Coupled sonar inertial navigation system for pedestrian tracking

Hichem El Mokni*, Lars Broetje*, Felix Govaers*, Monika Wieneke*
* Sensor Data- and Information Fusion
Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE
Wachtberg, Germany.
{hichem.elmokni, lars.broetje, felix.govaers, monika.wieneke}@fkie.fraunhofer.de

Abstract – This paper presents a novel method for indoor pedestrian tracking using inertial sensing and sonar sensors. The zero velocity updating technique, which is used to enhance the performances of inertial sensing, cannot observe heading, resulting in a horizontal position drift. Sonar sensors are used as complementary technique to correct heading. The main idea is to extract a partial map of surrounding walls. Sonar returns are processed via the Hough transformation to extract wall segments (line features). The first detected segments will initiate a wall landmark. The pedestrian relative distance to next detected segments that are associated to a stored wall landmark, is considered to be his relative distance to that wall. An extended Kalman filter based solution to Bearing-Range simultaneous localization and mapping is used. The state vector is a concatenation of pedestrian position and positions of first points of each stored wall landmark, to which the virtual Bearing-range measurements are pointing. Both inertial and coupled sonar inertial tracks are visually compared to the true trajectory.

Keywords: inertial navigation, zero velocity updating, pedestrian tracking, sonar sensors, Hough transformation, Kalman filtering, simultaneous localization and mapping, SLAM.

1 Introduction

The production of pedestrian tracks (for fire-fighters for example) is a challenging task. The necessary "on the man" mounted sensor technology has to be lightweight and must not limit the freedom of movement. Beyond that, places of action are frequently within buildings or in urban area, so that GPS availability is usually reduced.

An inertial navigation system (INS) cannot be disturbed by external influences. An INS is a system, which combines acceleration sensors with orientation sensors, so that the position of a carrier can be extrapolated by a double integration and a known initial position. Thus it guarantees a constantly available, complete navigation solution; however this is consistent only for short time and suffers from the increasing navigational error with time, especially heading drift.

The stability of an inertial navigation can be extended, if the positioning procedure is externally enriched. That is the aim of this work, trying to enhance the INS navigation using some sonar sensors mounted on the tracked person. Sonar sensors belong to the class of range finders, widely used in robotic field to locate a robot in its environment and extract a map. Recently, most robot research focused on using laser scanners rather than sonar sensors, due to their higher accuracy and measurements density compared to the relatively wide beam of sonar. Nevertheless, the expansive price of laser scanners (more than 1000€ compared to roughly 50€ for a sonar sensor) and their non suitability to be mounted on persons (a weight of 1 kg and a volume of 15 cm cube approximately) make them inappropriate for our work. Some techniques will be used to overcome the inaccuracy of sonar returns. Section 2 starts with an overview of the tracking system. Section 3 presents the inertial sensing technique. Section 4 shows how are sonar data treated and coupled with inertial data. A short evaluation is presented in section 5.

2 Raw data extraction

For our work we used the MTi Xsens [9] sensor for inertial sensing, providing 3D orientation as well as 3D acceleration, 3D rate of turn and 3D earth-magnetic field data. The Devantech SRF08 [10] Range Finder was used as sonar sensor.

2.1 Hardware interconnection

Figure 1: sensors interconnection

Figure 1 shows how the sensors are mounted on the pedestrian. The INS sensor is mounted on the left foot.

Two sonar sensors are mounted near the shoulders, one oriented to the right and the other to the left of upper body direction (green arrow). The sonar sensors where simply mounted on an elastic band. All sensors are connected to a laptop via USB cables.

2.2 Data collector program

The program is based on a C++ program delivered with the MTi Xsens INS sensor which collects data delivered by accelerometers and gyro meters. We just injected some piece of code to collect in "parallel" sonar sensors returns. For each sonar sensor we created a separated thread (light process). In that way, all threads and main process will run in "parallel" and the operating system on the laptop will alternate in running each of them. To each sonar thread we pass the timer variable maintained by the main INS process. This timer is used to add a time stamp for every collected sonar return, which is necessary to synchronize INS and sonar data later.

3 Working on inertial navigation data

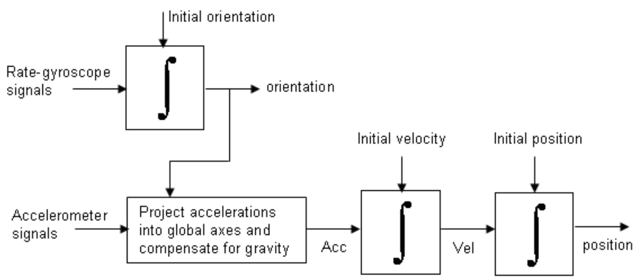


Figure 2: sensors interconnection

The error characteristic of inertial sensing makes it impossible to solely rely on this technique to develop a navigation solution. Even a small drift rate in gyros can lead to a position drift of more than 1 meter in only 10 seconds. As depicted in figure 2, rate-gyroscope signals are integrated to get the orientation. To track position, the three accelerometer signals are projected into global frame coordinates using the gyroscope orientation. After compensation of gravity acceleration, global acceleration is integrated to get velocity (using initial velocity), which in turn is integrated to get position (using initial position). This procedure is valid for strapdown inertial navigation system as the used MTi Xsens sensor, but stable platforms rely on the same principle [5]. Error in rate gyros is linearly related to acceleration error but the double integration results in a cubically growing position error. Zero velocity updating technique tries to overcome this drift problem by breaking open navigation into periods of about 0.5 seconds. A person walk is an alternation of two phases of approximately 0.5 seconds each of them: a stance phase where the foot stays motionless on the ground, and a moving phase. The idea is to apply zero velocity updates (ZUPTs) into the extended Kalman filter (EKF) navigation error corrector at stance phases (apply a virtual measurement of a zero velocity) [2]. A stance phase is detected when the sum of absolute accelerations is below a given threshold. Thus, when detecting a stance phase, the EKF is able to correct the velocity error, breaking the cubically growing error into a linear accumulation in the number of steps. The big advantage of introducing ZUPTs as measurements into the EKF instead of just resetting the velocity to zero is that it lets the EKF retroactively correct position, velocity, accelerometers biases, pitch, roll and the pitch and roll gyro biases. It sounds like a kind of magic, it only makes use of the strong correlation between acceleration, velocity, position, roll, and pitch errors. What we should retain is that introducing a correct velocity (ZUPT) or a correct position (GPS position if available) as a measurement into the extended Kalman filter navigation error corrector allows a retroactive correction of many navigation parameters. It is important to mention that zero velocity update is unable to correct the yaw (heading) and yaw bias. The coupled sonar technique of this work tries to solve this problem.

4 Coupling sonar data with INS data

Zero velocity updates are not able to observe yaw (heading) and yaw bias. So a navigation solution based on an inertial navigation system like the used MTi Xsens device will suffer from a drift in horizontal position. Thus, the tracked path of a person walking in a large closed loop path starts to curve and does not reach the starting position. To solve this problem we need a complementary technology providing position correction. In outdoor environments, GPS position fixes are commonly used for this purpose. In indoor environments, if a map is present, one can force walking through walls only at doors positions, which can be fed to the inertial system EKF as position fixes. For our work, however, we have the constraint of navigating in a mapless environment. The simultaneous localization and mapping [8] (SLAM) technique is used, with ultrasonic sensors as rangefinders. The main idea is to detect walls, and by preventing the track path crossing the walls, the heading is corrected. In fact we are detecting an eventual segment of the wall at each step. First we initiate the wall line using the first detected segments. Now each time we detect a new segment which does not fit into the wall line, the segment drift from the wall is used to correct the pedestrian position. The system works like depicted in figure 3.
First, during a step (from one stance phase to another), the INS EKF is updated each 10ms to compute inertial increments. In parallel, all sonar returns are accumulated. Once a stance phase is detected (step end), a zero velocity update is applied on the INS EKF. The last step position increments and accumulated sonar returns are now passed to the sonar subsystem to be treated. The detection of wall segments (next called features) is done by examining the accumulated sonar sensors returns and via the Hough transformation as explained in section 4.2. The evaluation process of the feature is presented in section 4.3. The previously identified walls will be next called landmarks, and so each detected feature will be either associated to a stored landmark, or will be the initiation of a new

landmark (new detected wall), or simply rejected; this process is explained in section 4.4. Finally, for the SLAM problem, an extended Kalman filter based solution is implemented, using range-bearing measurements. Section 4.5 explains the states of the tracked person and of the walls (the landmarks), and how are detected segments (features) transformed into range bearing measurements. The filtered SLAM EKF position produces the coupled Sonar INS track, which is drawn with the INS track on the indoor test map to be compared with the true trajectory. The conception of the sonar subsystem followed a paradigm for sensor data fusion presented in figure 4, inspired from reference [4].

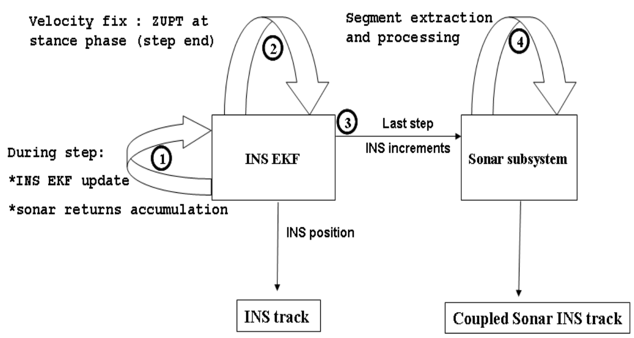


Figure 3: system architecture

4.1 Data alignment

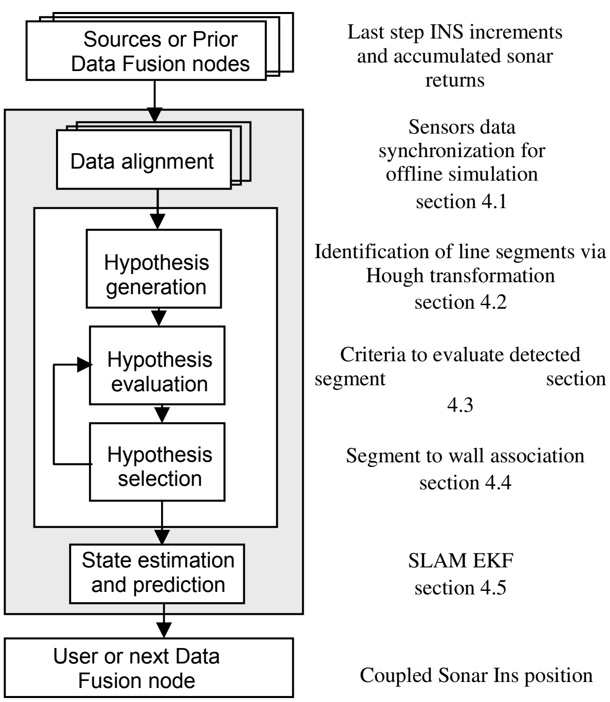


Figure 4: Sonar subsystem paradigm

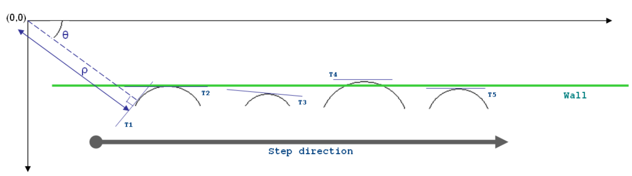
All our work is done offline, and so the first thing we need is to synchronize all gathered data from the MTi Xsens inertial sensor and both sonar sensors. The associated time stamp to each data sample (section 2.2) is used for that purpose. A first pass by the INS EKF on gathered inertial data file (containing accelerations and orientation at every 10ms) produces a trace file (INS track). The file is loaded into a linked chain where each node contains the time stamp, INS position and a Boolean indicating if the instance is a stance phase or not. Eventually, if at that time stamp sonar data were gathered (look in sonar output files), it will be concatenated to that node. A second pass is now performed by the sonar subsystem on the track linked chain, treating it node by node, to produce the coupled sonar INS track. The first and second passes are equivalent to the previously presented system architecture in figure 3.

4.2 Hypothesis generation

The hypothesis generation process extracts line features (wall segments) from raw sonar returns. The idea is that sonar returns are presented as arcs (see figure 7) that take into account sonar beam width uncertainty (only in 2D) [1]. The arcs originated from a wall are tangent to a common line segment representing the detected wall part position. We have used the OpenCV library to implement this process by graphically representing the sonar returns as arcs. For this task, we have made two assumptions. First we allowed displacements (steps) only in body upper part direction, so that the axis direction of both right and left sonar sensors is perpendicular to the displacement direction. This is necessary to know exactly the sonar sensors direction from one step to another, and so to draw the arcs corresponding to the returns in the correct direction. Using the stance phases which were used to apply zero velocity updating into INS EKF, we are able to detect steps. Thus, two stance positions separated by a moving phase determine the step direction, and consequently sonar sensors directions during this last step. The second assumption is related to the relative sonar sensors positions in the body frame of the MTi Xsens housing. We have used the INS position to be the position of the left sonar sensor, and we shifted with 40 cm to the right of step direction the position of the right sonar sensor (40 cm is the distance between the shoulders). Of course this second assumption is not quit rigorous, regarding that the INS mounted on the foot has an increasing velocity during the first half of the step and a decreasing velocity during the second half, whereas a sonar sensor mounted on the shoulder has a nearly constant velocity during the step. Nevertheless, this has no influence on the process. In fact, the sonar sensor positions are needed as origins for the arcs presenting the returns, and this assumption only translates those positions forwards (first half of the step) or backwards (second half of the step) in step direction, which will consequently translate the arcs also in step direction without affecting their tangibility to the wall segment if present. Contrariwise, INS positions, linearly related to sonar sensors positions in that way, are a little bit dispersed around the step direction and are never perfectly aligned, which is realistic in our application regarding that the shoulder relative position (sonar sensor position) to the foot (INS position) is not static during a person walk: shoulders drift to right and left during a walk. We believe that dealing with such uncertainty gives our application a more realistic aspect.

To illustrate, the feature extraction process starts as following:

- During a stance phase, we accumulate sonar returns (sensor position and range)
- Starting a moving phase (first non ZUPT moment), we save the last stance position, and we continue accumulating sonar returns.
- When first stance position reached (step end), we compute the step direction which corresponds to the angle of vector between actual position and saved last stance position.
- Now, we go back to all accumulated sonar returns, and we draw each of them as a circle arc using the previously stored sensor position. The arc is centred on an axis of orientation (step direction + 90°) if corresponding to the left sonar sensor, and of orientation (step direction - 90°) if corresponding to the right sonar sensor. The view angle of the arc is equal to 70°, which is greater than the 55° beam width of the used SRF08 sonar sensor. This compensates any drift in the orientation of the main axis of sonar sensors supposed to be perpendicular to step direction.



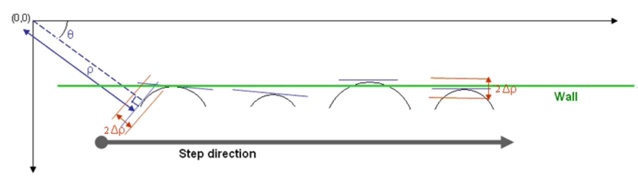
Partial Hough space view

ρ \ θ	.	.	.	.
.	0	0	2	0
.	0	1	0	2
.	0	0	1	1
.	0	1	0	0

Figure 5: missed wall detection

The feature detection process is based on Hough Transformation [7], which transforms a line into a point in the Hough space of coordinates ρ (distance between the line and the origin) and θ (angle of the vector from the origin to this closest point of the line), as explained for line T1 in figure 5. Hough transformation is useful when looking for a specific type of lines in a picture, which is exactly the case in our application: we have a number of arcs corresponding to sonar returns, and we are looking for a common tangent to the maximum number of arcs which corresponds most probably to a wall. The Hough space is a two dimensional array initialized to zeros, where the first axis corresponds to ρ and the second axis corresponds to θ. A line transformation corresponds to an increment by 1 of the Hough space array case of coordinates ρ and θ corresponding to that line. We take the set of points of all accumulated arcs corresponding to sonar returns of last step. For each point p, we are going to consider the line T passing through p and tangent to the corresponding arc. T is simply the line passing through p and perpendicular to the line formed by p and the centre of its corresponding arc. T is transformed into Hough space. After iterating through all points, the line corresponding to the maximum accumulation value in Hough space is exactly the line tangent to the maximum number of arcs (the number of arcs is equal to the accumulation value).

After implementing this feature detection process, we noticed that the detection rate was low. In fact, the accumulated errors of sonar sensors positions due to the assumptions that we have made can easily shift the drawn arcs, which causes the detection missing of a common tangent with a distinctive maximum accumulated value in Hough space. In figure 5 we are presenting an illustration of one step made by the pedestrian and the accumulated sonar returns of only the left sonar sensor. The real wall is represented in green. The wall was missed because it is tangent to only one arc (only tangent T2 fits into the wall), so the wall line has an accumulation value in Hough space equal to 1, whereas any common tangent between two arcs (there are several of such tangents) has an accumulation value equal to 2 in Hough space.



partial Hough space

ρ \ θ	.	.	.	.
.	0	2	3	1
.	2	3	0	2
.	2	0	4	1
.	1	2	1	1

Figure 6: surrounding parallel lines to tangents

The solution for this issue is to take under account the range uncertainty of the arcs, and that is to say the position drift of the tangents, and so for each arc tangent T, parallel lines to T and distant to T till 20 cm are considered and transformed into Hough space, as shown in figure 6 (the red lines). We found out that setting the maximum position drift to 20 cm is quiet reasonable. Let's consider a tangent T of coordinates ρ and θ. The surrounding parallel lines to T have coordinates ρ´ in the interval [ρ - Δρ, ρ + Δρ] and θ´ equal to θ, where Δρ is equal to 20 cm, and so the number of those lines depends on the resolution used for ρ (two lines for example if the resolution is equal to 10 cm).

Of course such an extension will increase many accumulation values in Hough space, precisely a line will be voted not only if it is tangent to an arc, but also if it is parallel and enough close (Δρ at most) to any arc tangent. Nevertheless, the line that will be most voted, and so will have the maximum accumulation value in Hough space, is the line enough close to the maximum number of arcs, and parallel to one tangent of each arc, as the real wall does. Figure 7 shows the increased feature detection rate after extending the tangents to surrounding parallel lines. The detected features (segments) are drawn back in white on the map, and it is shown that during the same scenario, extending the tangents to surrounding parallel lines increased the detection rate from one segment to four segments. Those segments are the features that are going to be used next for the SLAM Kalman filter update.

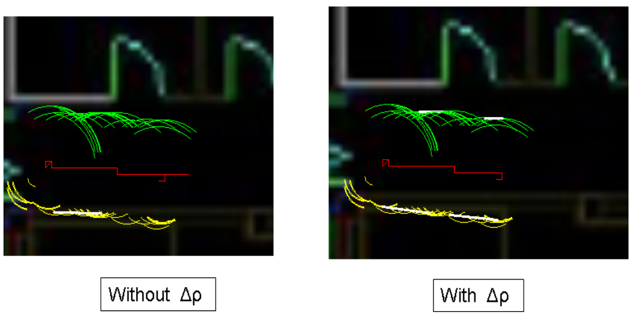


Figure 7: feature detection after Δρ extension

4.3 Hypothesis evaluation

To evaluate the detected line segments, we have used the length as criteria. The length of a line segment has to be at least 40 cm, so it can be considered as a detected feature, which is quit reasonable to eliminate small false detections. One can think about other evaluation criteria, especially by comparing the detected right and left line segments from a given position. For example, when walking inside a corridor the detected right and left line segments have to be parallel, and so if it is not the case, we can either reject the non aligned segment with corridor direction, or correct the direction of both segments to the average depending on the case. Of course this will include a corridor detection process, which will associate parallel detected line segments to a common corridor object. In our work we did not preceded in that evaluation direction simply because we were limited to use only the left sonar sensor. The main problem that we faced when working with both left and right sonar sensors was segment to wall association. We have to remember that the main idea of the work is to associate detected segments to previously detected walls, which corrects the pedestrian relative position to that wall. This means that segments detected in an uncertainty area (around a wall) will be associated to a same wall. Inside a corridor for example, both detected right and left segments are associated to a same wall, instead of being associated to the right and left corridor walls respectively, due to the intersection of large uncertainty areas for eventual segment detection of right and left walls. A possible solution is to separate walls detected by each sonar sensor to avoid this problem. However we have to exchange the wall landmarks (previously detected) of both sonar sensors once the pedestrian returns inside the corridor in opposite direction. A more extensive work is needed to deal with multiple sonar sensors. Nevertheless, even with one sonar sensor, some good results were obtained as presented next.

4.4 Hypothesis selection

The hypothesis selection process associates a detected feature (wall segment) to a previously stored landmark (a previously detected wall). If no association is possible, the detected segment will initiate a new landmark, or will simply be ignored in the following. Each extracted feature *seg* (line segment) from the previously presented hypothesis generation process is defined by the distance ρ_seg and angle θ_seg, the parameters of its corresponding line, plus two points *extremum1* and *extremum2* corresponding to both segment extremes. We take the previous stored walls (landmarks) one by one, starting from the oldest to the recent one, and we check if we can associate the detected feature to it. A wall landmark is also defined by the distance ρ_wall and angle θ_wall, the parameters of its corresponding line, plus *firstPoint* corresponding to the first detected point of the wall. The segment *seg* will be associated to landmark wall if:

- The difference between θ_seg and θ_wall in absolute value is smaller than 35 degrees.
- The segment has to be quit close to the wall (approximately 1m).

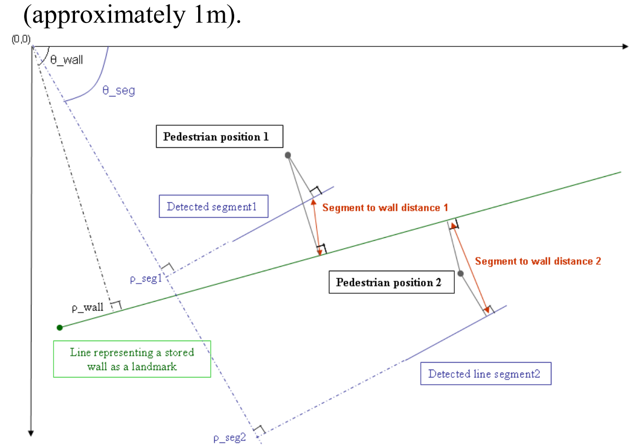


Figure 8: hypothesis selection process

To check the second condition, we project the pedestrian position on the detected segment and on the stored wall as landmark, respectively. The distance between the two projection points should be smaller than 1m. If no association is found after going through all landmarks, the feature (segment) will be used to initiate a new landmark: ρ_wall and θ_wall will be respectively set to ρ_seg and θ_seg, and *firstPoint* is set to the projection point of the pedestrian position on the segment. Figure 8 illustrates the hypothesis selection process. Two segment detection cases are presented. The two segments are equal in the angle parameter θ_seg, but differ in the distance to origin parameters ρ_seg1 and ρ_seg2. For each case, the two projections of pedestrian position on segment and wall are presented. The distance between the two projection points represents the segment to wall distance. *Segment1* is associated to the wall, whereas *segment2* (which is too far from the wall) initiates a new landmark, with *ρ_wall* equal to *ρ_seg2*, *θ_wall* equal to *θ_seg* and *firstPoint* set to the projection point of *Pedestrian position 2* on *segment2*.

4.5 Kalman filter SLAM implementation

The current state estimates for pedestrian location and known feature locations are stored in the system state vector $\hat{X}$, and the uncertainty of the estimates in the covariance matrix *P*, partitioned as follows:

$$\hat{X} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ . \\ . \\ . \end{pmatrix} \qquad P = \begin{pmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & . & \cdots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & . & \cdots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & . & \cdots \\ . & . & . & . & \cdots \\ . & . & . & . & \cdots \end{pmatrix}$$

$\hat{x}_v$ is the 3D person position estimate (x, y, z_floor) where z_floor represents the floor altitude, and $\hat{y}_i$ is the estimated 2.5D location of the ith landmark (wall), which corresponds to the position of its first detected point (2D coordinates and floor). $\hat{X}$ has 3(n+1) elements; where n is the number of known features. *P* is the covariance matrix, symmetric, with size 3(n+1)×3(n+1). $\hat{X}$ and *P* will change in dimension as features are added or deleted from the map. The 3x3 matrices like P_{xx} and $P_{y_1y_1}$ for examples (diagonal elements of the total covariance matrix *P*) represent the location uncertainty of the person and the feature y_1 respectively. The 3x3 matrix P_{xy_1} represent the correlation between the person and feature y_1 locations in all directions. A classical EKF-based Solution to Bearing-Range SLAM is used as detailed in reference [3]. The interesting point is the Bearing-Range measurements generation. In fact, the measurements are pointing to first points of stored wall landmark, which are part of the state vector as explained previously. The Bearing-Range measurements are combining INS and sonar measurements, and generated as following.

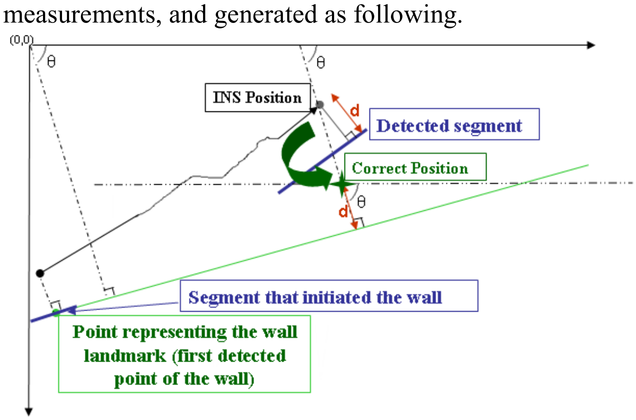


Figure 9: INS and correct pedestrian positions

Going back to the previous sections, we have detected a segment (feature) and got its association to the corresponding previously stored wall (landmark). Via this association, the pedestrian relative distance to the segment is considered to be its relative position to the corresponding wall, as explained in figure 9. In this scenario, and starting from the down left pedestrian position, a segment was detected that initiated a landmark (green wall). After few steps, and relaying only on INS increments, the pedestrian reached the *INS position* where a second segment fragment of the green wall was detected (hypothesis selection section). The relative distance of the pedestrian to the segment is *d*, and so we "know" that his relative distance to the wall is also equal to *d*, but he is standing too far from the wall. The INS system drifted from the *correct position* presented in the figure, and so we have to make a Bearing-Range measurement to the *firstPoint* of the wall (landmark) as if the pedestrian was at that *correct position*. In that way, the filtered position gathered after updating the sonar EKF state vector by this measurement will be closer to *correct position*.

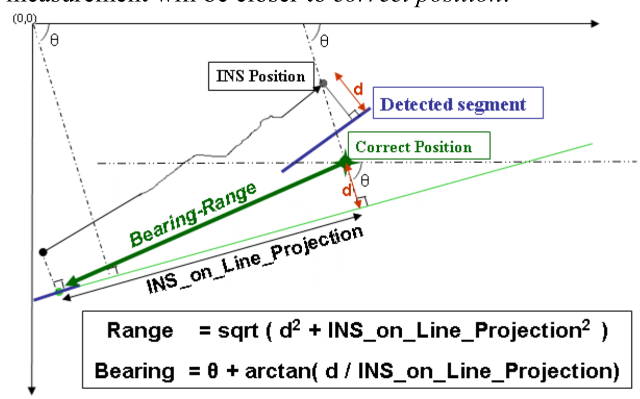


Figure 10: Bearing-range measurement

The Bearing-Range measurement of the landmark corresponds to the vector from *correct position* to the first point of the wall, as depicted in figure 10. It is to notice that we have to add 180° to bearing parameter if the pedestrian and coordinates origin are not on the same side of the wall.

5 INS heading correction

Before going on with this new point, let's recapitulate the actual situation:

- An INS EKF based on zero velocity updates is used to treat collected inertial data. Stance phases are used to separate the pedestrian displacement into steps. Starting from current position (a stance phase), we add INS increments to reach the next stance phase (one step displacement). It is there where sonar and INS data are coupled.
- The sonar data accumulated during the previous step are now treated to detect an eventual segment feature. Once associated to a previously stored wall (landmark), a bearing range measurement is generated using the relative distance to the segment and INS position

projection on the wall, and updating the sonar EKF gives back the filtered pedestrian position.

- From the filtered pedestrian position, we move to the next stance position, and the same cycle is redone.

The resulting problem of the current situation is illustrated in figure 11. The left picture shows sonar returns and extracted segments at current position, whereas the right one shows the up-to-now developed tracks: the white track corresponds to INS track, and the red one corresponds to the coupled sonar INS track. The pedestrian is moving in the upper direction. The INS heading is drifting to the left direction. To the left, after each segment detection, the sonar EKF is correcting the INS position back in the corridor, but starting from this filtered position, and using the INS increments to reach next stance position, we drift again from the corridor to the left. Consequently, the detected segments orientation is more and more drifting from the vertical direction of the corridor wall, and finally the detected segments are no more associated to the vertical wall. A new erroneous inclined wall is initialized as a landmark. The next filtered positions follow the erroneous inclined wall (right picture).

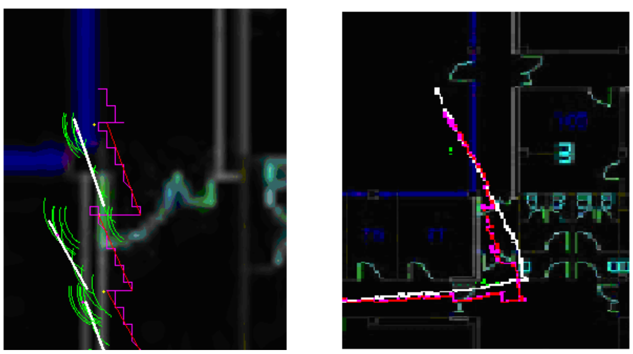


Figure 11: Heading drift

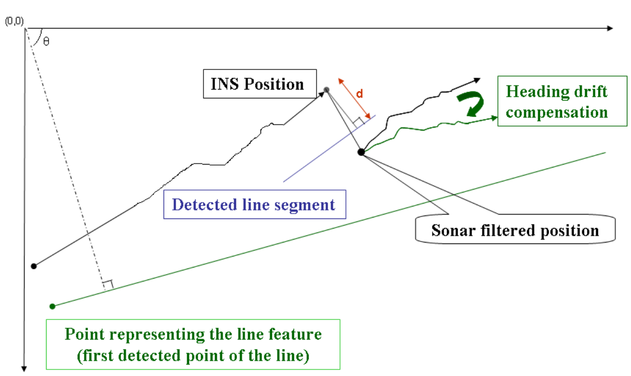


Figure 12: Bearing-range measurement

To overcome the INS heading drift problem, we introduced a heading correction: each time a detected segment is associated to a previously stored wall, we compensate the segment to wall direction drift next in the INS heading. Figure 12 continues with the same scenario depicted previously in figure 10. After segment detection, the sonar filtered position comes nearer to the wall. For the next INS step increments starting from the filtered position, INS heading is corrected to compensate segment to wall direction drift.

6 Evaluation

The user interface of the developed application offers mainly two output windows for evaluation: the track window and the sonar window. The track window sets as background the real map of the indoor test environment (floor). The user chooses the initial position of the tracked pedestrian on the map and the initial orientation of the on mounted inertial navigation system (left foot orientation). Two tracks are drawn on this window, the INS track (white) and the coupled sonar INS track (red). The real map of the indoor test environment in window background allows a direct visual comparison between the two developed tracks and the true trajectory of the pedestrian. The sonar window (to the left) is a zoomed window on the actual pedestrian position. Only the coupled sonar INS track is shown there. It is on this window where the arcs representing sonar returns are drawn, and where the corresponding eventual extracted segments are also presented, which gives a good idea about the status of the system. Figure 13 shows the initial situation for the first scenario. The track window (to the right) presents a partial view of the indoor map, where the true trajectory of the pedestrian is drawn in yellow, a round trip journey in four corridors of roughly 120 meters in total.

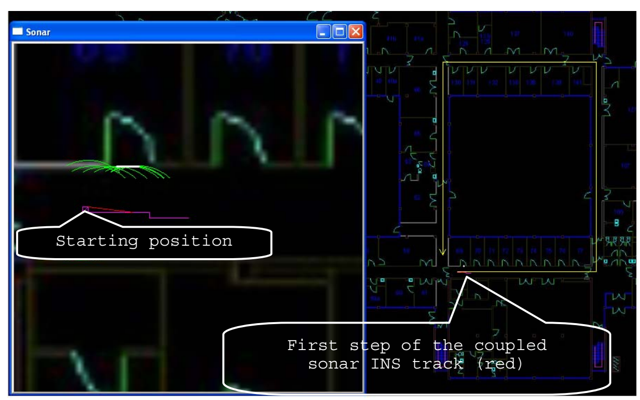


Figure 13: initial situation

The first step of the coupled sonar INS track is presented. The INS track is initially confused with the coupled sonar INS track, and will appear next in white in the track window. On the sonar window, the initial sonar returns are presented with the first corresponding extracted segment. This segment initiates the first wall *landmark0*.

Figure 14 depicts an interesting situation where another person walked aside of the tracked pedestrian. We can see the special sonar returns that were generated, and as expected the Hough transformation process was able to correctly extract a new segment by ignoring the non tangent arcs to the wall.

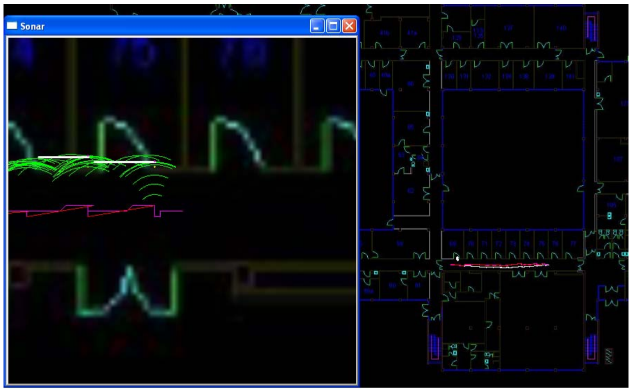


Figure 14: a person walking aside

Figure 15 depicts the final situation. The coupled sonar INS track ends with a position error of about 1 meter from the real pedestrian end position presented as a yellow point. The INS track ends with a position error of roughly 30 meters. This is mainly due to heading drifts accumulated during the walk as can be seen from the deformed white path compared to the rectangular journey. We have made another experiment which we are not going to present in details here, now with a close loop path. We were able to relocate the first detected wall landmark, reducing the location uncertainty of the pedestrian and all detected landmarks during the journey. The INS track drifted again suffering from accumulated heading drifts.

Figure 15: final situation

7 Conclusion

This work was a first attempt to use sonar sensors for pedestrian tracking. We have shown that extracting surrounding walls during a pedestrian walk can correct the heading drift, which is not visible by an inertial navigation system. The Hough transformation technique and arc presentation of sonar returns can overcome the beam width uncertainty of sonar sensors.

The pedestrian displacement constraint which was considered can be cancelled by breaking the dependency between INS and sonar positions: develop a separate localization system for on mounted sonar sensors rather than using INS position and last step direction. Heading correction can also be done by feeding back the sonar filtered position to the INS EKF as a position fix. Future work will focus on those points.

References

[1] Juan D. Tardos, Jose Neira , Paul M. Newman, John J. Leonard ,*Robust Mapping and Localization in Indoor Environments using Sonar Data*, The International Journal of Robotics Research 2002

[2] E. Foxlin, *Pedestrian Tracking with Shoe-Mounted Inertial Sensors*, IEEE Computer Society Press, 2005.

[3] Andrew John Davison, *Mobile Robot Navigation Using Active Vision*, Robotics Research Group Department of Engineering Science University of Oxford 1998

[4] David L.Hall, James Llinas, *Handbook of Multisensor Data fusion*, CRC Press, 2001

[5] Oliver J. Woodman, *An introduction to inertial navigation*, UCAM-CL-TR-696, ISSN 1476-2986, University of Cambridge, August 2007

[6] L. Ojeda, J. Borenstein, *Non-GPS Navigation for Security Personnel and First Responders*, Journal of Navigation, Vol. 60, No. 3, pp. 391-407, 2007

[7] Richard O. Duda,Peter E. Hart, *Use of Hough transformation to detect lines and curves in pictures*, Comm. ACM, Vol 15, No.1 pp.11-15, 1971

[8] Hugh Durrant-Whyte, Tim Bailey, *Simultaneous Localization and Mapping*, Australian Centre for Field Robotics (ACFR), 2006

[9] *http://www.xsens.com/en/general/mti*

[10] *http://www.acroname.com/robotics/parts/R145-SRF08.html*