4$	FeetSLAM	FeetSLAM
1	0.40 (0.72)	0.43 (1.10)	0.40 (0.76)	0.40 (0.75)	0.40 (0.74)
2	0.37 (0.77)	0.51 (0.83)	0.48 (0.82)	0.65 (2.01)	0.41 (0.77)
3	0.48 (1.38)	0.37 (1.46)	0.23 (0.64)	0.41 (1.67)	0.31 (0.57)
4	0.30 (0.59)	0.22 (0.64)	0.32 (0.65)	0.31 (0.63)	0.34 (0.67)
5	0.45 (1.69)	0.21 (0.48)	0.21 (0.50)	0.35 (1.54)	0.44 (0.80)
6	0.46 (1.50)	0.45 (1.04)	0.43 (1.84)	0.68 (1.23)	0.28 (1.05)
7	0.71 (2.32)	0.26 (0.67)	0.27 (0.74)	0.66 (2.32)	0.25 (0.68)
8	0.63 (1.81)	0.57 (1.30)	0.58 (1.27)	0.49 (1.66)	0.40 (0.92)
9	0.25 (1.13)	0.34 (0.84)	0.33 (0.84)	0.31 (1.05)	0.18 (0.44)
10	0.35 (1.05)	0.26 (0.56)	0.25 (0.55)	0.41 (0.77)	0.37 (0.78)

Table 3 shows the mean and maximum errors for the 10 walks with 6 sensors mounted on the foot for partly successive FeetSLAM as a function of different iterations. The values for  $i = 3, s = 3$  are already given in Table 2 and can be compared to it. From this table, one can see that zero iterations do not help to provide good results because we do not use any prior information at all and the windowing does not allow FeetSLAM to correct the drift, which would be the case when using single-user FeetSLAM over the whole walk. Therefore, iterations are as expected necessary for drift correction in partly successive FeetSLAM. The results are best for  $i = 3$ ; therefore, we choose this value for further results. For more iterations, especially for drifted trajectories of same directions, the drift enlarges because FeetSLAM learns more and more the drifted trajectory. This strongly depends on the drift of the walks used. If there are more walks with a drift in one direction, this effect is higher.

**Table 3.** Mean position error  $\bar{e}$  and maximum error  $e_{\max}$  (in brackets) of 10 different walks with 6 sensors mounted on the foot for partly successive FeetSLAM with different number of iterations  $i$ . The window size  $s$  was 3.

Walk Number	Iteration Numbers			
	$i = 9$	$i = 6$	$i = 1$	$i = 0$
1	0.51 (1.89)	0.48 (1.58)	0.45 (1.41)	1.20 (20.59)
2	0.47 (0.83)	0.47 (0.83)	0.51 (0.83)	1.53 (8.08)
3	0.33 (1.18)	0.38 (1.51)	0.54 (1.95)	0.50 (1.80)
4	0.29 (0.63)	0.29 (0.63)	0.37 (0.62)	0.52 (2.63)
5	0.21 (0.48)	0.21 (0.48)	0.22 (0.55)	0.33 (0.71)
6	0.54 (0.99)	0.54 (0.96)	0.46 (0.95)	1.88 (7.46)
7	0.42 (1.07)	0.42 (1.06)	0.57 (1.70)	1.17 (5.55)
8	0.57 (1.32)	0.57 (1.32)	0.60 (1.38)	1.09 (10.08)
9	0.34 (0.84)	0.34 (0.84)	0.34 (0.84)	1.31 (15.66)
10	0.26 (0.56)	0.26 (0.56)	0.25 (0.60)	0.89 (5.03)

### 5.3. Experimental Results for Multiple Users

For emulating the multi-user case, we used a trajectory combination from 8 of the 10 multi-sensor walks that were starting at different starting points (see also Figure 3). Since the first 2 walks are repeated walks, we used the remaining 8 walks following similar paths. From each of the 8 walks, we used the trajectory of a different sensor and we alternated the sensors used so that we obtained 6 data set combinations comprising walks starting at 8 different points. It should be noted that the

times for the windows differ now and that the windows do not end at same positions due to the varying trajectories. The results for the 6 walk combinations are given in Table 4. One can see that partly successive FeetSLAM results in mean error values below 0.5 m for 5 of the 6 walk combinations. For walk combination 2, we obtained a higher drift which is a result of combining walks with similar drift. This is a problem that naturally cannot be avoided: When using walks that suffer from similar drifts, the result of partly successive FeetSLAM will also be drifted.

Table 4 shows the results for a smaller number of walks in a combination. It would have been expected that the results are getting worse with less number of walks. As it can be realized from the table, this is not always the case. The reason for it is that the results strongly depend on (i) the drifts of the walks chosen and (ii) the number of walks ending at a different starting position. With the multi-user walks, it has been observed that the error is larger at the end of the walks because they end at different positions. This is also one reason for getting worse when using the trajectories of more users. For walk combination 2 (8 and 4 pedestrians), a bad walk combination with higher non-compensating drifts is the reason for getting worse results. The results show that partly successive FeetSLAM is also successful in the multi-user case except for a bad walk combination.

**Table 4.** Mean position error  $\bar{e}$  and maximum error  $e_{\max}$  (in brackets) of 6 walk combinations of 8/6/4 pedestrians starting at different starting positions in front of the building for partly successive FeetSLAM. The number of iterations was 3 and the window size was 3.

Walk Combination	Walk Numbers		
	$n = 8$	$n = 6$	$n = 4$
1	0.36 (0.82)	0.51 (0.91)	0.37 (0.84)
2	0.72 (2.13)	0.51 (1.02)	0.73 (2.45)
3	0.38 (0.95)	0.38 (0.95)	0.40 (0.85)
4	0.36 (0.94)	0.36 (0.94)	0.37 (0.79)
5	0.48 (0.97)	0.36 (0.81)	0.38 (0.84)
6	0.47 (0.95)	0.48 (0.97)	0.35 (0.8)

#### 5.4. Experimental Results for Motion Mode “Running”

To test the motion mode running, we performed partly successive FeetSLAM on the 3 running walks. Since we observed slightly more drift due to some missing ZUPTs, we varied the  $\sigma_\alpha$ -values applied for windows  $w_j, j \geq s$ . The results are shown in Table 5. We obtained similar results for running as for walking. In walk 2, we observed that we had 4 trajectory results with drifts in the same direction, which resulted in a drift as well (same problem as mentioned before). Therefore, the results are worse than the others. For walk 3, we observed that the results for one sensor was worse than the others when using  $\sigma_\alpha(t_{j-s}) = 2.0 \forall w_j, j \geq s$ . With higher diversity for the heading we could solve this problem (see results for  $\sigma_\alpha(t_{j-s}) = 3.0 \forall w_j, j \geq s$ ), which slightly worsened the results for walk 2 by increasing the wrong average drift direction of the different sensor results.

For motion mode running, we had to adapt the ZUPT-threshold in the lower PDR. In addition, we had to enlarge  $f_s$  slightly, because the distance estimated was slightly lower than the real distance. Running is for instance intensively investigated in [32] for a motion capture suit. In addition to the fact that reasonable results are obtained with the motion mode running, a more comprehensive study of the lower PDR including ZUPT detection in the case of different motion modes like crawling, crouching, or foot drift is needed in order to adaptively adjust the parameters to different motion modes.

**Table 5.** Mean position error  $\bar{e}$  and maximum error  $e_{\max}$  (in brackets) of 3 multi-sensor running walks with 6 sensors mounted on the foot starting in front of the building for partly successive FeetSLAM. The number of iterations was 3 and the window size was 3.

Walk Number	$\sigma_{\alpha}(t_{j-s}) = 2.0$ $\forall w_j, j \geq s$	$\sigma_{\alpha}(t_{j-s}) = 3.0$ $\forall w_j, j \geq s$
1	0.42 (1.40)	0.36 (1.10)
2	0.76 (2.37)	0.84 (3.17)
3	0.48 (4.10)	0.37 (1.31)

### 5.5. Experimental Results for Disturbed Starting Conditions

Whereas disturbances of the SCs will lead to a shifted and rotated map in the case of wearing multiple sensors located on the foot—the map will not be correctly pinned to the global navigation frame but correctly estimated—they will influence the map estimation in the multi-user case. To evaluate the impact of disturbances of the SCs in the multi-user case, we tested one of the data sets (multi-user walk combination 1, 8 walks) with (i) higher sigma values for the SCs and (ii) disturbed SCs. In the first case, we applied two combinations for the sigma values as given in Table 6, lines 1 and 2, as the first SC  $SC_{i_0}^{w_j}(t_0) \forall j < s$ . In the second case, we added randomly  $\pm 0.2$  to the mean values of the known  $x, y$ -positions and 2.0 deg to the mean angle, and applied the sigma values of Table 6, line 3.

**Table 6.** Varied sigma values for the SCs.

Sigma Combination	$\sigma_{p_x}(t_0)$	$\sigma_{p_y}(t_0)$	$\sigma_{\alpha}(t_0)$	$\sigma_{f_s}(t_0)$
1	0.5	0.5	1.0	0.0
2	2.0	2.0	4.0	0.0
3	0.5	0.5	2.0	0.0

The results are given in Table 7. One can see that with moderate uncertainties on the SCs, the results are comparable to those of applying no uncertainty. In addition, if the SCs are disturbed, we achieved also comparable results. As a result that we can add uncertainty to the SCs, partly successive FeetSLAM is able to handle disturbances of the SCs. If we apply high uncertainty to the SCs, the values are much worse. We observed that the resulting map seemed to be correctly estimated but it was shifted in  $x$ - and  $y$ -directions and not rotated. That leads us to the conclusion that the algorithm is not very sensitive to uncertainties in the heading angle and that it is nevertheless possible to obtain a reasonable map that might only be shifted. Due to the fact that we used similar trajectories, the curves in the trajectories lead to convergence. In addition, because the algorithm can handle heading uncertainties we are convinced that the heading angle correction is most probably obsolete.

**Table 7.** Mean position error  $\bar{e}$  and maximum error  $e_{\max}$  (in brackets) of the first walk combinations of 8 pedestrians starting at different starting positions in front of the building for partly successive FeetSLAM. The number of iteration was 3 and the window size was 3. The results are given for varying sigma values for the SCs (line 1 and 2) and additionally disturbing the SCs (line 3). The respective sigma values are given in Table 6.

Sigma Combination	$\bar{e} (e_{\max})$
1	0.58 (0.94)
2	1.42 (1.79)
3	0.62 (0.97)

## 6. Conclusions

The main problem for inertial pedestrian navigation is the inherent drift of the IMU sensors. This drift can be reduced for instance with a SLAM technique when revisiting areas. However, in emergency applications, the assumption of revisiting areas in indoor scenarios does not always hold. Moreover, signals of opportunity might not be available to make use of them because they depend on the availability of infrastructure. Therefore, a new technique called partly successive collaborative FeetSLAM is proposed that is able to reduce the drift by combining either the trajectories of multiple sensors worn by a single pedestrian or combining trajectories from multiple users following similar paths. With the new partly successive FeetSLAM algorithm, the mean error in indoor areas can be reduced to  $\approx 0.5$  m. Partly successive FeetSLAM can be performed in real-time and is especially advantageous when exploring new areas. This will help positioning based on inertial measurements in large rooms like an atrium or a factory hall, where FootSLAM has no chance to converge when there is no loop closure and a quasi random walk is performed.

In future work, partly successive FeetSLAM is combined with GNSS or, e.g., UWB in order to obtain the starting conditions, which are assumed to be known in this paper. The robustness of the algorithm will be tested with different motion modes. In addition, we will continue investigating the successive collaborative mapping especially for localization in large open environments. Furthermore, the communication part for collaborative SLAM will be investigated and integrated, and the algorithm will be tested in real scenarios. The overall goal of the work was to provide seamless outdoor/indoor navigation for professional use cases that is very accurate and reliable over a long period of time.

**Funding:** This research received no external funding.

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

## References

1. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 1067–1080. [[CrossRef](#)]
2. Gu, Y.; Lo, A. A Survey of Indoor Positioning Systems for Wireless Personal Networks. *IEEE Commun. Surv. Tutor. II* **2009**, *17*, 1281–1293. [[CrossRef](#)]
3. Harle, R. A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1281–1293. [[CrossRef](#)]
4. Correa, A.; Barcelo, M.; Morell, A.; Vicario, J. A Review of Pedestrian Indoor Positioning Systems for Mass Market Applications. *Sensor* **2017**, *17*, 1–26.
5. Robertson, P.; Garcia Puyol, M.; Angermann, M. Collaborative Pedestrian Mapping of Buildings Using Inertial Sensors and FootSLAM. In Proceedings of the ION GNSS, Portland, OR, USA, 20–23 September 2011.
6. Garcia Puyol, M. Merging of Maps Obtained with Human Odometry Based on FootSLAM for Pedestrian Navigation. Master's Thesis, University of Malaga, Malaga, Spain, 2011.
7. Kaiser, S. Successive Collaborative SLAM for Professional Use Cases: Map Learning Aspects. In Proceedings of the IEEE Indoor Positioning and Indoor Navigation Conference (IPIN), Pisa, Italy, 30 September–3 October 2019.
8. Burgard, W.; Moors, M.; Fox, D.; Simmons, R.; Thrun, S. Collaborative Multi-Robot Exploration. In Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24–28 April 2000.
9. Durrant-Whyte, H.; Bailey, T. Simultaneous Localization and Mapping: Part I. *Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
10. Krinkin, K.; Filatov, A.; Filatov, A. Modern Multi-Agent SLAM Approaches Survey. In Proceedings of the 20th Conference of Fruct Association, Saint Petersburg, Russia, 3–7 April 2017.
11. Saeedi, S.; Trentini, M.; Seto, M.; Li, H. Multiple-Robot Simultaneous Localization and Mapping: A Review. *J. Field Robot.* **2016**, *33*, 3–46. [[CrossRef](#)]
12. Zou, D.; Tan, P. CoSLAM: Collaborative Visual SLAM in Dynamic Environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 354–366. [[CrossRef](#)]

13. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2015**, *43*, 55–81. [[CrossRef](#)]
14. Huai, J. Collaborative SLAM with Crowdsourced Data. Ph.D. Thesis, The Ohio State University, Columbus, OH, USA, 2017.
15. Nettleton, E.; Thrun, S.; Durrant-Whyte, H.; Sukkarieh, S. Decentralised SLAM with Low-Bandwidth Communication for Teams of Vehicles. In *Field and Service Robotics. Springer Tracts in Advanced Robotics*; Yuta, S., Asama, H., Prassler, E., Tsubouchi, T., Thrun, S., Eds.; Springer Nature Switzerland AG: Basel, Switzerland, 2006; Volume 24, pp. 179–188.
16. Kleiner, A.; Sun, D. Decentralized SLAM for Pedestrians without direct Communication. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007.
17. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, L.; Leonard, J. Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [[CrossRef](#)]
18. Nilsson, J.O.; Skog, I. Inertial Sensor Arrays—A Literature Review. In Proceedings of the European Navigation Conference (ENC), Helsinki, Finland, 30 May–2 June 2016.
19. Skog, I.; Nilsson, J.O.; Händel, P.; Nehorai, A. Inertial Sensor Arrays, Maximum Likelihood, and Cramér-Rao Bound. *IEEE Trans. Signal Process.* **2016**, *64*, 4218–4227. [[CrossRef](#)]
20. Angermann, M.; Robertson, P. Footslam: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors—Hitchhiking on Human Perception and Cognition. *Proc. IEEE* **2012**, *100*, 1840–1848. [[CrossRef](#)]
21. Garcia Puyol, M.; Robertson, P.; Angermann, M. Managing Large-Scale Mapping and Localization for Pedestrians Using Inertial Sensors. In Proceedings of the 10th IEEE International Workshop on Managing Ubiquitous Communications and Services, San Diego, CA, USA, 8–22 March 2013.
22. Van Opdenbosch, D.; Aykut, T.; Oelsch, M.; Alt, N.; Steinbach, E. Efficient Map Compression for Collaborative Visual SLAM. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018.
23. Wolf, D.; Sukhatme, G. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. In *Autonomous Robots*; Springer Science + Business Media, LLC, Part of Springer Nature Switzerland AG: Basel, Switzerland, 2005; Volume 19, pp. 53–65.
24. Zampella, F.; Khider, M.; Robertson, P.; Jimenez, A. Unscented Kalman Filter and Magnetic Angular Rate Update (MARU) for an Improved Pedestrian Dead-Reckoning. In Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS), Myrtle Beach, SC, USA, 23–26 April 2012.
25. Munoz Diaz, E.; Mendiguchia Gonzalez, A.; de Ponte Müller, F. Standalone Inertial Pocket Navigation System. In Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS), Monterey, CA, USA, 5–8 May 2014.
26. Kaiser, S.; Munoz Diaz, E. PocketSLAM based on the Principle of the FootSLAM Algorithm. In Proceedings of the International Conference on Localization and GNSS (ICL-GNSS), Gothenburg, Sweden, 22–24 June 2015.
27. Garcia Puyol, M. Crowdsourcing Motion Maps Based on FootSLAM for Reliable Indoor Pedestrian Navigation. Ph.D. Thesis, Technical University of Munich, Munich, Germany, 2011.
28. Tan, T.; Chiasson, D.; Hu, H.; Shull, P. Influence of IMU position and orientation placement errors on ground reaction force estimation. *J. Biomech.* **2019**, *97*. [[CrossRef](#)] [[PubMed](#)]
29. Munoz Diaz, E.; Kaiser, S.; Bousdar Ahmed, D. Height Error Correction for Shoe-Mounted Inertial Sensors Exploiting Foot Dynamics. *Sensors* **2018**, *18*, 888. [[CrossRef](#)] [[PubMed](#)]
30. Bousdar Ahmed, D.; Díez Blance, L.E.; Munoz Diaz, E. Performance comparison of wearable-based pedestrian navigation systems in large areas. In Proceedings of the IEEE Indoor Positioning and Indoor Navigation Conference (IPIN), Sapporo, Japan, 18–21 September 2017.

31. Kaiser, S.; Khider, M.; Garcia Puyol, M.; Bruno, L.; Robertson, P. Map-Aided Indoor Navigation. In *Indoor Wayfinding and Navigation*; Karimi, H., Ed.; CRC Press, Taylor & Francis: Boca Raton, FL, USA, 2015.
32. Wouda, F.; Giuberti, M.; Bellusci, G.; Maartens, E.; Reenalda, J.; van Beijnum, B.J.; Veltink, P. On the validity of different motion capture technologies for the analysis of running. In Proceedings of the 7th IEEE International Conference on Biomedical Robotics and Biomechatronics, Enschede, The Netherlands, 26–29 August 2018.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).