IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 3, 2020, accepted February 22, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977501

A Hybrid Indoor Positioning System Using a Linear Weighted Policy Learner and Iterative PDR

WALTER CHARLES SOUSA SEIFFERT SIMÕES[1], (Associate Member, IEEE), WALMIR ACIOLI E SILVA[2], MATEUS MARTÍNEZ DE LUCENA[3], NASSER JAZDI[4], (Senior Member, IEEE), AND VICENTE FERREIRA DE LUCENA, Jr.[1,2], (Senior Member, IEEE)

[1]PPGI, Federal University of Amazonas, Manaus 69077-000, Brazil
[2]PPGEE, Federal University of Amazonas, Manaus 69077-000, Brazil
[3]CTC/LISHA, Federal University of Santa Catarina, Florianópolis 88040-900, Brazil
[4]IAS-Institut für Automatisierungstechnik und Softwaresysteme, University of Stuttgart, 70174 Stuttgart, Germany

Corresponding author: Vicente F. de Lucena (vicente@ufam.edu.br)

This work was supported in part by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) under Grant 001, in part by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), in part by FAPEAM (Fundação de Amparo à Pesquisa do Estado do Amazonas) through scholarships for the graduate programs of the Federal University of Amazonas (UFAM), Brazil. Additionally, it was funded by Samsung Electronics of Amazonia Ltda, under the terms of Federal Law n° 8.387/1991, according to for in Article 48 of Decree n° 6.008/2006, through grant agreement n° 004, signed with CETELI (Center for Research and Development of Electronic Technology and Information) supporting the graduate program of Electrical Engineering-at UFAM.

ABSTRACT Electronic indoor positioning systems deal with the combination of sensors, actuators, and computational algorithms for precisely locating subjects, delivering navigation directives, and keeping track of particular objects. The main factors considered for the construction and evaluation of these systems are the localization accuracy and the time spent to calculate and deliver this information. The challenge in developing successful positioning systems is to find a tolerable relationship between those factors. In this proposal, after a careful analyses of related works, we associated different methodologies and technologies to construct a hybrid positioning model that uses a mapping algorithm called Linear Weighted Policy Learner, a navigation model called iterative Pedestrian Dead Reckoning (which uses the Kalman filter to deliver real-time location), and an obstacle detection algorithm that combines sounds and stereo vision sensorial capabilities. The adopted choices were based on the published state-of-the-art, and comparisons of the obtained results showed that our system is accurate and fast enough to be very competitive with the current stage of the technology.

INDEX TERMS Indoor positioning system, indoor localization system, pedestrian dead reckoning, Kalman filter, particle filter.

I. INTRODUCTION

The computational models of locating people and objects in outdoor environments have resulted in numerous applications such as navigation systems, tracking, etc. [1]. However, the widely used and precise outdoor Global Positioning System (GPS) is still impaired by internal building structures due to services that affect its satellite information reception process. That was discussed by Nebel that developed a study on the utilization of GPS, considering diverse conditions, using, directly and indirectly, obtained coordinates, intermediated by a signal repeater to observe its limitations and the real possibilities of application in indoor positioning systems [2].

Other strategies tried in various studies had as a goal to overcome the use limitations of GPS in indoor environments and to make the Indoor Positioning System (IPS) more accurate. The margins of error presented by approaches of the use of GPS in indoor environments were around 20.00 m, considered very high value, as indicated by Chu [3]. The signal transponder interposes the external GPS signals to the building structure, reducing the error margin to 10.00 m. However, its use has increased the time it takes to process

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

and identify a place [3]. Many research groups still working to obtain accessible indoor GPS.

Several other groups look for techniques and technologies that could make IPS more robust and provide accurate information in many types of scenarios, especially environments that affect data obtained from mono sensory systems [4]. The increase in the indoor positioning model's complexity is a new problem to be solved concerning the relationship between the level of accuracy and the acceptable time for information delivery, which must consider the limitations of the adopted devices [5].

In simpler systems, linear mathematical formulations require less processing and thus adopt approximation criteria requiring the acceptance of error margins, as indicated by the application of Alpha-Beta (ABF) and Linear Kalman Filter (LKF) [6], [7]. In non-linear systems, the more complex calculations adopted deliver better results of the reference values, however, requiring more time to reach them, as seen in the applications of particle filters [3], [8].

Hybrid systems have gained more considerable attention because they combine data fusion techniques and allow their use in scenarios with different characteristics [9]. However, many of the hybrid indoor positioning systems face as one of the main problems the search for the balance between performance and processing time [10], [11]. Each approach has the advantages of using it in its separate or combined application, requiring knowledge of its characteristics and the most appropriate context for its use [4].

This study presents an indoor positioning solution that deals with hybrid features, aiming to increase the final precision of the system, keeping the processing time within a tolerable scale for human navigation systems. We assume that the recognition of these human activities in indoor environments can provide a range of opportunities in location, navigation, and tracking services, as is already available in outdoor environments [12]. Our system maps a set of indoor human activities, such as moving around, going up and down the stairs, going through doors, or turning to the right and left sides. Besides these activities, it identifies objects arranged along a path, whether they are fixed or temporarily left in these places.

This paper presents this research topic for discussion, with the proposal and results in the following organization: Section 2 discusses the strategies and results of closely related works, the goal is to increase the understanding of the limits of the successful approaches. Section 3 shows the conception and the limitations of the proposed hybrid positioning system. Section 4 contains the test protocols and configuration of the test cases, as well as an evaluation of the obtained results. Section 5 summarizes the conclusion of this work and presents future approaches.

II. STATE OF ART

The related works are divided into four parts. The first part describes various approaches using Wi-Fi sensors. In the second part are the works that used inertial sensors. Part 3 brings work that applies computer vision to targeting, navigating, and tracking. Finally, part four describes work that used combinations of hardware and software to increase the accuracy of indoor positioning systems and shorten processing time.

A. WI-FI-BASED LOCATION

Wi-Fi sensors were the first devices used in the Indoor Positioning System approaches to replace GPS and provide reference values for target location [2]. The choice of the Wi-Fi sensor was motivated by the high availability of this type of sensor in the most diverse industrial, commercial, and residential environments [2].

Initial approaches used raw Signal Strength Indicators (RSSI) data to record addresses of interest and stored them in map schemes to allow the displacement of the target from one place to another as described by Liu, Waqar, and Bolat [10], [12], [13]. The results indicate that the scenario influenced the values of Wi-Fi caused an average error of 3.57 m when used alone and 1.74 m with the use of the fingerprinting scheme. Variations of the RSSI signal collected from the same address cause oscillation of the positioning error margins to vary between 1.00 m and 3.00 m, even when data is driven by redundant fusion schemes, which require a larger volume of data to indicate a location, as shown by Waqar [12].

Liu showed that increasing the amount of Wi-Fi nodes also allows a reduction of the margins of error, however, it is necessary to investigate this quantity so that there is no signal interposition and increase the volume of processed data [10].

Li focused on the application of signal-to-noise ratio (SNR) and noise reduction filters on the raw RSSI signals to discard the very divergent values of the mean of the set and at the same time, reduce the set to a sample space capable of identifying each internal address in a map [14]. The error margins obtained in the experiments were 2.40 m for the RSSI, 2.10 m for the SNR, 3.30 m for the noise level, and 1.40 m for the RSSI and SNR combination. These values show that the use of the SNR reaches hit levels very close to those of the RSSI and that the combination of the two decreases the oscillations of the obtained values. The noise level presented the worst result in the tests performed. Fingerprinting mapping further showed the need to be rebuilt when there is some change in the scenario or when the arrays of the sensors and records are changed.

Wi-Fi-based location systems tend to vary significantly in their information, which justifies the search for other sensor models for IPS projects. Inertial sensors have been tried as an alternative to locating targets in indoor environments [4]. The inertial sensor, also called inertial measurement unit (IMU), represents the combination of several sensors such as the magnetometer, gyroscope, accelerometer, and barometer [13]. These sensors have started to be offered in a variety of devices, such as smartphones, video games, and TV controls due to their precision levels and the low cost of processing and power consumption [15].

B. INERTIAL BASED LOCATION

Kealy and his co-authors addressed the information availability in indoor positioning systems concerning demand (when required) and continuity (always available) [16]. The author opted for the use of the inertial sensors because no physical intervention is necessary for the environment, being its implementation more like the projects that use the GPS for location outdoors. The results showed that the best result had an average error of around 0.20 m. This value is achieved through three factors: (1) detection capability of the sensors used, (2) physical arrangement of sensors, and (3) robustness of model integration. The viability of a positioning system is directly affected by the detection capability of the sensors used. System performance is directly affected by the adopted physical arrangement of the sensors. Performance depends on the robustness of the integration of the algorithms and their data.

Yan also used the inertial sensors as a reference for his indoor positioning system [5]. The author chose as strategy the use of twelve associated inertial sensors, applying a redundant data fusion to indicate the location. The approach used in the project intended to keep the data volume high, without increasing the time needed to process and present the results. The tested system used a fixed route for the planned navigation, hitting about 83.18% of the predefined markings with an average margin of error of 0.50 m. The use of more associated inertial sensors allowed the positioning system to present greater precision and stability, even in noisy environments.

Inertial systems perceive more subtle changes of position than Wi-Fi networks, but still show stable oscillations in places where there are sources that interfere with reading the Earth's magnetic field, even when using many combined sensors, as in the work of Yan [5]. The visual positioning system is an alternative to providing a location where inertial and Wi-Fi networks are affected [17]. Visual positioning systems use images captured by one camera (monocular model) or multiple cameras (stereo image), which provide a large amount of data, and it is necessary to establish criteria for the removal of spurious data to achieve a balance between the accuracy of level and time spent on visual recognition [18].

C. COMPUTER VISION-BASED LOCALIZATION

Pradeep built an indoor navigation system based on the stereo vision to guide the visually impaired in pre-mapped places, acting as a visual odometry system [17]. The prototype built for use on the user's head has captured a higher volume of information that allowed avoiding collisions with tall objects not previously perceived by devices used at lower altitudes. Visual localization system activates micromotors arranged on the user's shoulders and waist to indicate their movement orientation. The navigation model adopted in the tests was the Simultaneous Localization and Mapping (SLAM), which assembled a 3D data map, associating the map records in a neighborhood scheme. The visual positioning system was able to identify objects at a distance between 0.30 m and 2.00 m, captured in images received at a maximum speed of 14 frames per second.

Xue presents a study on the construction of a visual indoor location, with the focus on reducing mapping time, so that its use is not an impediment factor [19]. The author proposed the construction of maps extracting images directly from the video streaming, reducing the samples to a set of 100 images per record by the algorithm RANdom SAmple Consensus (RANSAC). The algorithm Speeded-Up Robust Features (SURF) was used to quickly construct the markers in a similar way to the Haar model; however, disregarding the angulation and lighting variations. The accuracy level obtained in the tests was 70%, resulting in a maximum error distance of 2.00 m to the location reported by the system.

Yuke and his co-authors have developed a mapping and visual navigation using an information reinforcement process based on a Weighted Policy Learner (WPL) [20]. The WPL reduces the complexity of the fingerprinting model in constructing the relationships between the mapped records of the scenario and the possible recognition during navigation through an auto-critical system, which collects a sample of the received data and applies a training [21]. The place, which had 20 rooms, was mapped in a simulator and the data submitted to the TensorFlow environment for training [22]. The data were provided to a C-Net network for the identification of features and learning [22]. The training lasted about 1.3 hours and had a margin of error ranged from 0.4 meters to 0.7 meters. These location variations are caused by the small amount of data used in training but have already generated a high wait time when considering real-time systems.

Systems based on visual recognition also have limitations such as light changes, viewing angles, and need to deal with a large amount of data and have high computational complexity [18]. Approaches that relate the use of two cameras improve the perception of the environment by bringing distance information through depth maps and identifying the scenery from a 3D perspective [23]. Many authors have associated different sensors and data in hybrid arrays to make the system more robust, making one subsystem reinforce the other, or at worst, maintain at least some reliable information about the location of the user or markers recorded in the scenario mapping [3].

D. HYBRID BASED LOCALIZATION

Bolat proposed a hybrid navigation solution for pedestrians in indoor locations, associating Wi-Fi signals to magnetic fields through a particle filter [13]. The objective was to create a collaborative scheme in which one technology compensated for the deficiency of the other [13]. A particle filter processed the data fusion and delivered the data to a fingerprinting algorithm, which relates the records to a map. The mapped registers were consumed by the Pedestrian Dead Reckoning (PDR) algorithm to aid in navigation on a reference route, with a distance of 65.50 m. The tests consisted of collecting

the route data during a ten-fold cycle to verify the behavior of the localization algorithms. The results obtained showed an average error rate in the first two laps was 4.80 m and a reduction to 0.45 m after the 10th lap.

Another author who used a hybrid arrangement of sensors and algorithms to form an indoor navigation system was Leppäkoski [11]. RSSI signals indicated a larger area, and the inertial information distributed in each region of the Wi-Fi reinforced the environmental perception, forming a more robust position identifier. The data received from the inertial and Wi-Fi sensors have different characteristics, and therefore it was necessary to use the complementary and Extended Kalman filters (CEKF) and a particle filter, which use the principle of propagation of state like CEKF. These filters are implementations of nonlinear Bayesian algorithms capable of merging redundant and complementary data. The margins of error obtained in the test results were 12.00 m in the Wi-Fi navigation, 5.00 m using PDR only, 4.00 m in the CEKF, and 3.00 m using the particle filter.

Chen associated the visual and inertial perceptions in his hybrid navigation system [24]. The Kalman Filter (KF) associates the visual depth map data generated by the red, green, blue, and distance (RGBD) camera with the inertial information to provide a 3D perspective of the recorded position. The KF algorithm also reduced the accumulated errors presented by the PDR algorithm during navigation. A total of 937 reference points was defined along 61.00 m, and a small robot developed for the tests, walked the designated path, observing the coordinates reached along the way. The error margins averaged 0.40 m as compared to the standard values recorded for each place.

Hanchuan proposed a hybrid indoor location solution, associating RSSI signals, provided by Wi-Fi base stations and RFID devices distributed throughout the environment, with visual information captured by the smartphone camera [9]. Hanchuan fixed the visual ground labels to reinforce an address, thus increasing the confidence in the data generated by the system. Such operation was done to overcome the oscillations of the Wi-Fi signals The positioning scheme associated two Wi-Fi base stations to indicate the distance of the mobile device to each of the bases and visual perception, within a range of 0.50 m and 2.00 m. The use of Wi-Fi signals only showed a mean margin of error of 3.00 m, while the hybrid system decreased and stabilized the error by 1.00 m.

Unlike the works described and discussed, which usually use the standard model to identify the user when stopped or moving, in addition to recording a lot of scenario data, this work presents a solution classified as a hybrid internal positioning.

The proposed system combines data captured by distinct sensors and mathematical approaches to allow the localization of visually impaired people indoors, making the system hybrid. Another aspect that helps us to classify the system as a hybrid is the use of autonomous methods such as proximity, PDR, and Kalman algorithms, as well as the use of training-dependent methods such as particle filter and visual analysis.

The sensors used to obtain environmental information are of the absolute and relative type to provide the system with a better understanding of the various characteristics of the registered location. In absolute positioning systems, there is the use of external devices and resources arranged in the scenario, such as Wi-Fi-based systems and visual information [12], [18]. A relative positioning system does not require the use of an external source, taking inertial systems as an example [4]. Another reason to choose the Wi-Fi, inertial, and camera sensors is to use the sensors adopted in most IPS approaches and take advantage of all available literature. The next sections present the details of each step and its subsystems.

III. DEVELOPMENT OF THE HYBRID IPS PROPOSAL

The proposed model has hybrid features in its two main modules, which describe the tasks required to construct the mapping and navigation process. Figure 1 shows a general architecture model of the indoor positioning system.

The proposed indoor system architecture utilizes data reduction by pre-fusion sampling to improve information quality, reduce data processing time, and maintain the right level of localization. Each subsystem works independently, having its data manipulation method capable of constructing a site representation according to the indication of the gain of its use in a decision-making process. Each step is detailed below, indicating its methods and algorithms.

A. PREPROCESSING

Data preprocessing represents the step where the algorithmic interventions on raw data are applied, improving their representations to reach the expected standards for marker definition. The objectives of manipulating the data before submitting to redundant and complementary fusion algorithms are to enhance data quality and reduce the time taken to identify the pattern of data received.

In general, this step indicates performing two main tasks: sensor type identification and feature extraction algorithms, as detailed below. Sensor identification allows you to direct data to the most appropriate algorithms. Feature extraction algorithms perform pattern search functions and use the data to fuse verified patterns.

1) SENSOR IDENTIFICATION

The sensors are physically connected to a central processing unit, where they send their data to be processed and used in the location and indoor navigation operations. The identification of sensor types occurs by inserting an ID with values 1, 2, and 3. These values identify the signals and data of the Wi-Fi, inertial, and visual locations, respectively.

The data can be requested from two sources: the mapping and the navigation, using a second ID. When the mapping requires the data, the ID triggers the sample reduction module. When browsing requires the data, the ID triggers the linear data fusion module.

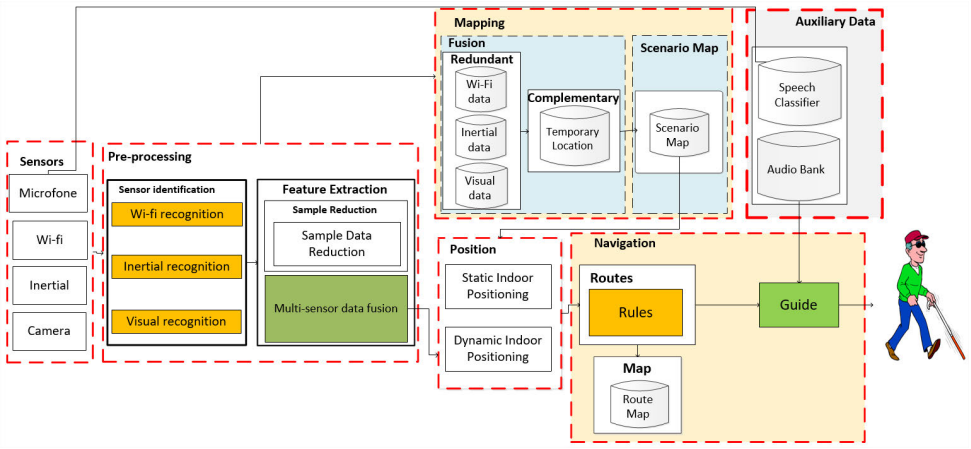


FIGURE 1. Indoor mapping and navigation system architecture.

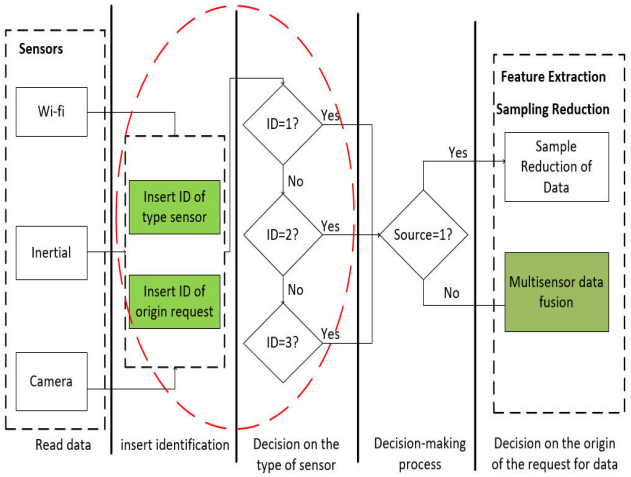


FIGURE 2. Sensor identification flowchart and data delivery for data fusions.

After receiving the data and inserting the sensor identifications, a decision algorithm observes the values of the IDs and directs the data to its proper processing, as indicated in Figure 2.

2) FEATURE EXTRACTION

Feature extraction has two functions: reducing the sample space used in mapping preparation, and multisensory data fusion, which applies linear processes to provide information in the navigation process.

Feature extraction uses a set of data obtained from the sensor type identification and applies a weight-based sort order to define a sample, providing a small subset of relevant features.

3) SAMPLE REDUCTION

The strategy used to reduce processing time while maintaining the robustness level of the marker was to select representative samples of the raw data of each sensor type. The algorithm for this data decrease in a robust sample was the RANSAC probabilistic model.

RANSAC randomly selects a subset of data samples and uses it to estimate model parameters. It then determines the samples that are within the error tolerance of the generated model [18]. We consider these samples as agreed with the generated model and called a consensus set of the chosen data samples [25].

Here in this paper, the data samples in the consensus behaved as inliers and the rest as outliers by RANSAC. If the consensus sample count is high enough, it will train the final consensus model to use them as the valid data set for the system. RANSAC repeats this process over several iterations and returns the model that has the smallest average error among the generated models [25].

Because it is a random algorithm, RANSAC does not guarantee to find the ideal parametric model concerning the inliers subset data. However, the probability of reaching the ideal solution is higher than using the raw data.

The RANSAC model used in this work allows the robust adaptation of a $y = f(x; \alpha)$ model to an S data set containing inliers and outliers. In this model, the restriction on the amount of data used and maintained in the collection of inliers is given by a limit value, randomly established by RANSAC. This process is repeated until it reaches the highest value, considered here as the best sample set of the original data set [19].

The planning of the RANSAC parameters considers some criteria before its use, such as:

1) The probability of choosing an inlier is obtained by the relation $Prob = \frac{\#\ of\ inliers}{\#\ of\ samples}$;
2) The construction of the sample set is a probabilistic process, obtained by $Prob^n$, where n is the available amount of sample data;
3) The probability of incorrect construction of a model during k iterations is $(1-Prob^n)^k$;
4) The success rate of RANSAC p is obtained by the ratio given by formula $(1-Prob^n)^k = 1-p$, where $k = \frac{\log(1-p)}{\log(1-Prob^n)}$.

Figure 3 shows the sample reduction process, starting with data collection and ordering until RANSAC is applied.

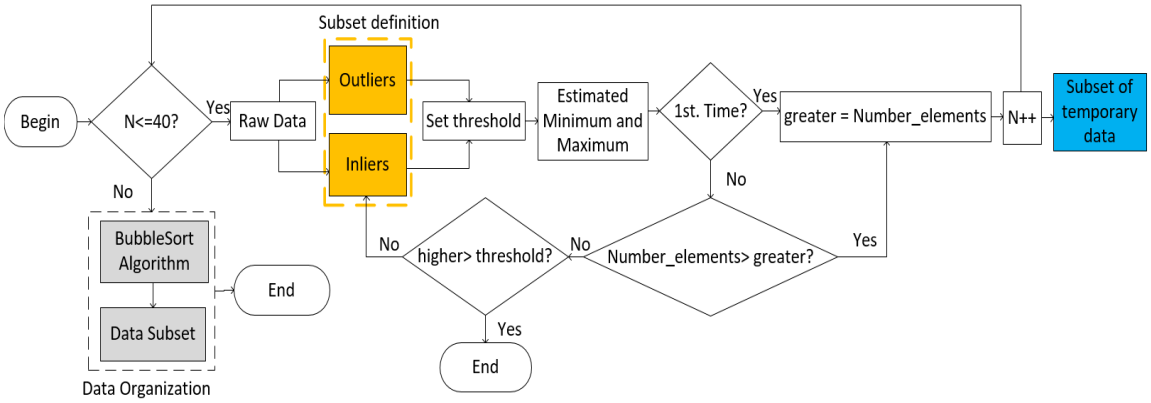


FIGURE 3. Flowchart of the process of creating the data subsets generated by RANSAC and the ordered final sampling.

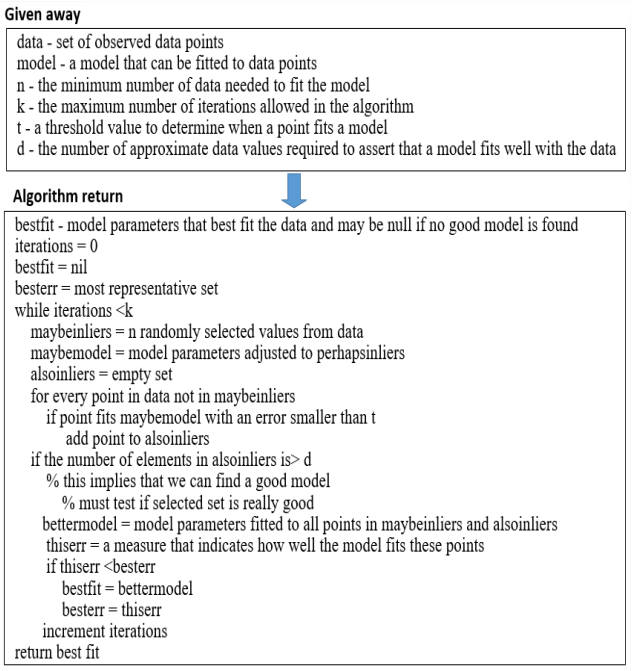


FIGURE 4. Flowchart of the algorithm used in data selection by RANSAC.

To make it clear how RANSAC was created, the algorithm flow is described below in Figure 4.

4) MULTISENSOR DATA FUSION

The multisensory data fusion represents the application of a Particle filter on the data and the fast delivery of the processed values to the navigation subsystem. The particle filter links the data received from the sensors to the data obtained from the iterative Pedestrian Dead Reckoning (i-PDR) associated with the linear Kalman filter.

The particle filter tracks the variables of interest, classifying each observed value according to the weight given by the importance of the particle. The particle filter selects and organizes in an iterative process all the data in a new set,

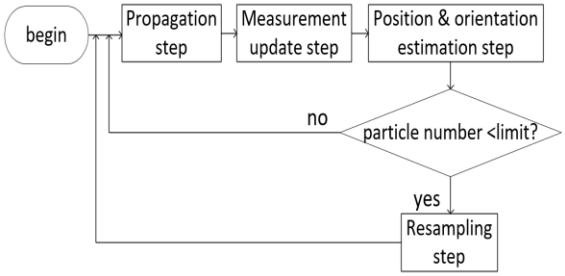


FIGURE 5. Particle field operating flowchart.

using as a criterion the weight, which indicates the highest probability of identifying a particular location, thus avoiding the low representativeness of the data [8].

The algorithm has recursive functions so that the most representative data are submitted in the prediction and update stage of the system data. After each iteration, all particle is modified according to the existing model (prediction stage). The inclusion of error margins allows the system to adjust to the noise received by the variables of interest, re-evaluating the data based on the last reading, and updating when necessary (update step).

The proposed particle filter has four steps: propagation, measurement update, position and orientation estimation, and resampling, as shown in Figure 5.

The propagation step represents the position update for each particle. Both the length of the pitch l_t and the direction orientation θ_t are passed by the PF component and assumed as data modified by Gaussian random noise. Hence the new location and orientation direction of the particle i^{th} at time t are

$$\theta_t = \varphi_t + \varepsilon \tag{1}$$

$$x_t^i = x_{t-1}^i + (l_t + \delta)\cos(\theta_t) \tag{2}$$

$$y_t^i = y_{t-1}^i + (l_t + \delta)\sin(\theta_t) \tag{3}$$

where δ is the Gaussian noise for the length of the pass, and ε is the direction of the direction.

The measurement update step is the phase responsible for the correction of the weights of the propagated particles. First, the particles are collected during time t. Particles that move away from the most representative group are neglected. The other particles indicate the most likely correct location. The particle weights are updated by Equation (4):

$$w_t^i = \frac{w_{t-1}^i}{\sum_{i \in P_t} w_{t-1}^i} \tag{4}$$

The center of mass of the particles is calculated and compared with the previously estimated positions, thus indicating an estimate of position and orientation.

Using this random inliers data limit value, a decision process checks whether the particle quantity has reached the maximum amount to conclude the process or continue running the process when the quantity is below the allowed limit.

The resampling step begins the process by eliminating particles weighing zero. All surviving particles generate a new filtration process under Gaussian distributions with their weights. The weighted center of all particles provides an estimate of the user's current position.

B. MAPPING

The mapping represents the construction of the markers and the application of neighborhood relation schemes between markers, to facilitate the location of a specific position and the building of routes.

Marker term is the terminology given for the representations of the physical (sensors) and logical (mathematically defined) registers inserted in the map. The presence of noise reduces the accuracy of the system and may make it unfeasible in certain situations [1]. These representations must possess characteristics that allow their perception under the highest possible conditions and the circumstances [26].

The construction of the markers uses the data provided by RANSAC and apply filters and fusion algorithms that combine the data according to a standard. Fusion algorithms initially treat data according to their source (Wi-Fi, inertial, and visual), creating their markers. The complementary fusion uses the processed data from the redundant fusions and provides a hybrid marker.

Once the Wi-Fi-based marker construction schemes, inertial sensors, visual information, and the combination of them are defined, here called the hybrid marker, the next step is to create the map. We construct the map using proximity relationships between markers through neighborhood algorithms. We detail the whole scheme for defining markers and the map below.

1) CONSTRUCTION OF THE MARKER BASED ON WI-FI SENSOR

Locations mapped by the Wi-Fi-based model use the received signal strength indicator (RSSI) captured by the sensor available on the wearable device.

The RSSI values suffer substantial impacts, which can cause refraction, reflection, diffusion, and dispersion of the radio waves, which may make it unfeasible for the localization task.

The construction of Wi-Fi markers applied a filtering noise called SNR (Signal-to-Noise Ratio), which allows separating the foreground signals (desired information) and background (undesired signals) [14]. The SNR formula is:

$$\text{SNR} = \text{signal} - \text{noise} \tag{5}$$

The location by RSSI and SNR is given by the trace of the radius between the access point (AP) and the measurement place. This measurement uses the received Wi-Fi value to indicate in the digital map the cell corresponding to the real environment, considering the size of the cell in the digital map, the cell size in the actual place, the signal strength emitted by the AP, path loss exponent n, reference distance, and shading. The prediction model of RSSI is:

$$P_R = P_T - P_{Lo} - 10n \log\left(\frac{d}{d_0}\right) - \sum_{j=1}^{N_w} k_{wj} w_2(\varphi) \tag{6}$$

where P_R is the power received (dBm), P_T is the transmitted power (dBm), P_{Lo} is the propagation loss at a reference distance from the AP (dB). The variable d is the propagation distance (meters), n is the loss exponent, k_{wj} is the number of walls of type j, w_2 is the empirical loss of propagation due to walls of type j (dB), and N_w is the number of walls between the transmitter and the receiver.

The power received information (P_R) was used as an indication of position, but the signal strengths showed jumps that made identification challenging. The use of the Kalman filter softened the signals and decreased the distances between the signals captured in the same position.

The formulation that indicates the distance and location of the origin of Wi-Fi signals is obtained by:

$$Z_t = \sum_{i=1}^{N} w_t^i (x_i, y_i) \tag{7}$$

where w_t^i is the weight of the router i at time t, and (x_i, y_i) refers to the location of the router itself.

2) CONSTRUCTION OF THE MARKER BASED ON INERTIAL SENSOR

The creation of an inertial marker uses data obtained from the three axes *(x, y, z)* of the gyro, magnetometer and barometer sensors, which indicate the direction based on the Earth's magnetic axis, the angular position in relation to the ground and altitude, based on the variation of the atmospheric pressure [4], [16]. The gyroscope, magnetometer, and barometer sensors, when combined with the accelerometer sensor, form an arrangement known as the Inertial Measurement Unit (IMU) [5].

The combination of the axes occurs by the application of a fusion combination using the Roll-Pitch-Yaw (RPY)

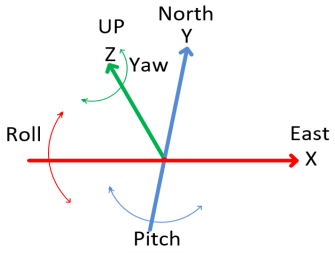


FIGURE 6. RPY system.

reference system as a model [4]. The RPY system considers that the x-axis (Roll axis) should point to the forward direction of the user's movement, the y-axis (Pitch axis) must be orthogonal ax pointing to the right, while the z-axis (Yaw) points downwards, giving a vertical orientation, as represented by Figure 6 [11].

The observation of the inertial data in some scenarios, however, can have strong oscillations caused by numerous factors that interfere in their values, and which reflect in erratic behavior, increasing inertial system error [4].

A particle filter implementation called the magnetic field compensates for perceived variations in magnetometer readings to define a standard [8]. Magnetic fields are microregions that increase the area of perception of the inertial sensors, reducing the disparities of the identification of an address but bring an increase in processing time [13]. To limit the captured inertial values, already considering the imperfections and influences of the place in the data, a margin of error between the collected data of 0.01 m was established.

In the particle propagation stage, a magnetic field is defined by aligning the magnetic data, m, which are confronted with a reference position r_{ref} and a reference orientation $\psi_{ref,t}$, which is expressed as

$$\psi_{ref,t} = \tan^{-1}\left(\frac{r_{ref,y,t} - r_{ref,y,t-1}}{r_{ref,x,t} - r_{ref,x,t-1}}\right) \tag{8}$$

where $[r_{ref,x,t}, r_{ref,y,t}]^T = r_{ref,t}$, and t is the index of time.

3) CONSTRUCTION OF THE MARKER BASED ON VISUAL INFORMATION

Visual markers combine fiducial and natural information to bring greater robustness to the reading and recognition process and to standardize visual information. The fiducial information represents the mapped object in its complete form, while the natural data represent textures and other elements [26].

The visual marker composition scheme applied the following elements: black outer edges to limit the area of the figure, an edge code to indicate the reading orientation, and a center object. Figure 7 shows its visual appearance.

The visual recognition algorithm applies a learning process to construct a marker, covering the maximum of visual variations to facilitate the decision on when the images represent

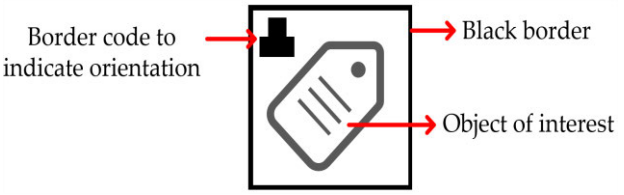


FIGURE 7. Visual marker model.

the same information or discarding those that do not represent any marker [27].

The definition of the visual marker occurs in an unsupervised way in a decision tree of the Haar-like Boosting type, where the classifier receives the objects and, from different parameters, allocates them in different classes to construct its discriminant function [28]. During the unsupervised learning process, the parameters of the Boosting tree are modified to adjust the discriminant function better, making the activity time-consuming and high computational cost. Although the AdaBoost model has a slow training, it presents good robustness and speed in the recognition process [29].

The construction of a Haar-Like classifier requires a set of images called positive, containing the desired object, and a set of negative images containing other visual information [30].

The construction of the classifier follows the scheme shown in Figure 8.

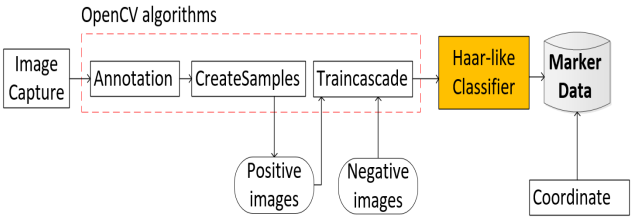


FIGURE 8. Construction of visual classifiers.

The Annotation, CreateSamples, and Traincascade functions, belonging to OpenCV, are used in the construction of the visual classifier, defining the standardization of brightness, illumination, and size of images for training and generation of the most stable Haar-Like classifiers [29].

The annotation algorithm is an image editor that selects and isolates the desired object from the set of positive images, creating a file in TXT format with the annotations of the cut object's coordinates [30]. Figure 10 shows the tool's parameters, indicating the options 'c' to accepted, 'd' to delete the latest selection, 'n' to proceed with the next image of the dataset, and esc to stop. Figure 11 shows an excerpt from the TXT file built by the Annotation tool containing 9 manipulated figures, indicating that only one object was selected from each image and its respective coordinates.

The CreateSamples algorithm standardizes the visual elements to be trained in a vector structure, removing the irrelevant information that causes delay and decrease of precision [30]. The parameters of the CreateSamples algorithm are detailed in Table 1.

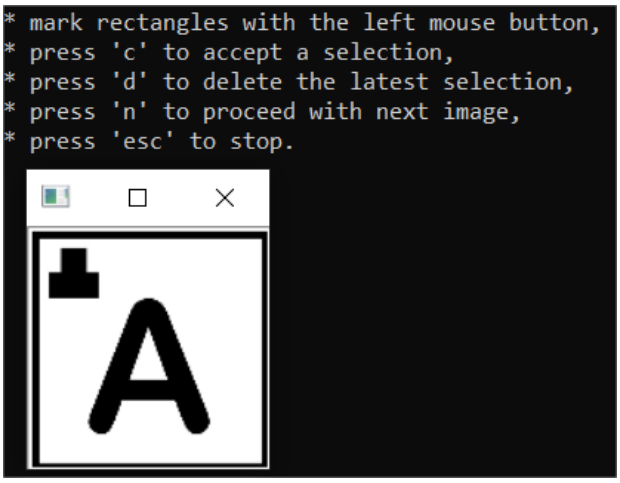


FIGURE 9. Selection of visual information by the annotation tool.

```
\positive\1.jpg 1 167 82 176 176
\positive\2.jpg 1 198 75 145 183
\positive\3.jpg 1 198 75 142 180
\positive\4.jpg 1 5 5 335 250
\positive\5.jpg 1 5 5 117 184
\positive\6.jpg 1 122 71 77 118
\positive\7.jpg 1 195 75 156 170
\positive\8.jpg 1 0 4 195 71
\positive\9.jpg 1 213 75 31 151
```

FIGURE 10. Registration of coordinates of objects selected by the annotation tool.

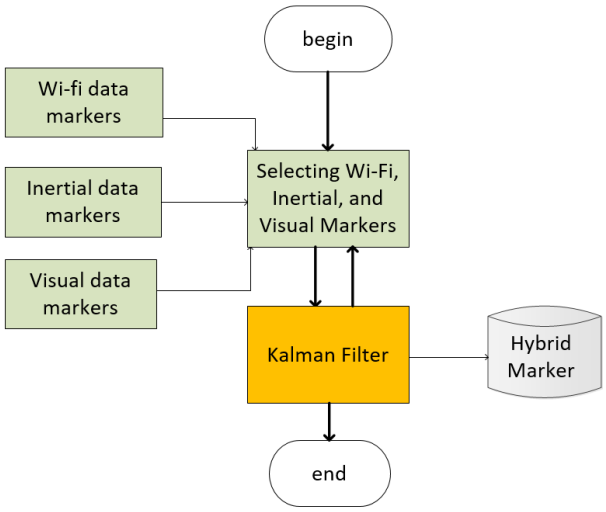


FIGURE 11. Diagram of hybrid marker definition.

Traincascade is a standalone algorithm that searches and allocates data in the AdaBoost tree, considering that each node is a subtree [26].

Adaptive Boosting (AdaBoost) is a machine learning algorithm used to increase the performance of other learning algorithms, such as decision trees [25, 26]. The algorithm gives an adaptive stimulus in the sense that subsequent classifications

TABLE 1. OpenCV CreateSamples parameters.

Parameter	Value	Function
bgcolor	0	Background color, parameterized in grayscale.
bgthresh	0	background_color_threshold.
maxxangle	1.1	max_x_rotation_angle.
maxyangle	1.1	max_y_rotation_angle.
maxzangle	0.5	max_z_rotation_angle.
maxidev	40	Maximal intensity deviation of pixels in foreground samples.
w	30	Width of the output samples.
h	30	Height of the output samples.
Info	Maker.txt	The source file of selected objects.
bg	tmp	List of used images as the background.
vec	pos.vec	Output file of the positive samples for training.
num	12	The number of positive samples.

made in a learning process fit instances negatively classified by previous classifications, making them less susceptible to loss of generalizability after many overfitting patterns have been learned when compared to most machine learning algorithms.

We used this feature of the AdaBoost algorithm on the decision tree based model, accessing the data learned in each classifier, reducing the learning time [25]. Thus, AdaBoost generates a weak classifier for each feature and combines these various weak classifiers, with a performance of about $50\% + 1$, to provide a robust sequential classifier, hence the name cascade [25], [26].

The parameters of the AdaBoost Traincascade algorithm are detailed in Table 2.

For the definition of the alarming rate, it was considered that in 50 negative samples (images without the presence of the object being recognized), the system should incorrectly detect about 10 samples, setting the false alarm in 0.2, calculated by the ratio 10/50. This value (0.2) allows the classifier to present a good generalization about the object learned, avoiding problems of overfitting and overtraining, which indicates a specialization in training [26]. Each image receives an equal weight at the beginning, and with the evolution of the processes, the weights are increased to the incorrect images. The process runs until the required accuracy and error rate are met, or the required number of resources is encountered.

For each resource, the training algorithm finds the best limit that will classify the markers as positive and negative. Obviously, there will be classification errors, and therefore, in the learning process, features that contain a minimal error rate are selected and classify the images of the markers more accurately. The ratio between the number of incorrectly reported alarms and the total number of stages defined in the tree construct provides the maximum value of false alarms [29].

TABLE 2. OpenCV Traincascade parameters.

Parameter	Value	Function
data	haar	Classifier storage location.
vec	pos.vec	vec-file with positive samples
bg	negative.txt	Background description file.
stageType	BOOST	Defines the type of decision tree
numStages	20	The number of stages of the cascade to be trained.
minHitRate	0.999	The minimal desired hit rate for each stage.
maxFalseAlarmRate	0.2	Maximal desired false alarm rate for each.
numPos	10	The number of positive samples used in training.
numNeg	50	The number of negative samples used in training.
w	30	Width of training samples.
h	30	Height of training samples.
mode	ALL	Use ALL features, including 45-degree rotations.
precalcValBufSize	2048	Size of buffer for precalculated feature values (in Mb).
precalcIdxBufSize	2048	Size of buffer for precalculated feature indices (in Mb).

The average time spent by the Traincascade algorithm to build the classifier was 0.9 minutes.

Each visual marker receives a location identification, manually indicated by the user responsible for building the markers and the map.

4) MARKER CONSTRUCTION BASED ON HYBRID INFORMATION

Hybrid markers are solutions that combine the resulting data from the markers created for each sensor with the purpose of one model compensating for the limitations of others, as indicated by Hanchuan [9].

The algorithm used to build the hybrid marker is the Linear Kalman Filter, which is used to define the processed data and a set that allows classification as prediction and update functions.

The steps for constructing the hybrid marker are shown in the flowchart of Figure 11.

The selection of the markers represents the application of a Euclidean distance calculation (straight line) between the data collected by the sensors and the markers on the map.

The linear Kalman filter was constructed to meet an intermediate profile between the Kalman filter, developed for linear problems, and the Extended Kalman filter, used in nonlinear problems [6], [31].

The linear Kalman filter assumes that the probability of the next state and the possibilities of measurement can be found in the propagations of the functions $f()$ and $h()$, where $f()$ is a nonlinear function of the process and $h()$ a non-linear function of the measuring system. In this case, the functions $f()$ and $h()$ can be used to propagate the state vector X_{k+1} and the output vector Y_{K+1}.

A nonlinear $f()$ function has a Gaussian also nonlinear, to avoid the distortion present in the linearization over the next state. The same principle applies to the $h()$ measurement function. By projecting Gaussian through this linear approximation, the later belief becomes linear.

Because it is an approximation, u_k and w_k represent the control and the process noise, respectively, while u_{k+1} is associated with the measurement noise.

The LKF algorithm uses a method called Taylor Expansion (first-order) [32]. The Taylor Expansion constructs a linear approximation of $f()$, from its value and slope. Equation (9) gives this slope:

$$f'(x_k, u_{k+1}) - \partial f\left(\frac{x_k, u_{k+1}}{\partial x_k}\right) \tag{9}$$

The linearization of $f()$ and $h()$ about the state and the process noise are called Jacobian matrices and can be seen in Equations 10, 11 and 12:

$$F - D_f(x) - \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{10}$$

$$H - D_h(x) - \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \tag{11}$$

$$W - D_f(w) - \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \cdots & \frac{\partial f_1}{\partial w_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \cdots & \frac{\partial f_n}{\partial w_n} \end{bmatrix} \tag{12}$$

In the EKF algorithm, the Jacobian matrices F and H are not constants, but they were evaluated about a specific value of the state vector, $x - x_0$, the corresponding matrices become constant. F is the Jacobian matrix of the user's movement, H is the Jacobian matrix measurement, and W is the Jacobian matrix of the user's movement noise. Equations 13 and 14 define the prediction step, and the correction and adjustment step of the data are defined by Equations 15, 16, and 17.

Prediction

$$\hat{x}^- - f(\hat{x}^+_{t-1}, w_{t-1}) \tag{13}$$

$$AP_{t-1}A^T + WQ \tag{14}$$

Correction

$$K_t - P_t^- H^T [HP_t^- H^+ + R_t]^{-1} \tag{15}$$

$$\hat{X}_t^- - \hat{X}_t^- + X_t[Y_t - H\left(\hat{X}_t^-\right)] \tag{16}$$

$$P_t^+ - P_t^- - K_t H P_t^- \tag{17}$$

where x indicates the user pose, P is the user pose covariance. Q means the Noise of the permanent movement, K is the LKF gain, and R Noise of continuous measurement. In the correction step, the values read by the sensors are indicated in the formulas by the variable Y, which are confronted with the user's pose by non-linear measurement functions $h()$. As the process is iterative, the time factor is represented in two moments as T - previous status, $T+1$ - current state, as well as the variables that are manipulated with the values of Prior information, indicated by the symbol $-$ and Posteriori, represented by the symbol $+$.

5) SCENARIO MAP

The environment received the mapping in two grids with different dimensions to allow a better location. The first grid, defined by Wi-Fi, has a distance of 1.00 m from one cell to another, and the second grid divides each Wi-Fi cell into nine addresses, where each sub-cell is 0.33 m. The initial size definition of 1.00 m observed the RSSI/SNR ratio, as described by Li and his co-authors [14].

In the mapping process, each marker was considered a node in a bidirectionally-connected graph, registered in an adjacency matrix due to its low degree of complexity [31].

The most commonly used models for constructing maps are fingerprinting and Weighted Policy Learner (WPL). The advantage of WPL for fingerprinting is that it does not need to reconstruct the map with the addition or removal of any record [21], [33]. The disadvantage of using WPL for this project is its high cost of processing, especially when the processing center is a mobile device with physical limitations of resources. To overcome the disadvantages of fingerprinting and WPL, we propose the development of the Linear Weighted Policy Learner (LWPL) algorithm. LWPL uses the WPL algorithm as the basis for map construction but inserting the concept of process linearization. LWPL is an algorithm that takes advantage of the Weighted Policy Learner (WPL) learning process, linearizing the mathematical formulations so that the construction of the data representation is faster [21].

The general idea of LWPL development is to initiate the pattern learning process faster than standard WPL using a BubbleSort sorting algorithm to sort the data according to the weight of each observed value.

The LWPL calculates the position of the target (wearable device) by the weight of the distance, from the target itself to all N markers identified in the range. The calculated distances are arranged in a vector $\left\{d_t^1, d_t^2, \ldots, d_t^N\right\}$ in time t. Then the weight of each distance is calculated as:

$$w_t^i = \frac{\frac{1}{d_t^i}}{\sum_{i=1}^{N} \frac{1}{d_t^i}} \tag{18}$$

The LWPL is an autonomous and sequential learning algorithm, which can update the observed data, adjusting two L_O and α parameters, as described in Equation 19.

$$S_t^i = PL_O + 10\alpha \log\left(d_t^i\right) \tag{19}$$

where S_t^i is the marker generated by the combination of data at time t, PL_O is the reference coefficient of path loss, α is the loss of trajectory and d_t^i is the distance between the marker i and the target at time t. Based on Equation (20), d_t^i can be expressed as:

$$d_t^i = 10^{\frac{S_t^i}{10\alpha}} \tag{20}$$

C. POSITION

The construction of markers and their neighborhood relationships on a map allow the location system to have reference points about how the user will be located and guided in the scenario. However, this information needs to be handled by algorithms that identify the user's position when he is not exactly at one point recorded on the map. In this work, we treat the location of the user while traveling in two ways: static and dynamic. In static localization, it is considered that the user is stationary and, in dynamic localization, the user is shifting through the scenario, requiring higher speed in data manipulation, because the reference points change rapidly. We detail each localization strategy below.

Static positioning is used to indicate the location of the user when the algorithm is stopped. It is also used when the construction of a route is required for navigation (dynamic positioning). Using the initial information with these characteristics is a necessary condition so that the initial data is quite robust, avoiding the contamination of the positioning result.

Dynamic positioning is used in navigation, indicating the position of the user within a tolerable relationship between the level of accuracy and delivery time of the information.

1) STATIC INDOOR POSITIONING

The Wi-Fi identifiers allow calculating the RSSI of a specific position observed in a straight line from the AP, maintaining the values within the predetermined limits, using the SNR to select the RSSI that had the least impact of noise.

Inertial localization uses an extensive collection of inertial information, obtained by an arrangement of three sensors arranged side by side on the wearable mobile device. This strategy was inspired by the formulation indicated by Civera, who adopted a grouping of six sensors to improve the quality of the position indicator quickly [25].

The visual localization system requires that some steps be performed, such as acquisition, preprocessing, segmentation, resource extraction, and recognition. Then, the interpretation of the visual information is associated with a coordinate. The identification of the information contained in one image occurs through the comparison of their features with a previously trained model. The complete visual recognition scheme is given in Figure 12.

The acquisition stage represents the submission of images to initial histogram treatments and radiometric calibration [24]. These operations normalize the contrasts and brightnesses, making the model adapted to the changing lighting and angle conditions of the venue [28].

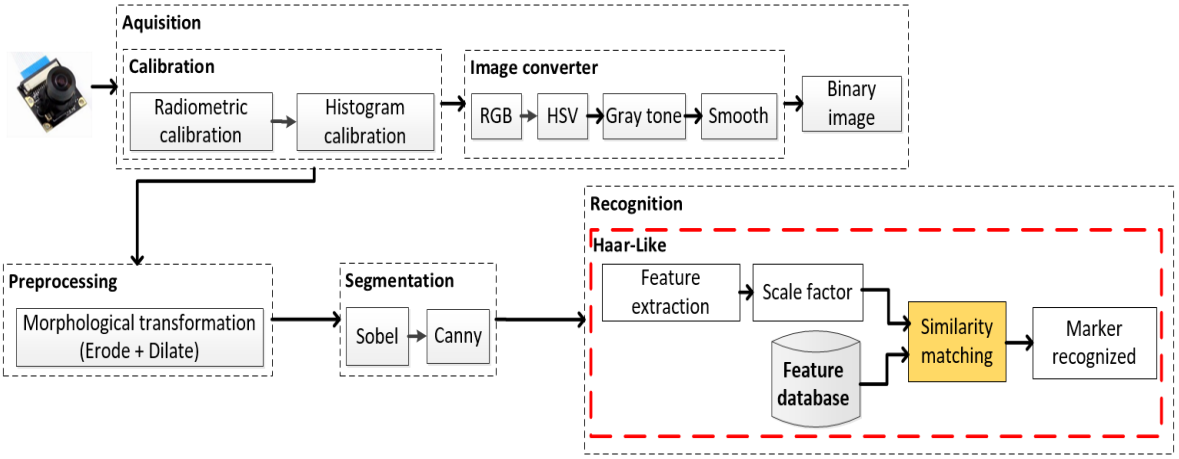


FIGURE 12. Visual marker recognition scheme.

Additionally, the images are scaled (20% increase or decrease) to improve the recognition of the objects due to the variation of capture distance from the camera to the marker [30].

In the image conversion phase, red, green, blue (RGB) images are converted to HSV color format, which is more robust in separating visual information [34]. The reduction in processing costs was achieved by converting hue, saturation, and value (HSV) format color images to grayscale images and then to full image [35]. An integral image uses a sum of the four corner pixels of an observed area to indicate its value, allowing the calculation always to be linear, regardless of the size of the observed area [36].

The preprocessing step represents the application of morphological transformation and smoothing algorithms to reduce the amount of noise and prepare the data for visual recognition [28]. Noises are caused by various reasons such as the resolution, lighting, and distance between the object to the camera. The morphological transformation combines the erosion and dilation algorithms, which follow a minimum neighborhood requirement [17].

In the segmentation stage, the combination of Canny and Sobel filters is applied, reinforcing the characteristics of frames and visual objects about the other elements of the images. The image resulting from the segmentation phase is delivered in the recognition extraction step [28].

In the segmentation stage, the combination of Canny and Sobel filters is applied, reinforcing the characteristics of frames and visual objects about the other elements of the images. The image resulting from the segmentation phase is delivered in the recognition extraction step [28].

The Sobel filter verifies the internal continuities of objects by the gradient intensity of all points in the image [37]. Mathematically, the Sobel filter operator represents the approximation calculation of the horizontal and vertical derivatives of the original image for two reference matrices with a 3×3 dimension [38].

Canny filter verifies the continuities of external edges of objects [37]. The Canny filter searches for the minimum and maximum value within the sample space to find the smallest possible distance that connects two points through the approximation obtained from the first Gaussian derivative.

Kalman's linear filter algorithm combines inertial and visual Wi-Fi marker data into a complementary fusion to provide hybrid localization using the same complexity adopted during mapping. The primary purpose of the hybrid locator is to correct the localization errors indicated by only one of its subsystems.

The hybrid location scheme combines a macro location provided by the Wi-Fi sensor that informs the coverage region to a micro-location provided by inertial sensors and cameras. Figure 13 shows the destination location scheme.

2) DYNAMIC INDOOR POSITIONING

We have built a set of linear and nonlinear algorithms to reduce the data in samples and mainly to maintain an acceptable relationship between precision level and processing time, which in practice represent less processor and memory usage (input and output operations) [16]. These algorithms are triggered from data receipt to results delivery, in addition to executing decision making, which maintains or discards one of the partial locators according to the error control indicators [33].

Linear and linearized algorithms were built to track and monitor the target during its displacement. This process is called guided navigation and is intended to correct any errors by continually updating your position compared to reference records saved on the map.

Wi-Fi-based information provides information regarding the distance from the start to the end of a route, while other sensors and virtual markers provide more subtle information and can represent curves and orientation changes during the route navigation.

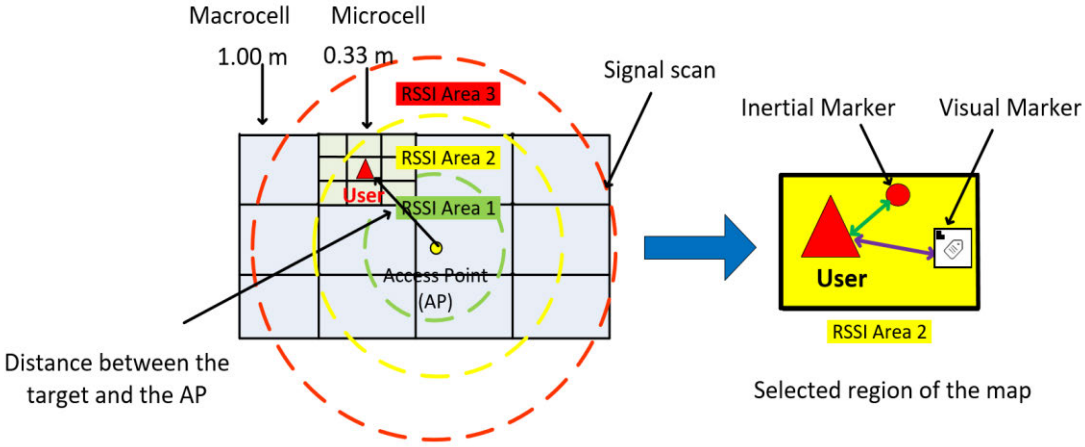


FIGURE 13. Target tracking scheme.

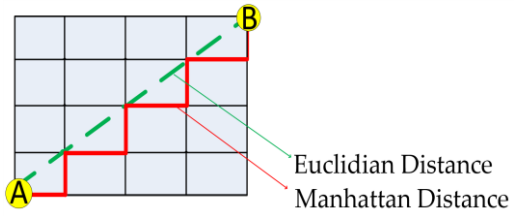


FIGURE 14. Scheme of identifying distances in a straight line or bypassing obstacles.

The distance between the two points in a straight line is obtained by applying Euclidean Distance (Equation 21). The Manhattan Distance (Equation 22) provides the distance between the start and end points considering the required curves and deviations, as shown in Figure 14.

$$Euclidean\ Distance = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)} \quad (21)$$

The sum of the legs obtains the geometric representation of Manhattan:

$$Manhattan\ Distance = |x_1 - x_2| + |y_1 - y_2| \quad (22)$$

Inertial sensors are susceptible and fluctuate a lot in the same position, so it is necessary to collect a high volume of data to have more confidence in the indicated location. However, this high volume of data requires a longer time for fusion algorithms to process and provide information.

The strategy used to decrease data collection time while maintaining the quantitative that allows the inertial location algorithm to provide reliable information was to associate three inertial sensors to work in parallel. Processing time has also been reduced by simplifying operation by disregarding the z-axis of the inertial sensor. This simplification considered that the target has a constant height above the floor during its displacement process, being necessary to manipulate only the x and y axes to cover the user's movement variations, using bilinear interpolation, giving the ability to perceive a 2D position [13]. The x-y magnetic map and angulation are indicated by

$$M_{xy}(r) = \sqrt{(M_x(r))^2 + (M_y(r))^2} \quad (23)$$

$$M_{\varphi}(r) = \tan^{-1}(M_y(r)/M_x(r)) \quad (24)$$

with the removal of the z-axis of the magnetometers, the angular perception capacity was lost. The 2D information generated by the magnetometers is enriched with data obtained from the gyroscope, which provides the angular perception capability, as shown by the formula:

$$q = {}^{S}_{G}q = [\hat{k} \sin\frac{\theta}{2} \cos\frac{\theta}{2}]^T \quad (25)$$

where {S} e {G} denotes the gyroscope and the global frames of reference, $\hat{k}$ is the axis of rotation, and θ means its magnitude.

The inertial marker also stores information regarding the length of the displacement (SL) for a given direction θ, at time t^*, as indicated by Equation (26):

$$\begin{cases} x_{t+1} = x_t + SL_t . \sin(\theta_{t^*}) \\ y_{t+1} = y_t + SL_t . \cos(\theta_{t^*}) \\ Z_{t+1} = z_t + SL_t . \tan(\theta_{t^*}) \end{cases} \quad (26)$$

The Haar-Like algorithm is relatively fast to recognize the features of the mapped objects, but it shows slow in the trace operation, which is necessary for the navigation, being a risk for the execution of the location. After the visual recognition of the marker, Haar-Like delivers the region of interest (ROI), which contains the recognized object to the CamShift algorithm associated with a Linear Kalman filter. CamShift is faster than Haar-Like because it ignores the region's growth, outline, smoothing, and forecasting considerations [36]. The Kalman filter increased CamShift's robustness in the process of recognizing objects received from Haar-Like, even when they suffer variations caused by elements of the scene, such as lighting, or suffer distortions, common in the process of displacement. If the received object is lost from the CamShift tracing, the Haar-Like algorithm is called again to perform the recognition. Figure 15 shows the relationship between

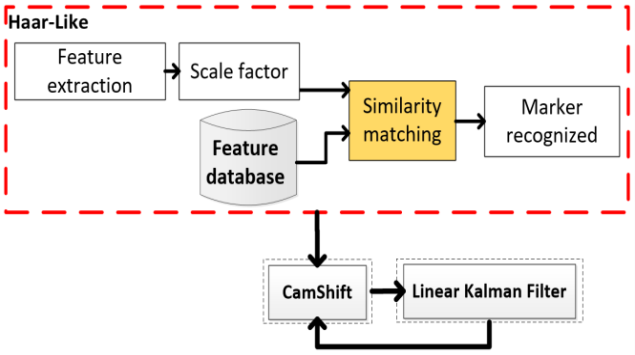


FIGURE 15. Computer-based indoor location.

the Haar-Like and the CamShift associated with the linear Kalman filter.

The hybrid location needs to be robust, fast, and continuous, using combinations that allow its use even in places that affect its subsystems. The construction of the hybrid identifier for dynamic positioning applies a simple decision that indicates the strength of the representativeness of each component to maintain or discard from the hybrid arrangement. In the calculation, only values that remain within limits indicated by the decision layer are considered, dropping from the calculation the most discrepant values, and that can contaminate the final result, thus compensating the limitations of some of the models, as already described by Hanchuan [9].

$$M_p = \frac{p_1.x_1 + p_2.x_2 + \cdots + p_n.x_n}{p_1 + p_2 + \cdots + p_n} \tag{27}$$

where M_p is the weighted arithmetic mean, $p_1, p_2, \ldots, p_n$ are the weights of the Wi-Fi, inertial and computational vision markers, and $x_1, x_2, \ldots, x_n$ are the values of the data obtained by the locations of the indoor location systems.

D. NAVIGATION

The system navigation step represents the combination of static and dynamic locations, associated with a set of rules and procedures for establishing and consuming routes. A rule is a statement about what to do in a specific situation. A procedure is a series of steps to follow to achieve a result.

In the model defined for indoor navigation, processing of subsystems based on Wi-Fi, inertial, and computational vision sensors occurs in parallel, and their results provide the basis for the formation of hybrid localization.

All data consumption and processing are performed directly on a wearable mobile device so that the information is always available to the user free of dependencies on external structures.

We detail below the entire route handling process, with its build, update and rebuild rules, how to deliver a guide to the user, and user protection against collisions with obstacles present on these routes.

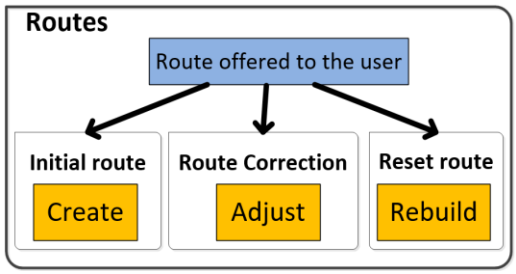


FIGURE 16. Route manipulation algorithm operations.

1) ROUTES RULES

To define the routes, it is necessary to indicate the initial position, the final position, and the routing algorithms, that verify the intermediate connections between the points based on the criterion of the processing cost [39]. The initial identification takes a longer time than the others because it is necessary to give robustness to the beginning of the process, avoiding that a weak initial identification contaminates the subsequent steps with more errors [1].

The process of route definition can be divided into three stages: creation, updating, and reconstruction, as shown in Figure 16.

The constructed route indicates the same markers belonging to the path of interest of the user and the relationships between them, which are recorded in an adjacency matrix, with the following scheme:

Dijkstra algorithm defines the shortest path to be used in navigation, consuming the data registered in the adjacency matrix [40]. The Dijkstra algorithm followed the four steps:

Step 1: Assign the infinite value to indicate the distance to for all the vertex pairs, except for the origin vertex;

Step 2: The current vertex is defined as the start, and all others receive the status of unvisited;

Step 3: From the current vertex, the distances are calculated to reach all the unvisited neighbor vertices, keeping the lowest value as valid and discarding the others;

Step 4: The nearest vertex is defined as the new starting point, and the algorithm performs step 3 again.

The Dijkstra algorithm uses the following scheme to relate the selected records of its search in the mapped markers to form the route:

$$\text{Ej k} \begin{cases} = 1, & \text{if the vertices are connected by edge} \\ = 2, & \text{if the vertex it is a crossing} \\ = 0, & \text{else} \end{cases} \tag{28}$$

The established route adjustment operation observes values referring to position, angle, and direction estimate. The position correction codes are passed to a set of algorithms belonging to the motion recognition module.

The reconstruction of the route uses as the parameter the maximum distance defined for the margin of error, which is equivalent to the worst location indicative of Wi-Fi. When this value is reached, the navigation guide process is interrupted, and the construction of the route is redone

using the current user's address as the starting point for new navigation.

2) GUIDE

The navigation guide has the function of relating the perceptions of the user's movement, the data obtained from the sensors or virtual markers, indicating the route and the presence of obstacles, and the delivery of the sound instructions to the user.

The guide module consists of the i-PDR navigation algorithm and the integration system.

The i-PDR navigation algorithm associated with the Kalman filter calculates the position to be reached by the user in their displacement and attenuates the effects of the route deviations by applying the Kalman Filter.

The algorithm, called iterative PDR (i-PDR), modifies the PDR by inserting an iterative update function. This intervention in the PDR algorithm addressed its two major shortcomings: velocity oscillations and location identification variations, as described by Zou and his co-authors [21]. These PDR deficiencies generate errors that accumulate along a path and create position divergences when compared to the map reference register [33]. The i-PDR represents the joining of the PDR to the Kalman filter, which provides iterative and tuning characteristics of the orientation algorithm.

The i-PDR starts its localization and adjustment of the user's positions after the initial location, which is performed by the LWPL algorithm, comparing the data with the mapped records of the scenario (static location). The initial location, provided by LWPL, is calculated by the formula:

$$(x, y) = \sum_{i=1}^{N} w_t^i (x_i, y_i) \tag{29}$$

where w is the weight given to the record, (x_i, y_i) is the destination location of the navigation. For each marker, the process is repeated until all routes are known.

The i-PDR uses the multisensory fusion of the preprocessing step and dynamic positioning to identify the user's positions during their travel, taking as standard the speed of about 1.1 m / s [11].

The i-PDR has a solution that relates the magnitude of the acceleration to the displacement size, reducing the variations of error margins. The expression describing this relationship is as follows:

$$L = \beta (a_{max} - a_{min})^{\frac{1}{4}} \tag{30}$$

where β is the adjusted coefficient when the acceleration variations are perceived, indicating the size of the displacement between one marker and another.

The candidate sites are indicated by the application of the K-Nearest Neighbors Algorithm (KNN), which calculates the approximation of the record built for navigation in the route map maps. In the KNN, the parameter k indicates the number of neighbors reads to estimate the coordinate, and the parameter q indicates the scheme adopted to calculate the distance between the center value (weighted average) and the neighbors selected by parameter k.

Due to a large amount of data presented as candidates for location indication, only k RPs are chosen according to the first k minimum distances identified by Equation (31).

$$\left(\hat{x}, \hat{y}\right) = \frac{1}{k} \sum_{t=1}^{k} (x_t, y_t) \tag{31}$$

where (x_t, y_t) is the RP coordinate at time t, $\left(\hat{x}, \hat{y}\right)$ is the estimated coordinate of the observed point. Parameter q can receive two values: 1 - indicates the use of the Euclidean equation; 2 - indicates the use of the Manhattan distance equation. The hybrid marker construction assumes that the value of parameter q is always equal to 1.

If the number of reference points (RPs) and the number of location markers (LMs) are respectively m and n, the distances are defined as follows:

$$\begin{cases} L_{qi} = (\sum_{t=1}^{n} |s_t - s_{it}|^q)^{1/q} \\ i = 1, 2, \ldots, m \end{cases} \tag{32}$$

where s_t is the sample collected to identify the location at time t where the data are compared, s_{it} is the sample collected from the data received from the Wi-Fi, inertial, and visual models.

The Dijkstra algorithm selects the adjacent locations using as a criterion the shortest distance between the location of the device position and the candidate points.

The selection of the adjacent places occurs by the application of distance calculations between the coordinates received by the sensors and the candidate coordinates of the route map.

The direction of the user's movement is obtained by the difference between the z-axis (azimuth) angle of the magnetometer and the geographic north [41]. This information is reinforced by the gyroscope, which indicates the angulation and correction of the user's posture for perception [41].

This movement status information is entered into the integration system, which powers the i-PDR algorithm and Kalman Linear filter itself to decrease error margins and provide the necessary adjustments to the next position [6], [42].

The Kalman filter adds its data to the preprocess data set to reduce the impacts generated by systematic and non-systematic errors. A systematic error occurs within a standard and is more straightforward to deal with, such as the refinements applied to variations in the user's distance and walking speed [41]. A non-systematic error occurs outside a standard, which hampers the standardization of the approach, such as displacement variations on uneven floors [41].

The recognition of curves and straight lines results from the fusion of magnetometer data and gyroscopic sensors using an angular acceleration calculation and a Gaussian filter [41]. This association identifies the direction of the curves, as shown in Figure 17.

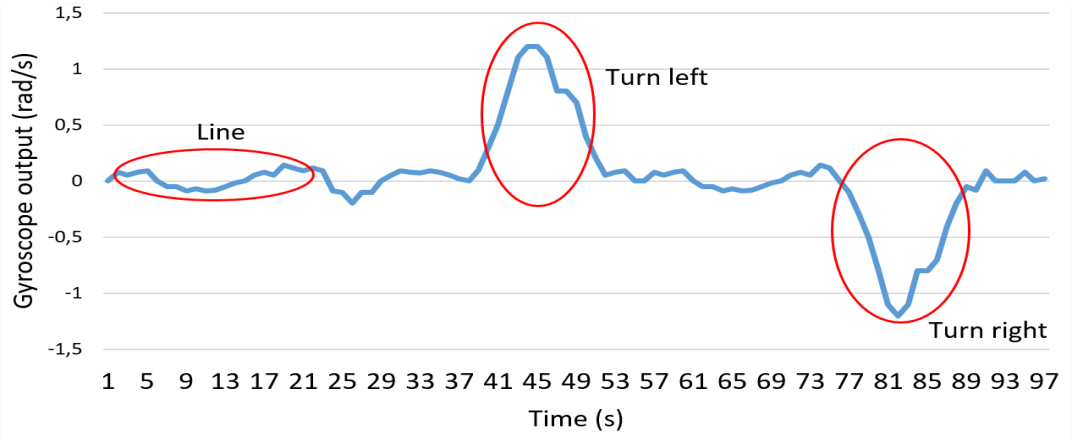


FIGURE 17. Identification of curves and lines.

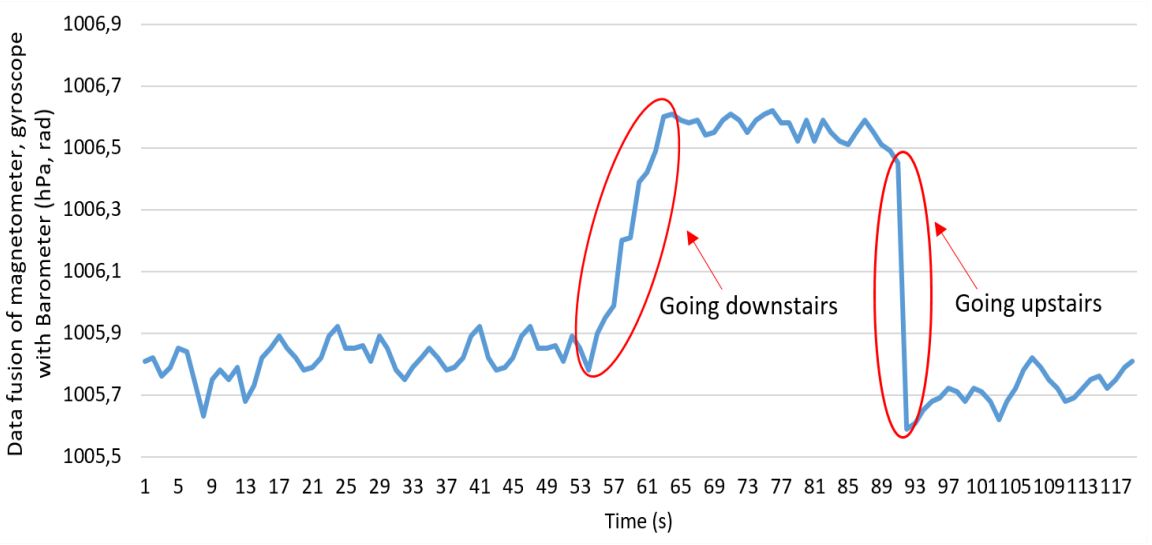


FIGURE 18. Association of magnetometer, gyroscope, and barometer to indicate ascents and descents.

The movements identified as upward and downward are obtained by data fusion of magnetometer and gyro sensors, and confirmed by the barometer so that it was not confused with the lateral movement (right, left) [4], [5], [24]. The barometer indicates the upward movement when the air pressure decreases and the lowering movement when the air pressure rises, as shown in Figure 18 [16].

3) OBSTACLE IDENTIFICATION

The Obstacle Perception System uses visual information captured from both cameras in a stereo-vision scheme to calculate the distance to the objects arranged in front of the user. The 3D data is decomposed into two 2D arrays in which the x and y-axes are on the ground, and the *y-axes* and *z-axes* represent the points belonging to the vertical matrix. Figure 19 gives the general scheme of horizontal and vertical perception of obstacles.

The integration module combines the processed data of the guide with the obstacle perception data, delivering the corresponding sound information to the user. The calculation of the safety distance module has algorithms that are responsible for the identification of obstacles positioned horizontally and vertically in the scenario, providing spatial reasoning to the user.

The cameras are affixed at a similar distance from human eyes. Given a pixel that describes some feature in the image to the right, that same feature is searched in the left image, calculating the distance between the reference points, as shown in Figure 20.

The distance between the two points of view, called disparity, provides the depth information and hence, the common reference points in the two images [18]. Objects close to the observer have a more significant disparity, whereas distant objects show less difference [43].

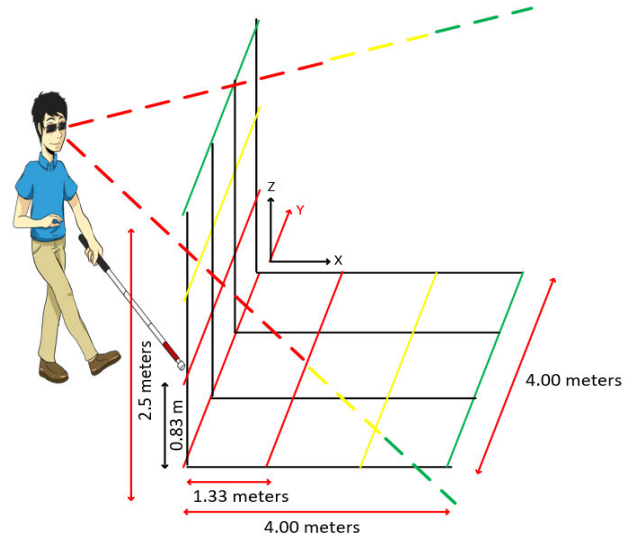


FIGURE 19. Obstacle detection scheme using stereo vision.

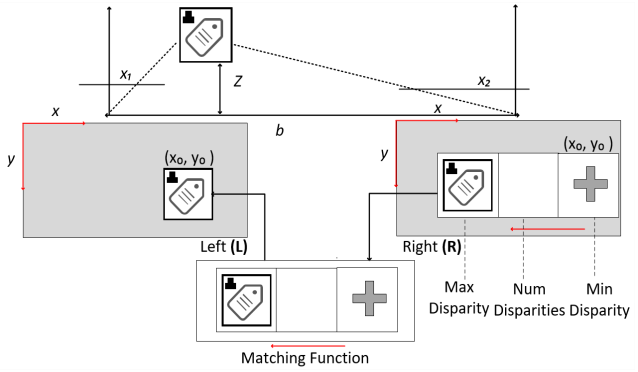


FIGURE 20. Systematics of the disparity map operation.

The matching function between the left and right images separates the image from the right into sample blocks. Then, a search for this block in the left image is applied, based on the parameters reported, starting with minDisparity and going left in numDisparity value or finding the corresponding block. The disparity is the result of the difference between the two blocks.

From the triangular relationship, it is possible to calculate Z as follows

$$Zxs = f\frac{b}{d} \tag{33}$$

where $d = x_1 - x_2$ is called the disparity between the left and right images, and the s is pixel size. Equation 37 also allows calculating the distance between the user and the obstacle in front of him.

The FindStereoCorrespondenceBM function of the OpenCV library calculates the disparity map, which uses block and sum (Absolute Difference Sum) matching methods to find matching points between two images. The algorithm uses the preFilterSize function to normalize the brightness and texture of images before the matching process in both images.

The SAD algorithm applies the SADWindowSize function to match the horizontal lines, eliminating bad combinations using a predefined number of pixels in the numberOfDisparities parameter. The preFilterCap function determines the central pixel of the images and serves as a guide to the combination of the two images [24].

The FindStereoCorrespondenceBM algorithm is based on block matching, which associates matching blocks and edges, which are information that generates high disparity [44].

The speckeWindowSize and speckleRange functions parameterize the size of the search window and the maximum acceptable variation to decrease the disparity [44].

The alerts for ground obstacles occur by the evaluation of the distance between the user and the nearest obstacle (up to 4.00 m). The activation of the vertically arranged barrier warnings occurs by evaluating the result of the distance between the user and the obstacle, up to the limit of 2.50 m.

The sound alert activation function uses the distance between the user and the nearest obstacle to trigger the corresponding sound, and the speed of its movement interferes with the frequency of sound emission. Thus, the user realizes that he is approaching an obstacle, the faster the sound emission.

The relationship of the sound schemes and the distances perceived by the stereo vision system uses an alert sound language similar to that used in the vehicular rear warning system, enriched with information concerning the height of the obstacle, as indicated in the flow chart of Figure 21.

The audible alerts are beeps on the A + and C + musical notes, in the bass and treble, with a difference of three octaves to reinforce the difference between the tones. Musical tones have been chosen as the most pleasing natural notes for human ears [45].

E. AUXILIARY DATA

The intended audience for the use of the system is visually impaired, requiring all interaction to occur as simply and

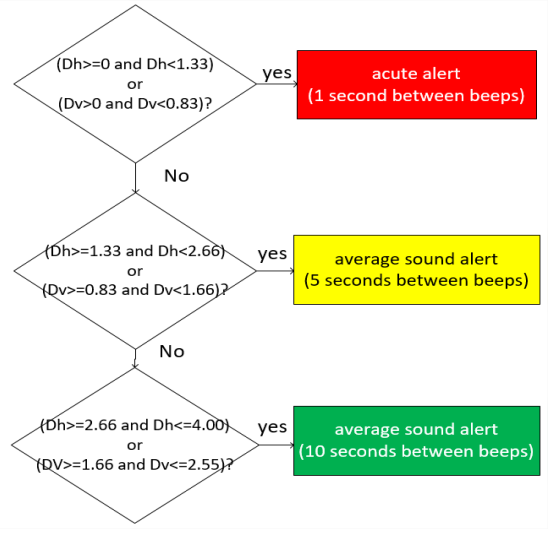


FIGURE 21. Sound alert scheme based on obstacle distance.

naturally as possible. The way adopted in this study was to use commands spoken by users to enter the commands, and all system feedback occurs through sound instructions.

The system uses two modules with two well-defined functions: learning spoken instructions and storing models and managing an audio bank. The speech learning system applies algorithms that generalize the speaker (independence of the speaker regarding gender and intonation), creating models that are used as a reference in the speech recognition process. The management of the audio bank is the process of manipulating a set of previously defined spoken instructions that correspond to navigational guidance and warnings of collisions with obstacles. We detail each role and its practical approaches in the topics below.

1) SPEECH CLASSIFIER

The voice, in its natural state, has stereo features that require high processing and time [46], [47]. Thus, the vowel sound was reduced to a monophonic label to allow the recognition of the pattern more quickly and to share the same decision tree, dividing the nodes according to binary properties of the phonetic context [6].

The voice is an analogical feature that must be subjected to a scanning process to be used in the indoor positioning system. Thus, Linear Maximum Likelihood Regression (MLLR) defines a sampling, which is quantized (converted to the numerical form) [46]. The numerical data are submitted to the training and extraction of characteristics, applying Gaussian averages and weight variations at each iteration, as shown in the diagram of Figure 22.

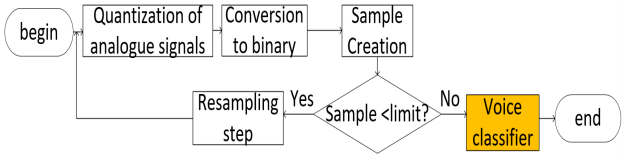


FIGURE 22. Scheme of the construction of the voice classifier.

A reduced set of five instructions allowed user interaction with the system and reduced learning time. Table 3 shows the voice input commands used in user interaction with the system.

2) AUDIO BANK

Sound information is the mode used to inform the user about the operation of the device, its location, and navigation. To facilitate the interaction, a set of sound instructions was established, containing eleven short phrases, five to respond to user vowel requests and six used to guide the user, turn the device on and off.

Table 4 shows the relationship of navigation guidance commands and system output sound instructions to the user.

The system architecture, considering the details of each subsystem and the linkages between them, are presented in Figure 23.

TABLE 3. Relationship between voice commands and system sound responses.

Action	Voice/sound command	Answer of the audio guide
Enable system	Enable	System enabled
Set destination address	PLACE_NAME	Address received
Auto Location	Where am I?	You are in PLACE_NAME
Pause navigation	Disable	System off
Confirm start navigation	Yes/ No	Yes: You are T minutes from the destination No: Navigation stopped

TABLE 4. Set of instructions for indoor navigation.

Action	Answer of the audio guide
Drive forward	Go ahead
Turn right	Turn right on X meters
Turn left	Turn left on X meters
Turn right immediately	Turn right
Turn left immediately	Turn left
Alert: Close obstacle	Stop! Obstacle detected

The implementation maintained the hybrid features proposed in the model of Figure 1. Its two main modules describe the tasks required to construct the mapping and allow navigation. In the mapping module, the algorithms used to create the markers based on the Wi-Fi, inertial, and visual technologies, and the algorithms used to relate the markers to a map are indicated. The navigation module contains the algorithms that define the route to be used and all the processes necessary to follow and adjust the routes when necessary are presented.

IV. EVALUATION

A protocol was defined to standardize the tests and evaluations of the algorithms. Two scenarios were used for the tests: one rectilinear and free of obstacles and one with curves and obstacles. In each place, a set of collection points was defined to observe the results of the distances of the Wi-Fi, Inertial, Visual, and Hybrid models for the mapped registers.

The procedures used for the rectilinear scenario are:

- Data collection begins at the door of the Intelligent Environmental Laboratory (AmILab);
- The user must enter the default destination address of the tests (auditorium door), distant about 75.00 m.

The procedures used for the scenario with curves and obstacles are:

- Data collection begins at the door of the Ambient Intelligent Laboratory (AmILab);
- The user must enter the default destination address of the tests (cabinet located on the right side of the room), and the laboratory should be circulated to arrive at the address.

A. EXPERIMENT PROTOCOL

The environment chosen to explore the behaviors and to apply the evaluation processes in the prototype and in the algorithms were the corridor (Figure 24a) and Intelligent

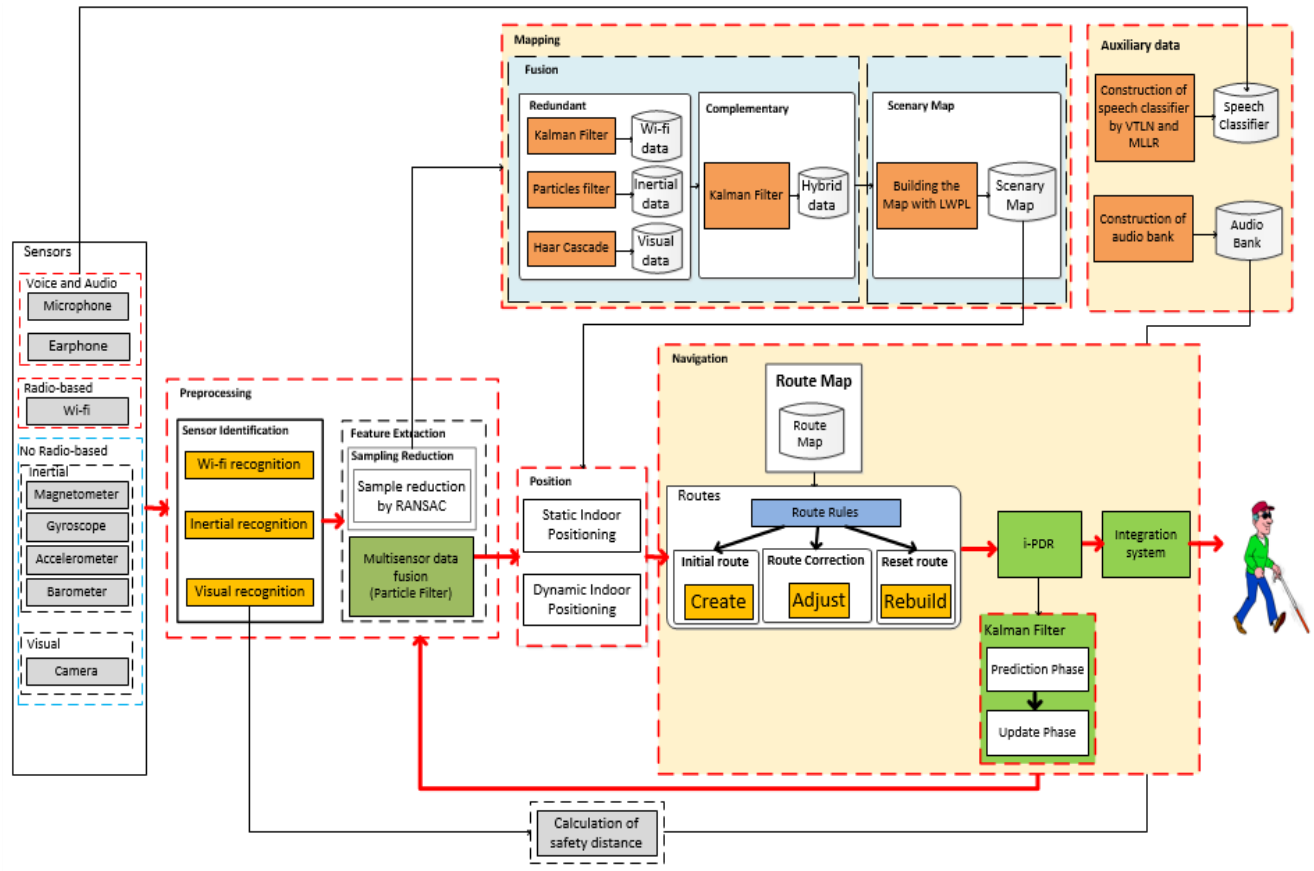
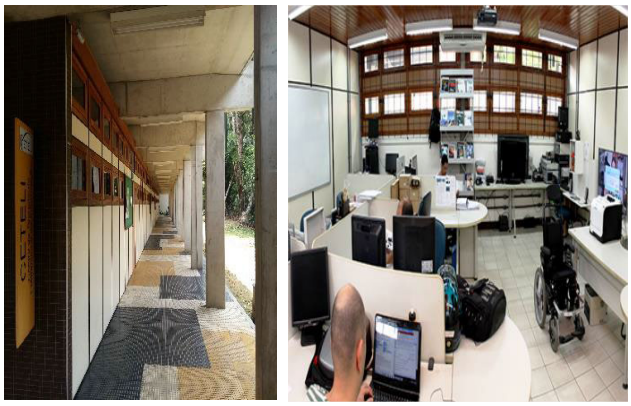


FIGURE 23. Detailed architecture of the indoor positioning system.



(a) (b)

FIGURE 24. (a) CETELI corridor and (b) Ambient intelligent laboratory (AmILab).

Environmental Laboratory (AmILab) (Figure 24b), belonging to the Center for Research and Development in Technology and Information Electronics (CETELI), Federal University of Amazonas (UFAM). The corridor has dimensions of 75.35 m × 4.00 m and was used as a rectilinear environment. The Intelligent Environmental Laboratory (AmILab), has dimensions of 10.83 m × 13.80 m and was used as curves and obstacles.

B. EXPERIMENT SETUP

Two devices were built to carry out the tests and evaluations: one fixed, with the function of emitting Wi-Fi signals, and one mobile, containing Wi-Fi, inertial sensors and RGB cameras, capable of processing the data locally.

The mobile device considers the hands-free condition, which is an essential requirement in experiments with visually impaired people [17]. Its construction started with a device affixed to the feet, however, because excessive vibration was changed to waist height. Although the level of vibration has decreased, the position of the device was not adequate to perceive obstacles arranged at head height. Finally, a pair of glasses was chosen as the device.

The construction of the two prototypes considers as principles the discretion and the lightness but guaranteeing sufficient space to accommodate their electronic resources, as shown in Figure 25.

The ESP8266 Wi-Fi sensor is used on the fixed device and the mobile device with specific functions. In the fixed device, the sensor provides a reference point for the location. In the mobile device, the sensor receives the RSSI signals and uses these values in the distance calculation. The association of the STMicroelectronics L3G4200D gyroscope sensors, the Analog Device ADXL345 accelerometer, the Honeywell HMC5883L magnetometer, and the Bosch

(a) (b)

FIGURE 25. (a) fixed device and (b) pair of glasses used for indoor mapping and navigation.

BMP180 barometer makes the inertial location. The cameras used are of the RGB type, the Raspberry platform, with five megapixels resolution and field of view between 30 and 60 degrees. All processing is done on the platform Raspberry Pi Zero, present in the pair of glasses to increase the perception of the markers.

C. RESULT OF EXPERIMENTS

The experiments followed the predefined protocols for each environment, and, in each test, the localization subsystems were observed alone and combined (hybrid) to record the performance evolutions (precision versus time) as well as to know the adjustment needs.

The system information was monitored remotely for data analysis and not to interfere with the experiments. The received data was organized in the following visual scheme: a circular marker (blue) indicates the fixed AP of the Wi-Fi present in the scene, the stars (yellow) represent the main mapped locations. The other displayed navigation addresses represent the map data and the computed positions of the established route.

The data collected from the navigation, including the start and ends of the routes, is visually represented in a location map. A solid blue line indicates the data belonging to the system map that are used as reference points. A dotted black line represents RSSI (Wi-Fi) signals. A solid green line indicates the locations recorded by the inertial localization. An orange dashed line shows the positions of the visual tracking system. A solid red line indicates the location registered by the hybrid location.

The navigation route began on the right side, near the entrance of the AmILab laboratory, and ended in front of the auditorium (left side of Figure 26 (a)). The navigation in the laboratory consisted of making a complete turn in the room, starting at the entrance door, and finishing in front of a cabinet on the other side of the laboratory (Figure 26 (b)).

The Wi-Fi location test used an access point with 228 cells in the corridor and 154 cells in the lab, all of them with a 1.00 m^2 dimension, to meet the maximum tolerable error limit.

The inertial localization test used 684 cells in the corridor and 462 cells in the laboratory, all of them 0.33 m^2 in size, to meet the maximum tolerable error limit.

The visual localization test used the same 684 cells in the corridor and 462 cells in the laboratory, which were defined for the inertial location, all with a dimension of 0.33 m^2.

The main factors that have generated impacts on the Wi-Fi signals in the two scenarios are listed in Table 5.

The application of the SNR to RSSI has diminished the effects of the obstacles on Wi-Fi signals. The relationship between SNR and signal quality is shown in Table 6.

Reading the RSSI and SNR data takes between 1 and 2 seconds, with a smaller volume than the one collected in the mapping. The increase in Wi-Fi data confidence is given by the selection of the 5 closest records of a map record.

The sensors that make up the arrangement and inertial location show deviations in the values of their axes, which cause misalignment at the x and y (horizontal) coordinates, and y and z (vertical). Table 7 lists the main causes of interference in the inertial sensors in the two study scenarios.

It is necessary to calibrate the system to record inertial marking information. For this purpose, the location

TABLE 5. Loss of Wi-Fi signals.

Obstacle	Interface level	Loss (dB)
Free space	Low	0.0
Wood	Low	5.0 to 8.0
Brick wall	Medium	15.0 to 20.0
Glass	Medium	13.0
People	Medium	5.0 to 12.0
Concrete wall	High	14.0 to 25.0
Metal	High	12.0
Wet wood	High	13.0
Wireless Devices	High	12.0

TABLE 6. SNR and signal quality.

SNR	Quality
Above 40 dB	Excellent
Between 25 dB and 40 dB	Good
Between 15 dB and 25 dB	Poor
Between 10 dB and 15 dB	Very Poor
Lower 10 dB	Extremely Poor

TABLE 7. Causes-effect ratio in inertial values.

Cause	Effect
Sensitivity in displacement	Magnetometer coordinate error
Sensitivity to temperature	Error in the vertical position indicated by Barometer
Fluorescent lamps	Magnetometer and barometer oscillated between vertical and horizontal axes
Presence of metallic materials	Magnetometer oscillated sharply in the same position

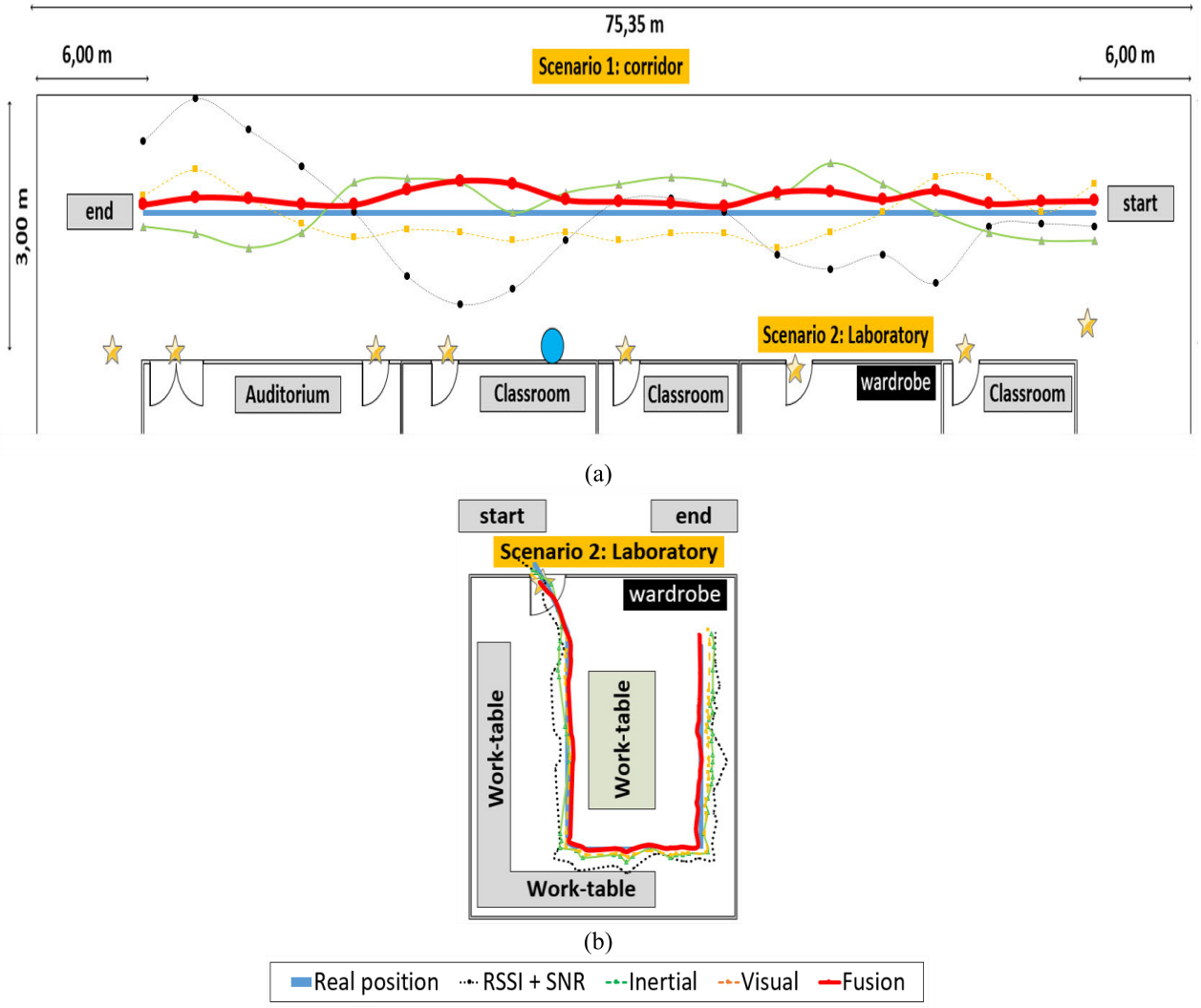


FIGURE 26. The result of target navigation in the corridor.

of the city of Manaus obtained in the public domain http://www.magnetic-declination.com/ was used. It is also possible to get the margin of error of the inclination of the magnetic axis. Figure 27 shows the inertial values of the city of Manaus.

The angle of observation of the magnetic axis follows the rule that at the poles, the angle is 90.0 degrees, and the equator is 0.0 degrees, but this declination varies year after year, and adjustments are required. For the city of Manaus, the current magnetic error is 0.1222173.

The inertial system design used three magnetometers to reduce the interference of the scenario on perceived values. However, metal structures and people interfered with the results, presenting a margin of error between 0.10 m and 0.30 m. Figure 28 shows the data received from the combinations of the three sensors, indicating that the difference (dispersion) between the axis values is minimal.

The positions showed by the visual system diverged from the expected values, caused mainly by the natural light present in scenario 1 and by the many artificial light

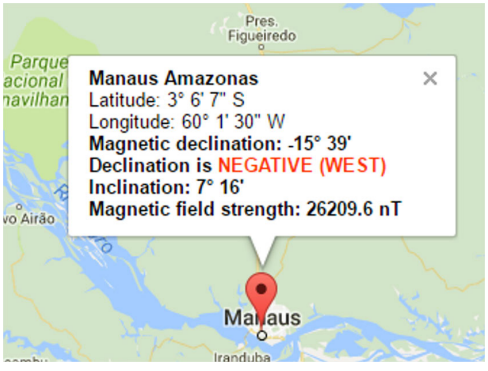


FIGURE 27. Inertial reference of the city of Manaus.

intensities in scenario 2. These variations confused the interpretation, causing the system to confuse in some moments the walls as part of the floor. The visual markers of Figure 29 show the same label observed under different levels of light intensity.

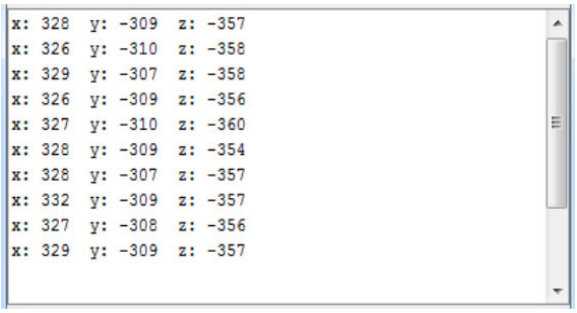


FIGURE 28. Shown values captured from the combination of magnetometer type sensors.

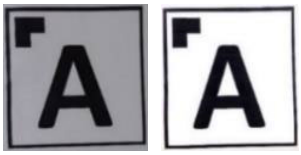


FIGURE 29. Visual markers under different lighting.

TABLE 8. Noise and its consequences on visual information.

Noise	Effect
Salt and pepper	Presence of granular noises that degrade image quality
Impulse	Margin border discontinuity and difficulty targeting the background
Gaussian	Variação da intensidade desenhada dos elementos estruturais da imagem

TABLE 9. The relation between the number of frames processed and time.

Technique	Frames Per Second (FPS)
Haar-Like	11
CamShift	22
CamShift with LKF	26

Visual information undergoes oscillations of value due to a series of factors, which generate unwanted effects, making it difficult to recognize the information. Among the main causes that produced noise on the visual system are the issues related to lighting and the presence of photoelectronic elements. Table 8 lists the consequences of noises on captured images.

A criterion used to test and analyze the visual positioning system was to count the number of frames processed every second using the frames per second (FPS) ratio. Table 9 shows the performance gain for each technique inserted in the system, indicating that higher values of FPS make the navigation system more efficient.

The hybrid model represents the combination of Wi-Fi, inertial, and visual systems, taking advantage of each component but inheriting its limitations.

To increase the level of precision, the hybrid model has a decision layer that maintains or removes the data according to the predetermined error limits. Table 10 shows the mean

TABLE 10. Mean of the margins of error presented by the Wi-Fi, inertial, visual, and hybrid subsystems.

Location Strategy	Error Margin (m)
Wi-Fi Marker	1.597
Inertial Marker	0.242
Visual Marker	0.454
Hybrid Marker	0.108

TABLE 11. Relation of the time factor and the use of subsystems.

IPS type	Time (s)
Wi-Fi Location	0.20
Location by inertial information	0.12
Location for visual information	0.17
Hybrid Location	0.07

distance of the margin of error of the Wi-Fi, inertial, visual subsystems, and the result indicated by the hybrid model.

The data fusion received from the inertial sensors provided a more robust horizontal and vertical orientation perception, with an average orientation error of around 0.186 rad, which occurred every 10.00 m.

Another criterion adopted in the evaluation of systems is the time factor and was used to indicate the evolution of time of each physical and logical intervention used in the system, as shown in Table 11.

The gain of time in the hybrid localization process was achieved by applying the perception of the standard deviation, where the subsystem that distanced itself from the standard is ignored from the second iteration. That is, the worst location indicator is used only in the first moment to aid in the understanding of the location and then is scorned so as not to contaminate the processing.

The i-PDR algorithm presented an average location error of 0.11 m, and an error in the direction of 1.8 degrees, remaining stable in static and dynamic positioning modes. The hybrid model still had a 93.23% lower margin of error than the Wi-Fi-based system, which had the worst result among subsystems.

The system presented an error rate of 11.3% for the ground level perception caused by light interferences in the computer vision system and the non-use of the z-axis in the inertial location system. However, the data combination from the inertial system allowed an error reduction to 3.60%.

The test applied to the visual perception of obstacles had as evaluation criterion the generation of alerts on the location and distance of the obstacle closest to the user. The environment used for this test was the AmILab, which had the area divided into four regions, with the presence of obstacles arranged along the way.

The objects chosen to be obstacles met the criteria of size (small and large objects) and spatial location (horizontal and vertical). The objects selected by size allow observing if the system creates the alerts so that the user knows when it is

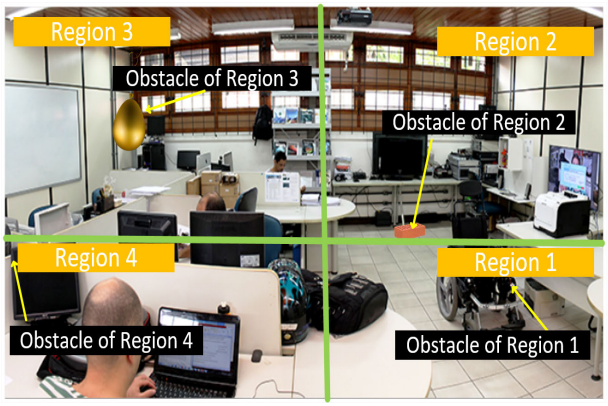


FIGURE 30. Perception of obstacles distributed in the four regions of the laboratory.

TABLE 12. Result of stereo visual detection of obstacles distributed in the laboratory.

Region	Height (m)	Distance (m)
Region 1	0.101	0.212
Region 2	0.205	0.647
Region 3	0.942	0.303
Region 4	0.942	0.129

necessary to circumvent the obstacle or when it is possible to pass between two obstacles. The objects selected due to spatial location allow observing the receipt of alerts to obstacles present at the height of the user's head.

In Region 1, a wheelchair was left on the route. In region 2, a brick with a height of 0.20 m was used as an obstacle to verify the perception of small objects. In region 3, an air balloon was hung at the height of 1.50 m to check the presence of objects arranged above the waist height of the user. In region 4, a table with a height of 0.75 m was used to verify the perception of complete objects arranged in front of the user, as shown in Figure 30.

Table 12 shows the average error of users' perceptions of obstacles in the horizontal and vertical planes.

The speech recognition algorithm has been tested in two scenarios to measure its effectiveness: one noise-free and the other affected by various noises. The noise-free environment was controlled by using the non-standard time environment so that no other people were present, and no other noisy sources would confuse receiving spoken instructions. The environment that had the presence of noises, such as doors opening and closing, people talking, air conditioners on, etc., allowed to observe the behavior of the system in a scenario closer to the real.

The models were trained by a set of human voice records, obtained from 20 people, regardless of the gender and age of these users for 2 hours of recording.

This data volume consists of 600 trainings on the three spoken instructions defined for the interaction: Enable system, Set destination address, and Auto location. We tested the trained models on a part of the data (20% of the total) to verify the independence of the speaker. The system presented 3.27% of word error when the environment was free of noise and 19.66% when experienced in an environment with noise.

TABLE 13. Result of recognition of speech commands in the controlled environment and the noisy environment.

Action	Hit rate % (no noise)	Hit rate % (noisy)
Enable system	98.00	84.00
Set destination address	92.98	76.00
Auto location	99.20	81.00

Table 13 shows the results of the three main voice commands used for system interaction in both environments.

Only by applying the Linear Maximum Likelihood Regression (MLLR) algorithm, the variations for the same spoken instruction indicated an error rate of 25%, even in a noiseless environment. This error rate reflected a delay in the user's navigation or confusion in receiving the wrong information. The application of Gaussian averages to the data in each iteration reduced this error rate to about 1% for the Auto Location command and 2% for the Enable System command. The use of the Gaussian average shows a gain in the performance of speech recognition, even with the use of larger and more complex sentences to be recognized.

We compared the system built in this study with some related works to assess the impact of the physical and logical choices used.

The evaluation process used a set of previous works that brought information related to algorithms, hardware, and databases in their publications, and the results obtained on error margins and information delivery time. This caution was necessary for the evaluation to reflect as much as possible the behaviors between similar research.

Bolat, Chen, Leppäkoski, Pradeep, and Kealy wrote the works chosen for this evaluation [13], [24], [11], [17], [16]. The authors used equivalent strategies, equipment with the same technologies, and test scenarios that brought in their physical structures the same factors to be treated. Figure 31 shows a comparative assessment of these related works, and this study in question graphically.

Bolat proposed an indoor navigation solution for pedestrians using a collaborative technology scheme to address the deficiency of the other through a particle filter [13]. Comparing with Bolat's work, the results showed greater stability due to the decision to apply a preprocessing step before the construction of the markers, thus reducing the noise present in the raw data [13].

Chen associated a visual data set of an RGBD camera and inertial sensors with a Kalman filter to generate a 3D perspective [24]. Compared to Chen's work, the model constructed and described in this paper reduced the computational costs of processing and memory consumption by dealing with 3D information in two 2D matrices. This reduction enabled faster delivery of information to the user.

Leppäkoski associated RSSI signals with inertial sensor data to record the scenario in more extensive areas

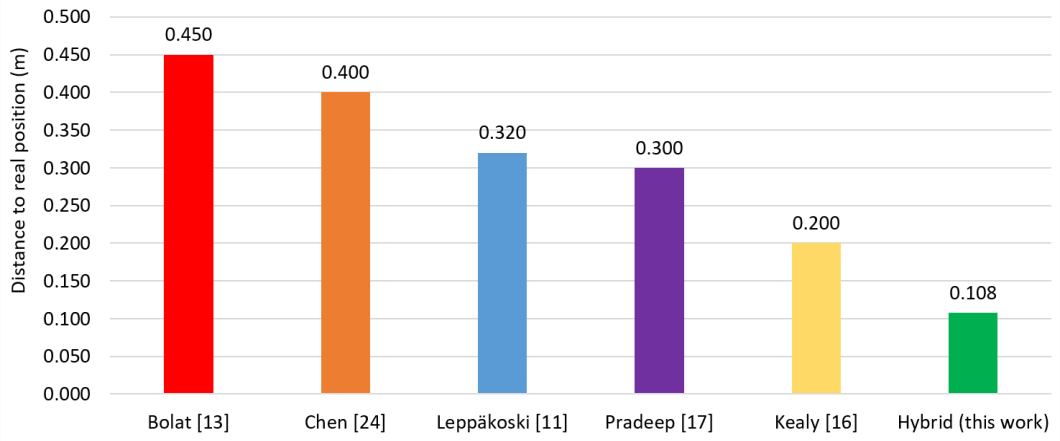


FIGURE 31. Comparison of the margin of error of the hybrid model and the related works.

(Wi-Fi signals) and subareas (inertial markings) to allow a better indication of location [11]. The data were combined with complementary and extended Kalman filters (CEKF) and a particle filter and nonlinear Bayesian algorithms. This work also used the combination of Kalman filters and Particulate Filter, however, the implementations linearized the formulations to reduce the processing cost, reducing the information waiting time. Error compensation is achieved by the iterative approach of the i-PDR algorithm, which corrects the data.

Pradeep's scheme for building its markers and maps had a performance impact caused by the nearest k-neighbor algorithm (KNN), configured to observe eight neighbors of each candidate [17]. In this scheme, the system acted as visual odometry, but the map had a static feature (once built, changing any record required the full reconstruction of the map). The approach given to the system developed and described in this paper used a linear strategy called LWPL, which allowed us to observe each candidate's neighbors more quickly and dynamically update the neighborhood degrees, without the need to reconstruct the entire structure.

Kealy built an inertial indoor positioning system to not interfere with any physical structure in the scenario [16]. The author opted for a direct positioning search, using several iterations to find a recorded position. The i-PDR model associated with the Kalman linear filter allowed us to achieve a higher level of accuracy from the first iteration, reducing the waiting time for information.

V. CONCLUSIONS AND FUTURE WORK

This work proposes the construction of a hybrid indoor positioning system capable of being used in environments with different characteristics such as open and unobstructed spaces and closed spaces and with the presence of fixed and mobile obstacles. The system combined data from three various sensors types and linear and linearized algorithms to improve the accuracy level without affecting the maximum delivery time of the results.

The development of hybrid systems brings as a significant concern the need to have mastery over the physical and logical elements used so that failure of one of the components does not contaminate the result.

Another relevant aspect is to know the device characteristics used to run the application. Although new mobile devices have shown higher processing power and memory, hybrid indoor positioning algorithms still require features, which can exceed the physical limits available. Therefore, for the execution of this work, it was proposed and built a wearable mobile device that had more processing power and more versatility of sensor application.

Indoor positioning systems should have architecture and robustness closer to the already consolidated model for outdoor environments to become a useful resource for various audiences and classes of problems [4], [9].

This study has thrown a new look at the problem described in several localization and navigation projects in an indoor environment: How to develop an IPS that has an acceptable relationship between the level of accuracy and speed to be used in a wearable device?

Briefly, the answers obtained were:

(1) The time of data collection is different for the mapping and navigation processes. For the mapping, it is tolerable that there has been a lot of time in collecting data and constructing the markers so that they are more accurate. For navigation, the time spent in the collection and identification of a candidate position is the least possible, so that the delivery of the information occurs within the tolerable limits of insurance to guide the user.

(2) The IPS accuracy still depends on the sensors types and mathematical approaches used, because, due to the different characteristics of the indoor environments, each strategy presents its limitations and may even make it unfeasible in some contexts [17].

An intermediate solution, using linearized algorithms, allows establishing tolerable levels of precision and processing time in the development of IPS for visually impaired people.

Research on IPS has been driven by the pursuit of improved accuracy levels, creating new possibilities in user interaction with its indoor environments. Thus, this work also brings its contributions.

Among the contributions, a dynamic mapping scheme was presented that combines cells of different sizes in macro-regions or subregions to meet the demands of each localization step.

The combination of data from inertial sensors to visual data has allowed us to improve ground level perception, allowing us to report elevation changes such as ramps and stairs as well as to identify curves and lines.

The proposed and tested hybrid indoor positioning model combined sensors, data, and algorithms to increase system accuracy and reduce response time. In this model, two algorithms played an important role: the LWPL and the i-PDR. LWPL has streamlined the mapping process to decrease data collection time and maintain a high quality of information. The i-PDR minimized the error margins of the PDR algorithm and allowed safer and more reliable indoor navigation.

The proposed hybrid model, which combines Wi-Fi location, inertial sensors, and computer vision, presented robust results when confronted with Wi-Fi location systems, inertial sensors, and computer vision. The hybrid system also showed lower position variation, providing a stable average precision, around 0.11 m.

This paper also introduced obstacle detection by a stereo-visual combination and a musical sound scheme. In this method, the stereo vision system extracts the protruding regions of obstacles placed in front of the user, triggering the sound corresponding to the suspect's visual region. Tests with stereo vision show that this method can detect obstacles effectively and correctly. However, by offering a new audio language, a training process is required to reduce users' learning time and make the proposed solution more visually impaired.

REFERENCES

[1] L. Mainetti, L. Patrono, and I. Sergi, "A survey on indoor positioning systems," in *Proc. 22nd Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2014, pp. 111–120, doi: 10.1109/SOFTCOM.2014.7039067.

[2] M. Nebel and B. Lankl, "Oscillator phase noise as a limiting factor in stand-alone GPS-indoor navigation," in *Proc. 5th ESA Workshop Satell. Navigat. Technol. Eur. Workshop GNSS Signals Signal Process. (NAVITEC)*, Dec. 2010, pp. 1–8, doi: 10.1109/NAVITEC.2010.5708038.

[3] H.-J. Chu, G.-J. Tsai, K.-W. Chiang, and T.-T. Duong, "GPS/MEMS INS data fusion and map matching in urban areas," *Sensors*, vol. 13, no. 9, pp. 11280–11288, Aug. 2013, doi: 10.3390/s130911280.

[4] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Survey Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013, doi: 10.1109/SURV.2012.121912.00075.

[5] W. Yan, X. Leng, Z. Wang, Y. Jin, J. Wang, and G. Shi, "High precision tri-axial MEMS gyroscope module based on redundant implementation and sensor fusion," in *Proc. IEEE Int. Conf. Cyber Technol. Autom., Control, Intell. Syst. (CYBER)*, Jun. 2016, pp. 376–379, doi: 10.1109/CYBER.2016.7574853.

[6] J. Bedoya Guapacha and S. C. A. Mantovanni, "Real time object detection and tracking using the Kalman filter embedded in single board in a robot," in *Proc. CHILEAN Conf. Elect., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Oct. 2017, pp. 1–6, doi: 10.1109/CHILECON.2017.8229695.

[7] Y.-S. Chiou, C.-L. Wang, and S.-C. Yeh, "Reduced-complexity scheme using alpha–beta filtering for location tracking," *IET Commun.*, vol. 5, no. 13, pp. 1806–1813, Sep. 2011, doi: 10.1049/iet-com.2010.0968.

[8] Y. Huang, Y. Zhang, N. Li, and L. Zhao, "Particle filter for nonlinear systems with multiple steps randomly delayed measurements," *Electron. Lett.*, vol. 51, no. 23, pp. 1859–1861, 2015, doi: 10.1049/el.2015.1899.

[9] H. Li, P. Zhang, S. Al Moubayed, S. N. Patel, and A. P. Sample, "ID-match: A hybrid computer vision and RFID system for recognizing individuals in groups," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2016, pp. 4933–4944, doi: 10.1145/2858036.2858209.

[10] Y. Liu, Q. Wang, J. Liu, and T. Wark, "MCMC-based indoor localization with a smart phone and sparse WiFi access points," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Lugano, Switzerland, Mar. 2012, pp. 247–252, doi: 10.1109/PerComW.2012.6197488.

[11] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals," *J. Signal Process. Syst.*, vol. 71, no. 3, pp. 287–296, Nov. 2012, doi: 10.1007/s11265-012-0711-5.

[12] W. Waqar, Y. Chen, and A. Vardy, "Incorporating user motion information for indoor smartphone positioning in sparse Wi-Fi environments," in *Proc. 17th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, Montreal, QC, Canada, Sep. 2014, pp. 267–274, doi: 10.1145/2641798.2641812.

[13] U. Bolat and M. Akcakoca, "A hybrid indoor positioning solution based on Wi-Fi, magnetic field, and inertial navigation," in *Proc. 14th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2017, pp. 16–22, doi: 10.1109/WPNC.2017.8250048.

[14] Z. Li, Z. Deng, W. Liu, and L. Xu, "A novel three-dimensional indoor localization algorithm based on multi-sensors," in *Proc. China Satell. Navigat. Conf. (CSNC)*, in Lecture Notes in Electrical Engineering, vol. 245. Berlin, Germany: Springer, 2013, pp. 623–631, doi: 10.1007/978-3-642-37407-4_58.

[15] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, Pittsburgh, PA, USA, Sep. 2012, pp. 421–430, doi: 10.1145/2370216.2370280.

[16] A. Kealy, G. Roberts, and G. Retscher, "Evaluating the performance of low cost MEMS inertial sensors for seamless indoor/outdoor navigation," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Palm Springs, CA, USA, May 2010, pp. 157–167, doi: 10.1109/PLANS.2010.5507132.

[17] V. Pradeep, G. Medioni, and J. Weiland, "Robot vision for the visually impaired," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, San Francisco, CA, USA, Jun. 2010, pp. 13–18, doi: 10.1109/CVPRW.2010.5543579.

[18] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 486–492, doi: 10.1109/IVS.2010.5548123.

[19] H. Xue, L. Ma, and X. Tan, "A fast visual map building method using video stream for visual-based indoor localization," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 650–654, doi: 10.1109/IWCMC.2016.7577133.

[20] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3357–3364, doi: 10.1109/ICRA.2017.7989381.

[21] H. Zou, H. Wang, L. Xie, and Q.-S. Jia, "An RFID indoor positioning system by using weighted path loss and extreme learning machine," in *Proc. IEEE 1st Int. Conf. Cyber-Phys. Syst., Netw., Appl. (CPSNA)*, Taipei, Taiwan, Aug. 2013, pp. 66–71, doi: 10.1109/CPSNA.2013.6614248.

[22] F. Ertam and G. Aydin, "Data classification with deep learning using tensorflow," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 755–758, doi: 10.1109/UBMK.2017.8093521.

[23] P. F. Alcantarilla, J. J. Yebes, J. Almazan, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 1290–1297, doi: 10.1109/ICRA.2012.6224690.

[24] C. Chen, W. Chai, and H. Roth, "A single frame depth visual gyroscope and its integration for robot navigation and mapping in structured indoor environments," *J. Intell. Robot. Syst.*, vol. 80, nos. 3–4, pp. 365–374, Jan. 2015, doi: 10.1007/s10846-014-0167-x.

[25] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for EKF-based structure from motion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 3498–3504, doi: 10.1109/IROS.2009.5354410.

[26] Z. Ni, S. Fu, B. Tang, H. He, and X. Huang, "Experimental studies on indoor sign recognition and classification," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Dec. 2014, pp. 2038–2050, doi: 10.1109/CIDM.2014.7008707.

[27] P. Ruangthong and S. Jaiyen, "Hybrid ensembles of decision trees and Bayesian network for class imbalance problem," in *Proc. 8th Int. Conf. Knowl. Smart Technol. (KST)*, Feb. 2016, pp. 39–42, doi: 10.1109/KST.2016.7440523.

[28] T. A. Heya, S. E. Arefin, A. Chakrabarty, and M. Alam, "Image processing based indoor localization system for assisting visually impaired people," in *Proc. Ubiquitous Positioning, Indoor Navigat. Location-Based Services (UPINLBS)*, Mar. 2018, pp. 1–7, doi: 10.1109/UPINLBS.2018.8559936.

[29] G. R. Rakate, S. R. Borhade, P. S. Jadhav, and M. S. Shah, "Advanced pedestrian detection system using combination of Haar-like features, AdaBoost algorithm and edgelet-shapelet," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, Dec. 2012, pp. 1–5, doi: 10.1109/ICCIC.2012.6510256.

[30] X. Yuan, X. Shan, and L. Su, "A combined pedestrian detection method based on Haar-like features and HOG features," in *Proc. 3rd Int. Workshop Intell. Syst. Appl.*, May 2011, pp. 1–4, doi: 10.1109/ISA.2011.5873387.

[31] L. Cheng, B. Song, Y. Dai, H. Wu, and Y. Chen, "Mobile robot indoor dual Kalman filter localisation based on inertial measurement and stereo vision," *CAAI Trans. Intell. Technol.*, vol. 2, no. 4, pp. 173–181, Dec. 2017, doi: 10.1049/trit.2017.0025.

[32] X. Li, J. Wang, and C. Liu, "Heading estimation with real-time compensation based on Kalman filter algorithm for an indoor positioning system," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 6, p. 98, Jun. 2016, doi: 10.3390/ijgi5060098.

[33] B. Koo, S. Lee, M. Lee, D. Lee, S. Lee, and S. Kim, "PDR/fingerprinting fusion indoor location tracking using RSS recovery and clustering," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 699–704, doi: 10.1109/IPIN.2014.7275546.

[34] P. Gharani and H. Karimi, "Context-aware obstacle detection for navigation by visually impaired," *Image Vis. Comput.*, vol. 64, pp. 103–115, Aug. 2017, doi: 10.1016/j.imavis.2017.06.002.

[35] J. Feng, L. Song, X. Huo, X. Yang, and W. Zhang, "Image restoration via efficient Gaussian mixture model learning," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1056–1060, doi: 10.1109/ICIP.2013.6738218.

[36] S. Kumar, P. Kumar, and S. Pandey, "Fast integral image computing scheme for vision based applications," in *Proc. 4th IEEE Uttar Pradesh Sect. Int. Conf. Elect., Comput. Electron. (UPCON)*, Oct. 2017, pp. 491–493, doi: 10.1109/UPCON.2017.8251098.

[37] A. Kalra and R. L. Chhokar, "A hybrid approach using sobel and canny operator for digital image edge detection," in *Proc. Int. Conf. Micro-Electron. Telecommun. Eng. (ICMETE)*, Sep. 2016, pp. 305–310, doi: 10.1109/ICMETE.2016.49.

[38] L. Chanama and O. Wongwirat, "A comparison of decision tree based techniques for indoor positioning system," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 732–737, doi: 10.1109/ICOIN.2018.8343215.

[39] Y. Zhou, H. Chen, Y. Huang, Y. Luo, Y. Zhang, and X. Xie, "An indoor route planning method with environment awareness," in *Proc. IGARSS - IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 2906–2909, doi: 10.1109/IGARSS.2018.8518507.

[40] C. Ruan, J. Luo, and Y. Wu, "Map navigation system based on optimal dijkstra algorithm," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst.*, Nov. 2014, pp. 559–564, doi: 10.1109/CCIS.2014.7175798.

[41] G. Liu, Y. Geng, and K. Pahlavan, "Direction estimation error model of embedded magnetometer in indoor navigation environment," in *Proc. IEEE 12th Int. Conf. Ubiquitous Intell. Comput. IEEE 12th Int. Conf. Autonomic Trusted Comput. IEEE 15th Int. Conf. Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 1842–1846, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.334.

[42] J. Krause, A. Perer, and E. Bertini, "INFUSE: Interactive feature selection for predictive modeling of high dimensional data," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1614–1623, Dec. 2014, doi: 10.1109/TVCG.2014.2346482.

[43] J. Dong, M. Noreikis, Y. Xiao, and A. Yla-Jaaski, "ViNav: A vision-based indoor navigation system for smartphones," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1461–1475, Jun. 2019, doi: 10.1109/TMC.2018.2857772.

[44] F. Wang, K. Jia, and J. Feng, "The real-time depth map obtainment based on stereo matching," in *Proc. Intell. Data Anal. Appl., Euro-China Conf. Intell. Data Anal. Appl. (ECC), Intell. Data Anal. Appl.*, vol. 535, J. S. Pan, V. Snášel, T. W. Sung, and X. Wang, Eds. Cham, Switzerland: Springer, 2017, pp. 138–144, doi: 10.1007/978-3-319-48499-0_17.

[45] E. B. Kaiser and M. Lawo, "Wearable navigation system for the visually impaired and blind people," in *Proc. IEEE/ACIS 11th Int. Conf. Comput. Inf. Sci.*, May 2012, pp. 230–233, doi: 10.1109/ICIS.2012.118.

[46] H. Yanzhang, T. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. Sim, T. Bagby, S. Chang, K. Rao, "Streaming end-to-end speech recognition for mobile devices," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6331–6385, doi: 10.1109/ICASSP.2019.8682336.

[47] E. Bartoli, A. D'Ausilio, J. Berry, L. Badino, T. Bever, and L. Fadiga, "Listener-speaker perceived distance predicts the degree of motor contribution to speech perception," *Cerebral Cortex*, vol. 25, no. 2, pp. 281–288, Sep. 2015, doi: 10.1093/cercor/bht257.

WALTER CHARLES SOUSA SEIFFERT SIMÕES (Associate Member, IEEE) was born in Santarém, Brazil, in 1975. He received the degree in data processing from the Institute of Technology of the Amazon (UTAM), in 2004, the Lato Sensu Specialist degree in project management from Gama Filho University (UFG) in 2006, the master's degree in informatics from the Federal University of Amazonas (UFAM), in 2014, and the Ph.D. degree in computer science from UFAM, in 2020. From 2002 to 2005, he was the Director of technology of information in Manaus. Since 2008, he has been a Professor and a Researcher in UNINORTE Laureate Universities of programming languages, robotics, and computational networks. He has authored two book chapters, more than 40 articles, and two registers of patents. His research interests are in computational vision, data fusion, data mining, and artificial intelligence, mapping, location and navigation systems, and autonomous robotic devices. He was the teacher honored of students who graduated in computer networks at UNINORTE Laureate from 2008 to 2019.

WALMIR ACIOLI E SILVA was born in Manaus, Brazil, in 1990. He received the degree in computer technician from the Federal Institute of Education, Science and Technology of Amazonas (IFAM), the B.Sc. degree in computer engineering from the Federal University of Amazonas, Brazil, in 2016, and the master's degree in electrical engineering from the Federal University of Amazonas, Brazil, in 2019. Has experience in electrical engineering, focusing on control and automation, acting on the following subjects embedded systems, vehicle automation, intelligent environments, and cyber-physical systems.

MATEUS MARTÍNEZ DE LUCENA was born in Manaus, Brazil, in 1994. He received the B.Sc. degree in computer engineering from the Federal University of Amazonas, Brazil, in 2018. He is currently pursuing the M.Sc. degree in computer science with the Federal University of Santa Catarina, Florianópolis Campus, Brazil. He has experience with embedded systems and systems security. He is currently working on the research and development of security in distributed systems at the Software/Hardware Integration Lab, Federal University of Santa Catarina.

NASSER JAZDI (Senior Member, IEEE) received the Diploma degree in electrical engineering and the Ph.D. degree in remote diagnosis and maintenance of embedded systems from the University of Stuttgart, Germany, in 1997 and 2003, respectively. In 2003, he joined the Institute of Industrial Automation and Software Engineering (IAS), University of Stuttgart as the Deputy Head, a Researcher, and a Lecturer. In 2009, he worked for two months as an Invited Researcher with Prof. Zadeh at the University of California at Berkeley. He is currently the Deputy Head of the Institute of Industrial Automation and Software Engineering, University of Stuttgart. He gives two lectures in software engineering and reliability and security of automation systems. His research interests are in software reliability in the context of IoT, learning aptitude for industrial automation, and artificial intelligence in industrial automation. He is a member of the VDE Association for Electrical, Electronic and Information Technologies, the VDI-GPP Software Reliability Group, and the Berkeley Initiative in Soft Computing (BISC).

VICENTE FERREIRA DE LUCENA, Jr. (Senior Member, IEEE) was born in São Paulo, Brazil, in 1965. He received the B.Sc. degree in electrical engineering from the Federal University of Amazonas, in 1987, the M.S. degree in electrical engineering from the Federal University of Campina Grande, in 1993, and the Ph.D. degree in electrical engineering (emphasis in industrial automation and software engineering) from the Universität Stuttgart, Germany, in 2002.

After several years of industry experience in Manaus, he joined the academic career at UFAM and IFAM. He was a Career Professor at the Federal Institute of Education, Science, and Technology of Amazonas (formerly ETFAM and CEFET-AM) from 1989 to 2012. Since 1991, he has been a Lecturer/Professor with the Federal University of Amazonas, where he has also been a Full Professor since 2019. He teaches courses in electrical engineering and computer engineering programs, both of which belonging to the Faculty of Technology. He also works as a Permanent Professor of the postgraduate program in electrical engineering (PPGEE), and in computer science (PPGI) of the Federal University of Amazonas (master's and Ph.D. Courses). He has interest in electrical engineering and computer science, with emphasis on embedded systems and software engineering, working recently in the following subjects industrial automation systems, applications of cyber-physical systems, new applications for industry (Industry 4.0) software reuse techniques, applications for electronic health and assistive technologies (eHealth systems), intelligent environments, and ambient intelligence (home and industrial usage). He has coordinated several research and development projects in the related topics with financing support of CNPq, FAPEAM, FINEP, SUFRAMA, and of companies located in the Industrial Hub of Manaus.

Dr. Lucena obtained the title of Senior Member of ACM, in 2013, and a Senior Member of ISA, in 2014. He served as a Representative of the Scientific Community in the Superior Council of the Foundation for Research Support of the State of Amazonas—FAPEAM, from 2003 to 2004, and as a Teaching Representative (elected by peers) in the research chamber of that institution from 2012 to 2018.