*Sensors* **2012**, *12*, 6764-6801; doi:10.3390/s120606764

OPEN ACCESS

*sensors*

**ISSN 1424-8220**

www.mdpi.com/journal/sensors

*Article*

# Enhancing Positioning Accuracy in Urban Terrain by Fusing Data from a GPS Receiver, Inertial Sensors, Stereo-Camera and Digital Maps for Pedestrian Navigation

**Baranski Przemyslaw * and Strumillo Pawel**

Institute of Electronics, Technical University of Lodz, Wolczanska 211/215, 90-924 Lodz, Poland;
E-Mail: pawel.strumillo@p.lodz.pl

* Author to whom correspondence should be addressed; E-Mail: przemyslaw.baranski@p.lodz.pl;
Tel.: +48-42-631-2685.

*Received: 6 March 2012; in revised form: 19 April 2012 / Accepted: 29 April 2012 /*
*Published: 25 May 2012*

**Abstract:** The paper presents an algorithm for estimating a pedestrian location in an urban environment. The algorithm is based on the particle filter and uses different data sources: a GPS receiver, inertial sensors, probability maps and a stereo camera. Inertial sensors are used to estimate a relative displacement of a pedestrian. A gyroscope estimates a change in the heading direction. An accelerometer is used to count a pedestrian's steps and their lengths. The so-called probability maps help to limit GPS inaccuracy by imposing constraints on pedestrian kinematics, e.g., it is assumed that a pedestrian cannot cross buildings, fences *etc*. This limits position inaccuracy to *ca.* 10 m. Incorporation of depth estimates derived from a stereo camera that are compared to the 3D model of an environment has enabled further reduction of positioning errors. As a result, for 90% of the time, the algorithm is able to estimate a pedestrian location with an error smaller than 2 m, compared to an error of 6.5 m for a navigation based solely on GPS.

**Keywords:** particle filtering; stereovision; digital maps; GPS; Monte Carlo; dead reckoning

## 1. Introduction

GPS-NAVSTAR (Global Positioning System-NAVigation Signal Timing And Ranging), popularly known as the GPS system, has considerably gained in civilian interests, since May 2000. Earlier, the system was practically reserved for military purposes. The positioning accuracy for the civilian sector was *ca.* 100 m due to an intentional error, called Selective Availability. According to report [1], the horizontal positioning error is less than 17 m for 99% of the time in average conditions or 17 m for 90% of the time in worse outdoor conditions. The error depends on many factors, like atmospheric conditions, sun activity, geographical location, terrain type, satellites' constellation, *etc*. In an open space, positioning errors are of *ca.* 2–3 m. However, in dense built-up areas, the location error may reach 100 m [2,3] or even more [4]. The error is introduced due to multipath propagation of signals transmitted by the satellites when there is no line-of-sight. A satellite signal is bounced off the walls of a building before finding its way to a GPS receiver. The propagation time of the signal is delayed and the GPS receiver miscalculates its location with a reference to the satellites.

There are many techniques to improve the location accuracy. Along coasts, for marine purposes, special ground DGPS (Differential GPS) reference stations broadcast differential corrections that allow a GPS receiver to eliminate tropospheric, ionospheric, ephemeris and clock errors. The overall error is reduced to 10 m with accuracy decreasing by 1 m with each 150 km increase in distance from the reference station. The corrections are transmitted on a *ca.* 300 kHz carrier frequency. A receiver must be, however, equipped with an additional antenna [5].

A-GPS (Assisted GPS) is a technique that downloads from the Internet the data concerning GPS satellite constellation [6]. Otherwise a GPS receiver may require up to 12.5 minutes to receive data about satellite constellation. By employing the A-GPS, the first fix is provided within few seconds. Additionally, a GSM modem can support a GPS receiver with a rough location which is obtained by measuring the strengths of signals from GSM base stations.

Apart from L1 = 1,227 MHz, a second frequency called L2C = 1,575 MHz, has been made available to the civilian sector with the aim of reducing the ionospheric and tropospheric errors which are a well defined function of frequency. Also, the so-called augmentation systems effectively reduce tropospheric and ionospheric errors by sending differential corrections from geostationary satellites directly to GPS receivers. The geostationary satellites imitate also GPS satellites and improve mainly the vertical accuracy. EGNOS (European Geostationary Navigation Overlay Service) works in Europe, WAAS (Wide Area Augmentation System) in the US, MSAS (Multi-functional Satellite Augmentation System) in Japan and GAGAN (GPS aided Geo-Augmented Navigation) in India.

Real Time Kinematics technique achieves the accuracy of millimetres in an open space and is designated for cartographic measurements. However, two GPS receiver are necessary. The base receiver is placed in a known position and calculates tropospheric, ionospheric, ephemerids and clock errors of satellites. The corrections are sent via a radio link to a mobile GPS receiver. In the trials reported in [4] RMS error of 2 cm on a mountain high-way was noted whereas trials in urban canyons yielded a 50 m RMS error.

All the above mentioned techniques are helpless against multipath propagation errors. Algorithms harnessed in car navigation alleviate these errors by taking advantage of car kinematics and a comparatively sparse network of roads, as described, e.g., in [7] or [8]. An error of 20 m is of no bigger importance to a driver. In case of navigating pedestrians, especially blind ones, the target accuracy should ideally not exceed the pavement width, *i.e.*, *ca.* 2 m.

This work presents a navigation scheme using GPS readouts, digital maps, inertial sensors and stereovision images with an aim of navigating a blind pedestrian. An accelerometer is used to detect pedestrian's strides and estimate their length. A gyroscope serves for estimating the heading direction. Those data sources help eliminate gross positioning errors and outliers. The digital maps are used twofold. Firstly, the so-called probability map is built to eliminate improbable user transitions, like traversing water ponds, crossing walls and buildings, *etc*. Secondly, a 3D model of the environment is built. The model is compared to stereoscopic images recorded by a mobile stereo camera. Interestingly, this comparison of 3D geometry of the environment provides good positioning accuracy in the surrounding of buildings, where GPS readouts are compromised. The proposed scheme employs the particle filter, also known as a sequential Monte Carlo method.

## 2. Related Work

The topic of pedestrian navigation including navigation aids for blind pedestrians has been described in many publications [9–11]. The first step to correct GPS readouts is to apply inertial sensors. Inertial sensors are mainly used in aviation to compute the orientation and position of an aircraft. Calculating the location requires double integration of the acceleration vector. Prior to these calculations the gravity acceleration must be removed, as accelerometers cannot distinguish gravity from an aircraft's accelerations. Therefore, the orientation of an aircraft with respect to the Earth's surface must be calculated from gyroscopes. The technique is known as INS (Inertial Navigation Systems) and warrants very precise and expensive laser sensors using the Sagnac effect. However, the strict implementation of INS using MEMS sensors (Micro Electro-Mechanical Systems) is useless after few seconds due to errors growing quadratically with time [12].

A travelled distance of a pedestrian can be estimated with surprisingly good accuracy by measuring the length of steps. This is done by analysing the acceleration in the gravity axis [10,13]. As a person walks, the body undulates according to the strides. The technique is accurate from 0.5% to 10%, depending on the gait style. ZUPT (Zero Velocity Update) technique exploits the fact that a foot is at rest for some short period. An accelerometer must be mounted to a foot which is an inconvenience, offset however by better accuracy compared to the previous method [14].

A heading direction can simply be read out from a magnetic compass, optionally supported by a gyroscope. A compass is sensitive to local distortions of a magnetic field due to cars, power lines *etc.* [10]. An electric tram can compromise a compass readouts within the radius of up to 100 m. A gyroscope, coupled by the Kalman filter, can reduce erroneous readouts [15].

The combination of GPS and inertial sensors readouts provides continuous estimates during GPS outages in harsh environments like tunnels, underground passages, dense urban areas *etc*. Positioning data from these two sources are usually integrated by the Extended Kalman filter or particle filter, which perform well when errors can be modelled by white noise which has the property of being uncorrelated with itself. However, the GPS errors are characterized by coloured noise [2,16]. This is because when a GPS receiver loses track of satellites its position is estimated by using the history of previous locations. Secondly, signals from satellites occluded by the same building are equally delayed, which introduces a bias in a given direction. This feature of errors corrupting GPS readouts was reported in earlier studies [14,17]. Jirawimut *et al.* [18] present an interesting concept where the height of buildings were utilized to check if a given satellite is occluded. The authors carried out simulation which yielded good results.

To improve the accuracy of dead-reckoning techniques, an additional source of the absolute pedestrian location should be introduced. Experiments in [4] showed that a combination of GPS and GLONASS navigation systems reduces root-mean-square (RMS) error by half and outliers by several times. Another solution is introducing the so-called probability map as a new source of data, which defines regions of a terrain that the user is most likely to enter or cross. A pedestrian is more likely to walk along pavements, parking places, paths, *etc*., rather than crossing walls, water ponds, *etc*. This cuts down the positioning errors as they start to aggregate. This map-based navigation concept proved its usefulness in outdoor pedestrian applications [19,20] as well as indoor positioning [21–23]. In built-up areas, a probability map limits inaccuracy to a width of an urban canyon, usually delineated by buildings on both sides.

A further improvement of positioning accuracy requires positioning with respect to known landmarks, *i.e.*, the so-called exteroception. The accuracy of landmarks' location should be approximately an order of magnitude better than the target accuracy of *ca.* 2 m. A vision-based technique called SLAM (Simultaneous Localization and Mapping) enables to build a map of the surrounding space and provide localization at the same time without employing any global navigation method. The SLAM based technique can be applied to build a 3D map of the environment. A review of SLAM techniques, modifications, results, *etc*., are given in book [24]. An application of SLAM for pedestrian navigation with results is given in [25]. Also a navigation system for the blind using a stereo-camera for tracing landmarks is presented in [26].

On the other hand, precise maps of outdoor or indoor environments may already be available, e.g., a plan of a building or plans of a city which are accurate to single centimetres, accuracy hardly achievable by any SLAM techniques. There are different sensors used to compare the environmental map with a robot's or pedestrian's location. Ultrasound sensors measure distance with 1% accuracy, being at the same time very cheap and compact. The maximum achievable distance is *ca.* 10 m. The angular resolution, however, is poor, ±15°. Laser scanners provide 0.1% distance measurement accuracy with a maximum distance of *ca.* 50 m. The angular resolution is of a fraction of a degree. The scanning is omnidirectional. The power consumption is *ca.* 10 W for longer distances. The size of a laser sensor is around 5 cm by 5 cm by 5 cm. The cost is rather high (*ca.* $4,000 USD depending on accuracy, maximum distance *etc*.). These sensors are commonly used in robotics. Time-of-Flight (ToF) cameras offer similar parameters as laser sensors providing 3D reconstruction of the scanned environment. The angle of view is *ca.* 30°. Application of laser scanners and ToF cameras is limited to indoor environments due sunlight interference.

In [27] ultrasound and laser sensing techniques are compared for indoor positioning of a robot. Work [23] presents an indoor localization system based on the particle filter. The user wears ultrasound sensors which measure a distance to adjacent walls. Therefore, the user location can be compared against the alignment of building walls. Application of an ultrasound sensor improved the positioning accuracy by 7 times in good trial conditions, *i.e.*, with no additional objects in the building's corridor, closed doors *etc*. A similar technique using a laser sensor mounted on a white cane is reported in [22]. A laser sensor detects corners of corridors what provides a comparison with a building's plan.

Finally, stereovision is a passive imaging technique that can be used indoors and outdoors. The cameras can be compact and inexpensive. However, stereovision imaging is limited to the environments featuring good lighting conditions, requires calibration of the stereovision optics and offers worse depth estimation accuracy than the earlier outlined active methods [28]. Moreover, demanding computations are required for calculating depth maps. In stereovision, depth accuracy can range from a few centimetres for objects located in a few metres range to a few metres for more distant objects. The so-called subpixel interpolation methods were proposed to improve depth estimation accuracy [29].

The work presented here combines global and local positioning techniques. GPS location estimates are augmented by dead reckoning techniques derived from inertial sensors, *i.e.*, gyroscopes and accelerometers. A stereo-camera is used for distance measurements. Finally, a digital map of the outdoor terrain is incorporated into the algorithm. Each of the listed source of positioning data features different error characteristic, e.g., in open spaces GPS readouts are most accurate whereas stereovision may yield large depth estimation errors. Conversely, for city canyons where GPS accuracy falters, stereovision and the digital map offer good positioning. A particle filtering algorithm is proposed to optimally fuse all data source.

## 3. A Prototype System for Pedestrian Positioning

A block diagram of the proposed system is presented in Figure 1. From the hardware point of view, the system is made up of three parts: a PC platform, stereo camera connected through FireWire interface and an electronic module housing a GPS receiver and 6DOF sensor. The data was acquired by walking through the University Campus with the stereovision camera attached to the chest of an experimenter. The laptop stored data from the electronic module and stereo camera. Then the data was processed off-line on a PC.

### *3.1. Electronic Module*

The built electronic module, see Figure 2, is a dedicated PCB with a microcontroller, 6DOF sensor and GPS receiver. The microcontroller reads out data from the 6DOF sensor, ADIS16355 at a rate of 820 Hz. Downsampling is carried out to limit data stream to the PC. An 8th order, lowpass Chebyshev filter is used to avoid aliasing. Samples are sent to the computer at a rate of 205 Hz. The module sends data through the USB interface.

#### 3.1.1. 6DOF Sensor

The 6DOF sensor, ADIS16355 from Analog Devices, comprises a 3-axial accelerometer and 3-axial gyro. The data from the sensor is read out through a digital interface, SPI. The samples are of 14-bit resolution whereby practically the three least significant bits are random. The ADIS16355 has been superseded by much less noisy ADIS16375. The static parameters of the former were investigated by the Allan variance [30].

**Gyroscope.** The gyroscope is used to estimate the pedestrian orientation. Angular velocity $\omega(t)$, returned by the gyroscope, is corrupted mainly by white and flicker noise. The relative direction change $\Delta\phi$ is calculated by integrating the angular velocity according to Equation (1).

$$\Delta\phi(t) = \int_0^t \omega(T)dT = \int_0^t (\hat{\omega}(T) + \omega_w(T) + \omega_f(T))dT \tag{1}$$

$$\Delta\phi(t) = \Delta\hat{\phi}(t) + \phi_{\text{ARW}}(t) + \phi_F(t) \tag{2}$$

$$\phi_{\text{ARW}}(t) = \int_0^t \omega_w(T)dT \tag{3}$$

$$\phi_F(t) = \int_0^t \omega_f(T))dT \tag{4}$$

where $\hat{\omega}(t)$ denotes the true angular velocity, $\omega_w(t)$ white noise and $\omega_f(t)$ flicker noise in angular velocity readouts. $\Delta\hat{\phi}(t)$ is the true change in the heading direction. Integrated white noise $\omega_w(t)$ results in a first-order random walk, called in this case angular random walk (ARW), and is denoted by $\phi_{\text{ARW}}(t)$ in Equations (2) and (3). Angular random walk $\phi_{\text{ARW}}(t)$ is described by the Gaussian distribution but it is not a stationary process as its standard deviation is growing with the square root of the integration time [31]. For the sensor used in the project the standard deviation of $\phi_{\text{ARW}}$ grows at a rate of $2.77°/\sqrt{\text{h}}$.

**Figure 1.** The block diagram of the system. The electronic module is a dedicated electronic circuit comprised of a 6DOF sensor, GPS receiver and microcontroller. The module is connected with the PC via USB interface. The stereovision block represents a stereo camera.

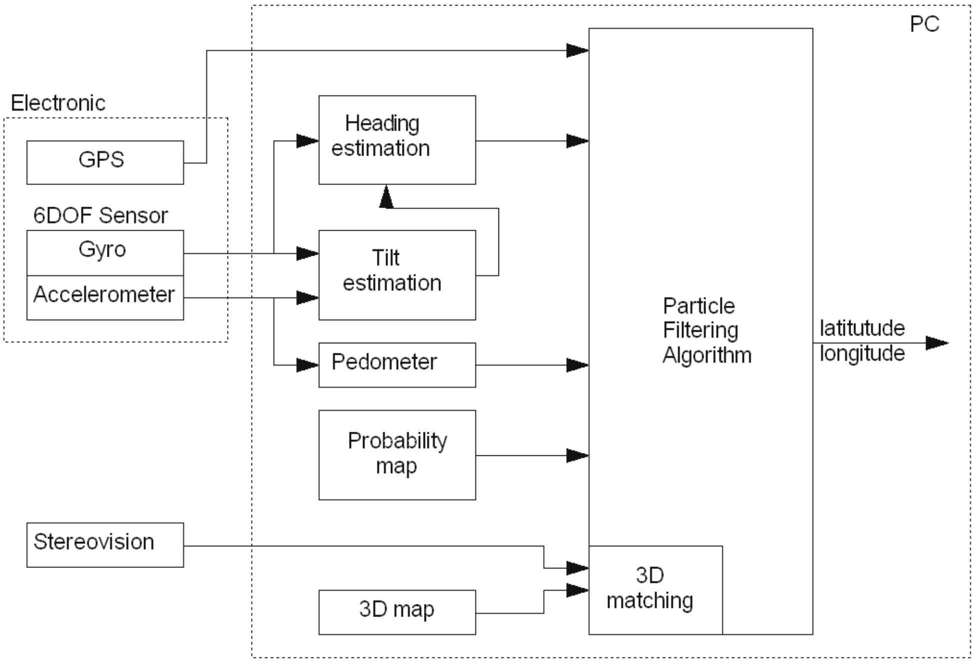
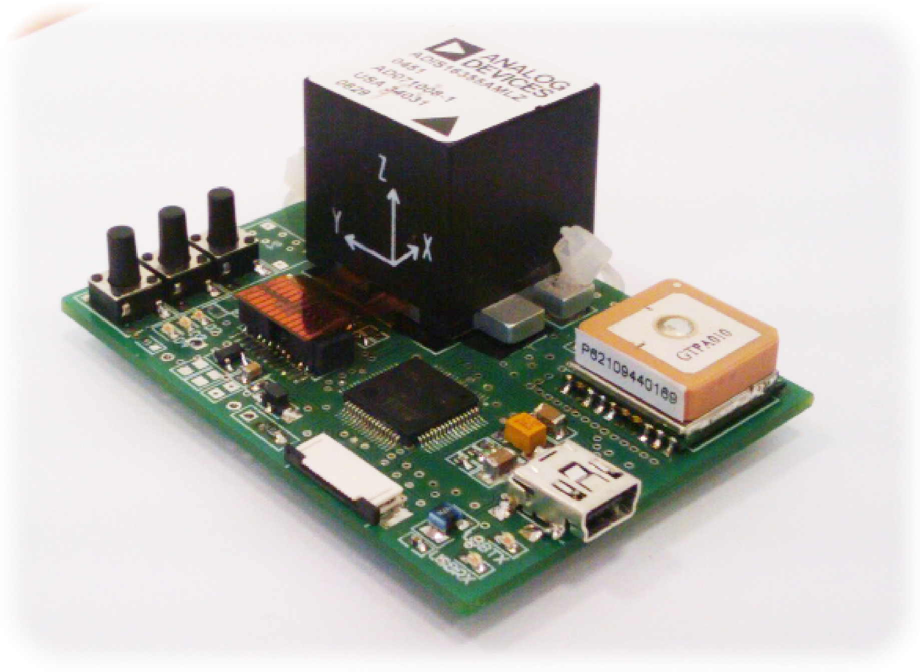


**Figure 2.** A picture of the built electronic module. The black box is a 6DOF sensor—3-axial gyroscope and 3-axial accelerometer. The orange cuboid is an antenna of a GPS receiver. The module is connected with the PC through a mini-USB socket.



Similarly, integrated flicker noise $\omega_f(t)$ causes an angular error $\Delta\phi_F(t)$, according to Equation (4). The values of $\Delta\phi_F$ are described by the Gaussian distribution whose standard deviation grows proportionally with the time and therefore it is a non-stationary process. In case of the gyroscope used in the project, the standard deviation of $\Delta\phi_F$ grows at a rate of $48^\circ/\mathrm{h}$. Therefore, confidence intervals for estimating $\Delta\phi(t)$ widens with the time $t$. After an hour, a motionless gyro can drift by $\pm 144^\circ$ for $\pm 3\sigma$ confidence interval. Hence, dead reckoning is justified for short periods only, e.g., during GPS outages.

Inertial sensors also suffer from other errors like non-linearity or temperature random bias. The former error is $\pm 0.3^\circ/\mathrm{s}$. The experienced random bias due to temperature was much higher than the manufacturer claimed and it reads 0.5$^\circ/\mathrm{s}$ for $\pm 3\sigma$ confidence interval. This is, however, easy to compensate by leaving the sensor motionless and measuring the constant offset to be later subtracted from the readouts.

**Accelerometer.** The accelerometer is used in the project to estimate the user's steps and their lengths. This method is more precise than integrating twice the acceleration readouts to obtain the displacement. The accelerations readouts are also used to estimate the gravity direction which is necessary for correct calculation of the orientation change. By analogy, velocity random walk is a product of integration of white noise in acceleration. On average, the velocity random walk was 0.5 $\mathrm{m/s/\sqrt{h}}$. Flicker noise introduces an error of 9.3 $\mathrm{m/s/h}$. The displacement errors are much larger since they are products of integrating velocity errors. Any offset in acceleration readouts grows quadratically with the time. This quickly leads to aggregation of errors and renders the analytical approach ineffective after a dozen of seconds.

### 3.1.2. GPS Receiver

The PCB board houses a standard GPS receiver with an integrated ceramic antenna, measuring 15 mm by 15 mm. The receiver is built on the MTK MT3329 chipset and supports WAAS, EGNOS, MSAS, GAGAN corrections. GPS readouts are based on the WGS84 coordinate system aligned in the centre of the Earth, being represented by an ellipsoid. The conversion to a local coordinate system follows in two steps. Firstly, the polar coordinates, *i.e.*, latitude, longitude and height above the ground, are transformed to the WGS84 Cartesian coordinates $(x_{\text{WGS84}}, y_{\text{WGS84}}, z_{\text{WGS84}})$. This step, along with associated errors and corrections, is described in detail in [32]. Secondly, the local coordinates $(x, y, z)$ are obtained by simplified Helmert's transformation (5).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R}\left(\begin{bmatrix} x_{\text{WGS84}} \\ y_{\text{WGS84}} \\ z_{\text{WGS84}} \end{bmatrix} - \mathbf{T}\right) \tag{5}$$

The rotation $\mathbf{R}$ and translation $\mathbf{T}$ matrices are structured in such a way that the centre of the local coordinate system is on the Earth's surface. Versor $\mathbf{1_x}$ is directed along with longitude, $\mathbf{1_y}$ with latitude (towards the pole), $\mathbf{1_z}$ is determined by $\mathbf{1_x}$ and $\mathbf{1_y}$. The conversion accuracy is in an order of 10 cm for 20 km from the centre of the local coordinate system.

A GPS receiver provides the HDOP parameter (Horizontal Dilution of Precision) informing about the accuracy of the estimated coordinates, *i.e.*, latitude and longitude (6). Higher values of HDOP should inform a dead reckoning filter to trust to other sources of positioning (e.g., inertial sensors) and disregard GPS readouts.

$$\text{HDOP} = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{6}$$

The standard deviations of estimating $x$ and $y$ on the Earth surface are denoted by $\sigma_x$ and $\sigma_y$ respectively. Providing that $x$ and $y$ have the Gaussian distribution and are independent, the probability density function of distance error $r_e$ is given by the Rayleigh distribution [33], which is not a monotone function of its argument. The function has a maximum for $r_e > 0$. Tests showed that the function defined by Equations (7), (8) and (9) yields definitively superior results. This can be explained by a poor relationship between the HDOP parameter and the error $r_e$. Since $r_e$ can assume only positive values, the Gaussian function $p_{\text{GPS}}(r_e, \sigma_{\text{GPS}})$ is accordingly scaled by a factor of 2. The coefficient $\beta_{\text{GPS}}$ in Equation (9) is chosen by trial and error.

$$p_{\text{GPS}}(r_e, \sigma_{\text{GPS}}) = \frac{2}{\sqrt{2\pi}\sigma_{\text{GPS}}} \exp\left(-\frac{r_e^2}{2\sigma_{\text{GPS}}^2}\right) \tag{7}$$

$$r_e = \sqrt{\tilde{x}^2 + \tilde{y}^2} \tag{8}$$

$$\sigma_{\text{GPS}} = \beta_{\text{GPS}}\text{HDOP} \tag{9}$$

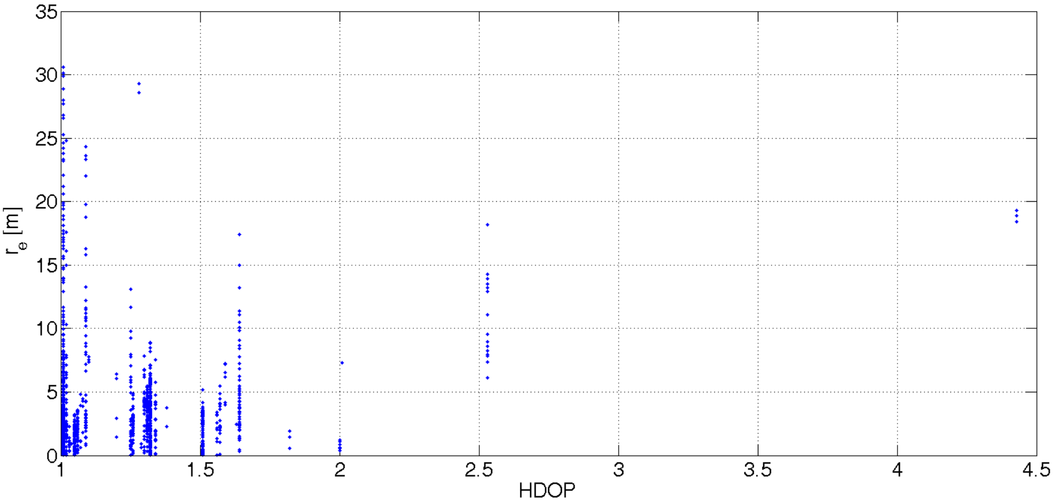
If a GPS receiver loses track of satellites, prediction of its position is based on the previous velocities and position. The first order autocorrelation coefficients for $\tilde{x}$ and $\tilde{y}$ were calculated for a 2.6 km path—see Equation (10). They had considerable values of $a_x \approx a_y \approx 0.86$. Thus $\tilde{x}$ and $\tilde{y}$ can be modelled by Equation (11), which is known as exponentially correlated noise.

$$a = \frac{\sum_{t=1}^{n-1} \tilde{x}(t)\tilde{x}(t+1)}{\sum_{t=1}^{n-1} \tilde{x}(t)^2} \tag{10}$$

$$\begin{bmatrix} \tilde{x}(t) \\ \tilde{y}(t) \end{bmatrix} = \begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix} \begin{bmatrix} \tilde{x}(t-1) \\ \tilde{y}(t-1) \end{bmatrix} + \begin{bmatrix} v_x(t) \\ v_y(t) \end{bmatrix} \tag{11}$$

where $v_x(t)$ and $v_y(t)$ are random realizations of white noise, whereby $E[v_x^2(t)] = (1 - a_x^2)\sigma_x^2$ and $E[v_y^2(t)] = (1 - a_y^2)\sigma_y^2$. This is a very important point. There are periods of e.g., 30 s when GPS readouts stray away by 40 m in one direction. A dead reckoning system, weighting GPS coordinates with inertial sensors' readouts to produce a better position estimate, will succumb to this bias error sooner or later. When the GPS receiver regains good visibility with satellites, the estimated position will be all of a sudden in a quite different location. The dead reckoning filter will have to take some time to adapt to a new, accurate GPS position. During this time the filter output will be encumbered with large errors. A way out is to recognize the situation and reinitialize the filter. To account for these random bias errors another source of absolute positioning should be introduced. Maps, video data are examples thereof.

**Figure 3.** The relation between the values of the HDOP parameter and the distance errors $r_e(t)$.



Another problem associated with GPS readouts is the correspondence of the HDOP parameter with the actual accuracy of latitude and longitude. Equation (6) shows that the value of HDOP is interpreted as the distance error. Figure 3 sheds light on the problem. It was noticed that the interpretation of the HDOP parameter depends on a GPS receiver at hand. The correlation coefficient (12) shows a poor linear dependence between distance errors $r_e(t)$ and the values of HDOP. The number of observed satellites and position error is not correlated. This confirms the statement about the necessity of additional positioning data.

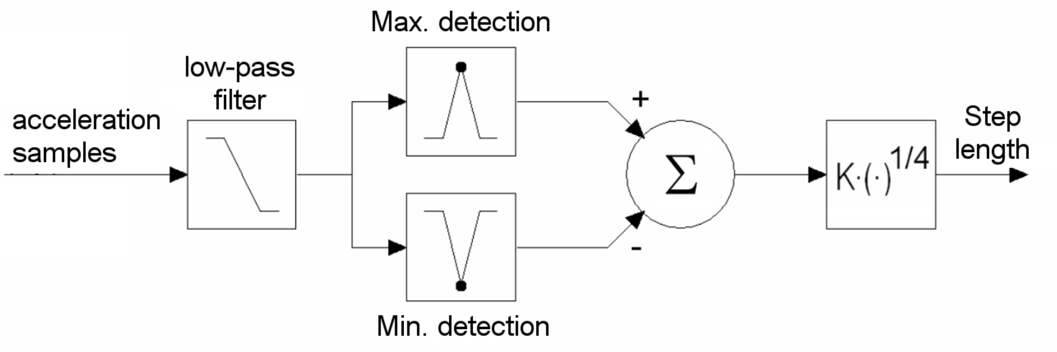
$$\rho(r_e(t), \mathrm{HDOP}(t)) = \frac{\mathrm{Cov}(r_e(t), \mathrm{HDOP}(t))}{\sqrt{\mathrm{Var}(r_e(t))\mathrm{Var}(\mathrm{HDOP}(t))}} = 0.19 \tag{12}$$

### *3.2. Pedometer*

When a person walks, the body undulates in unison with the steps. The step-detection algorithm examines the acceleration of the body only in the vertical direction (parallel to the gravity axis). The gravity axis is obtained from the algorithm presented in Section 3.3. The acceleration signal also reflects knee movements, body swings, sensor jerks and so forth. A 3 Hz 4-order low-pass Butterworth filter was applied to eliminate undesired constituents. The step detection algorithm is presented in Figure 4. The minimum $A_{\mathrm{min}}$ and maximum $A_{\mathrm{max}}$ acceleration in the filtered signal are detected. The difference between the peaks is raised to the empirical power of $0.25$. The result is multiplied by the scale factor $K$, depending on the individual. Equation (13) expresses an estimated step length $d$. Related work can be found in [10,13,34]. Step length accuracy measurements are between 1% to 10% depending on an individual and gait style.

$$d = (A_{\mathrm{max}} - A_{\mathrm{min}})^{0.25} \cdot K \tag{13}$$

**Figure 4.** The algorithm for estimating the number of steps and their lengths.



### *3.3. Tilt Estimation*

Tilt estimation, *i.e.*, estimation of the direction of the gravity axis, is important for two reasons:

1. Calculation of the relative change in the heading direction. When the sensor is not levelled, the gyroscope's axis does not coincide with the pivotal axis of the pedestrian. When the user turns, the relative change in the heading direction is underestimated. For example, when the gyroscope axis is tilted by $25°$ from the pivotal axis, the angular velocity is underestimated by $\approx 10\%$. Consequently, the relative direction change might be $81°$ instead of $90°$—*c.f.* Figure 5.
2. Estimation of the camera orientation with respect to the ground. The stereo camera is used to measure distance to nearest objects like buildings. When the camera is not levelled the measured distance will be overestimated.

A human rotates along the gravity axis when changing the walking direction. Thus, the task of estimating the correct change in direction comes down to finding the gravity axis. The problem is known in aeronautics as Inertial Navigation Systems. The estimation of six coordinates of an aeroplane (three for orientation and three for position) is not a trivial task. A standard approach based on off-the-shelf sensors and cosine direction matrices is very inaccurate after few dozens of seconds [12]. However, humans move with much smaller accelerations. The dominant acceleration comes from the gravity. When a person moves, the body is usually aligned vertically. The pitch and roll are limited to ≈ ±30°. These assumptions help greatly to design an algorithm estimating the gravity axis. The algorithm uses Kalman filter to integrate the readouts from the gyroscopes and accelerometers. The angles between the sensor's axes and the gravity axis (see Figure 5) can be calculated from Equations (14) and (15), given the sensor is not moving.

$$\alpha_x = \arccos(\frac{a_x}{g});\ \beta_x = \frac{\pi}{2} - \alpha_x \tag{14}$$

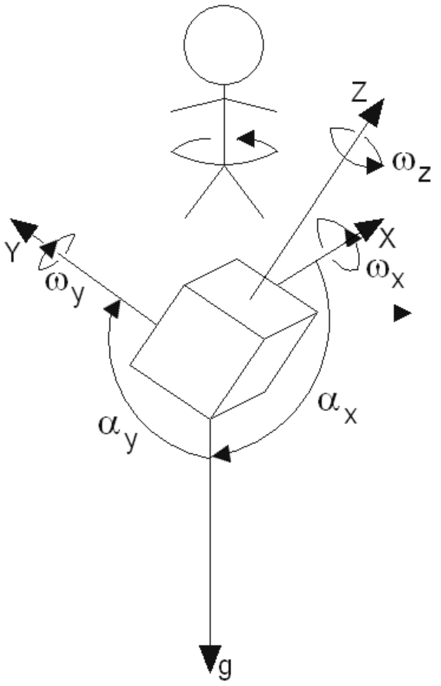
$$\alpha_y = \arccos(\frac{a_y}{g});\ \beta_y = \frac{\pi}{2} - \alpha_y \tag{15}$$

$$\alpha_z = \arccos(\frac{a_z}{g}) \tag{16}$$

where $a_x$, $a_y$ and $a_z$ are accelerations measured by the 3-axial accelerometer, $g$ is the gravity constant, 9.81 $\mathrm{m/s}^2$. When a user walks, the accelerations will fluctuate by ±0.25 g, depending on the walking style. The acceleration vector due to walking is treated here as measurement noise. Hence, calculating the direction of the gravity axis from Equation (16) would result in an error of $15°$. The accelerations need to meet the constraint Equation (17).

$$\sqrt{a_x^2 + a_y^2 + a_z^2} = g \tag{17}$$

**Figure 5.** The local coordinate system of the 6DOF sensor. $g$ denotes the gravity acceleration. $\omega_x$, $\omega_y$ and $\omega_z$ are angular velocities which are clock-wise oriented with their rotation axes. $\alpha_x$, $\alpha_y$ and $\alpha_z$ are angles between the sensor's axes and the gravity.



This reduces the number of degrees of freedom by one. Gyroscopes can be used to improve the estimation of the sensor orientation. The transition equation of the Extended Kalman Filter (EKF) is given by Equations (18) and (19). Appearance of angular velocity $\omega_y$ causes the $X$ axis of the sensor changing its orientation with respect to the gravity. This change also depends on the orientation of the $Y$ axis and is expressed by the $\sin\alpha_y$ term.

$$\mathbf{x}(k) = \begin{bmatrix} \alpha_x(k-1) \\ \alpha_y(k-1) \end{bmatrix} = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \mathbf{w}(k-1)) = \tag{18}$$

$$= \begin{bmatrix} \alpha_x(k-1) + (\omega_y(k-1) + w_y(k-1))dT \sin\alpha_y(k-1) \\ \alpha_y(k-1) - (\omega_x(k-1) + w_x(k-1))dT \sin\alpha_x(k-1) \end{bmatrix} \tag{19}$$

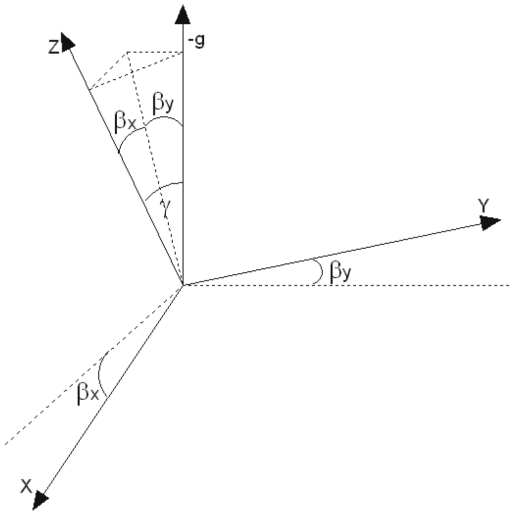
$k$ denotes time instant, where $t = k \cdot dT$ and $1/dT$ = 205 Hz. $w_y(k)$ and $w_x(k)$ are the gyroscope errors. The measurement equation of the EKF is given by Equation (20)

$$\mathbf{y(k)} = \begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix} = \mathbf{g}(\mathbf{x}(k), \mathbf{v}(k)) = \begin{bmatrix} g\cos\alpha_x(k) - v_x(k) \\ g\cos\alpha_y(k) - v_y(k) \end{bmatrix} \tag{20}$$

The accelerometers readouts are corrupted by noise $\mathbf{v}(k)$. Further implementation of this Extended Kalman Filter is straightforward and follows through Taylor linearisation. Details can be found in [35]. Having estimated $\alpha_x$ and $\alpha_y$ and complementary angles $\beta_x$ and $\beta_y$, the angle $\gamma$ between the $Z$-axis gyroscope and the inverted gravity axis (see Figure 6) can be calculated from Equation (21).

$$\gamma = 2\arcsin\frac{\sqrt{\sin^2\beta_x + \sin^2\beta_y}}{2} \tag{21}$$

**Figure 6.** The relationship between the angles $\beta_x$ and $\beta_y$ and the angle $\gamma$ which describes the sensor tilt and is measured between the sensor's Z-axis and the $-g$ axis.



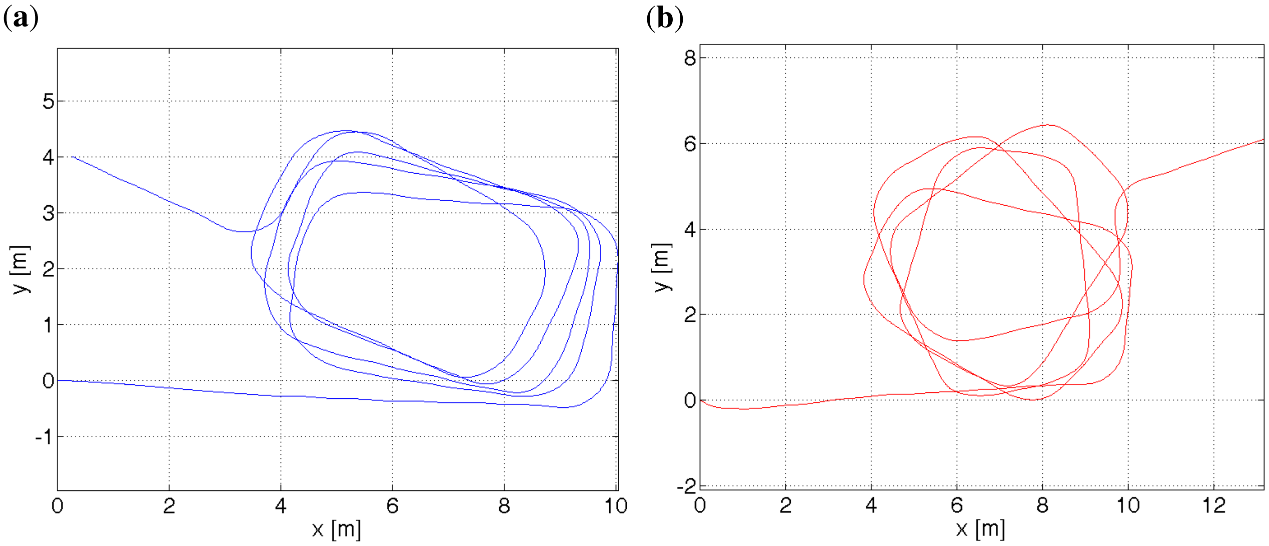
The angular velocity can be corrected by Equation (22). The relative heading direction should be calculated from Equation (23).

$$\hat{\omega_z}(t) = \omega_z(t)/\cos(\gamma) \tag{22}$$

$$\Delta\phi(t) = \int\limits_0^t \hat{\omega_z}(t)dt \tag{23}$$

As mentioned before, this approach is valid for limited range of pitch and roll, due to singularity in Equation (22) when $\gamma$ approaches $\pi/2$. The measurement of $\gamma$ can be included from Equation (16), however with a small benefit due to considerable fluctuations of the corresponding acceleration $a_z$. The average error of estimating $\gamma$ was measured to be $0.5°$ with a standard deviation of $0.45°$. Figure 7(a,b) shows a trial with the sensor tilted by ≈15°.

**Figure 7.** (**a**) Path plotted for the pedometer and gyroscope readouts. The gyroscope readouts are corrected by the gravity estimation algorithm. (**b**) Path with no gyroscope correction. Points $(x, y) = (0, 0)$ denote the starting point. There were 20 turns, each $90°$. The trial was started and ended in the same point. The difference between the heading direction at the beginning and at the end was $\approx 21°$, which corresponds to $1°$ of error per every rotation.



### 3.4. Heading Estimation

A gyroscope provides an angular velocity $\omega$. A direction change can be found by integrating $\omega$ (*c.f.*, Equation (23)). Thus a gyroscope can provide only a relative orientation as opposed to a magnetic compass. Experiments with a magnetic compass showed its susceptibility to magnetic field distortions due to tram lines, DC power lines with no return cable, cars changing a local magnetic field, *etc*. The magnitude of the Earth's magnetic field is *ca.* 50 $\mu T$. By comparison, a departing tram produces 120 m away a magnetic induction of 5 $\mu T$. Hence, a magnetic compass was not included. The absolute pedestrian orientation is sorted out run-time by the particle filtering algorithm based on other data. A magnetic compass could give a rough approximation $\pm 90°$ until the absolute orientation has been found and then switched off.

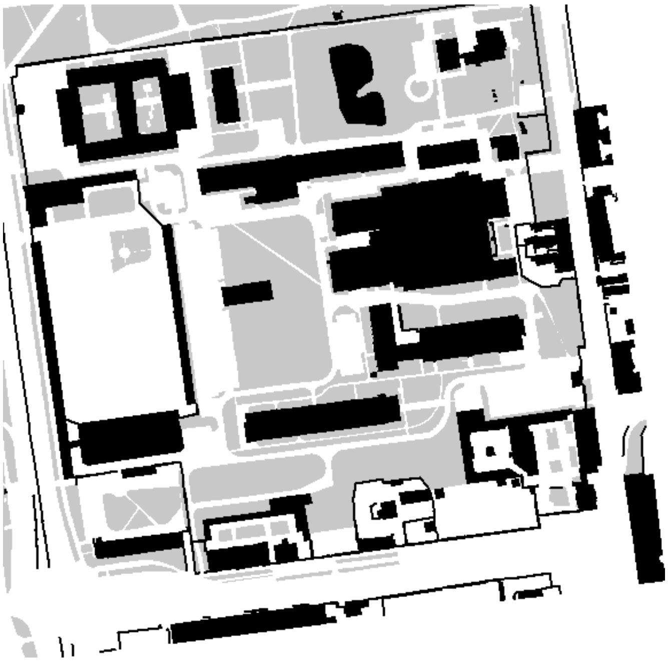
### *3.5. Probability Map*

The digital map is divided into three types of area. The areas have their associated weights:

- forbidden areas—buildings, ponds, walls, fences *etc.*, $w_{\text{map}}(x, y) = 0.0$,
- probable areas—lawns, fields *etc.*, $w_{\text{map}}(x, y) \in (0; 1)$,
- preferred areas—pavements, streets, squares, alleys *etc.*, $w_{\text{map}}(x, y) = 1.0$.

$w_{\text{map}}(x, y)$ denotes the type of an area at $(x, y)$ coordinates. Each weight corresponds to the probability of the user being in a given area. Since the system is destined for outdoor navigation, it is rather improbable that the user will traverse buildings, walls and fences—*c.f.*, Figure 8. The user is likely to travel along pavements and alleys, however, he or she can also walk across lawns, although less likely. The probability map mitigates the problem of gyroscope drifts and stepmeter inaccuracy. The particle filtering approach to the discussed problem will be explained in Section 3.9. Here the term particles denotes a collection of hypothetical locations of the user and the weights associated with each particle is understood as the probability of the user being at this location. When particles enter a forbidden area, they will be gradually eliminated. Particles that move along a preferred area will be preserved. Therefore, the direction will align itself and the direction drift will be reduced. The weight assigned for probable areas should not be too low, because otherwise when a user crosses a wide street, the particles' weights will be successively reduced and consequently die out. Consequently, the user's location will not be correctly calculated. The project was developed with a view of pedestrians, especially blind ones.

**Figure 8.** An example of a probability map of the University Campus. Forbidden, probable and preferred areas are denoted by black, grey and white respectively. The streets at the campus are paved and have the same texture as pavements. A blind user would rather refrain from walking through lawns.



### *3.6. Stereovision Camera*

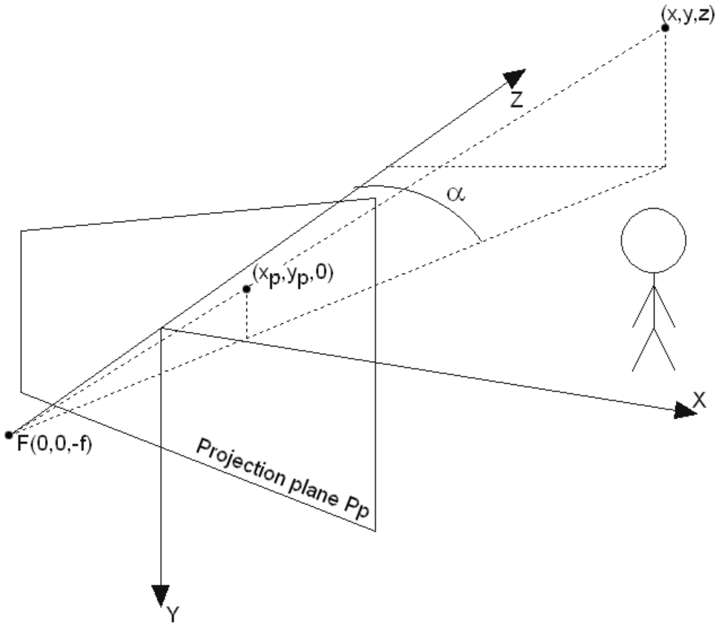
#### 3.6.1. Introduction

A stereovision camera [28] is comprised of two cameras which are shifted away along the $X$ axis, *c.f.*, Figure 9. The distance between the cameras is called the baseline of a stereo system and is denoted by $B$. The baseline distance makes the same scene point $P$ to be visible in the left and right camera at coordinates $x_L$ and $x_R$ respectively. The difference between the $x_L$ and $x_R$ values is called the disparity, and will be denoted by $x_d$. The $z$ coordinate of the observed object can be calculated from Equation (24), where $f$ is the focal length.

$$z = \frac{Bf}{x_d} \tag{24}$$

For the legibility sake, the stereovision system can be treated as one camera, whose simplified coordinate system is shown in Figure 9. The $Z$ axis coincides with the optical axis of the camera model.

**Figure 9.** The local coordinate system of the stereovision camera. $F$ is the focus of the camera, and $f$ is the focal length, $(x, y, z)$ are the coordinates of a point in the camera coordinate system. This point has the coordinates $(x_P, y_P, 0)$ on the projection plane $P_P$ of the camera and $\alpha$ denotes the angle from the $Z$ axis at which point $(x, y, z)$ is visible by the camera.



The coordinates of a point $(x, y, z)$ are projected onto the camera projection plane $P_P$ according to the transformation (25).

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \frac{f}{z+f} & 0 & 0 \\ 0 & \frac{f}{z+f} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{25}$$

The depth of a point $(x, y, z)$, denoted by $d_p(x, y, z)$, is understood as the distance from the point to the projection plane and equals Equation (26).

$$d_p(x, y, z) = z \tag{26}$$

The distance from the Y axis of the camera to a point with the $(x, y, z)$ coordinates is denoted by $d_Y(x, y, z)$ and calculated from Equation (27).

$$d_Y(x, y, z) = \sqrt{x^2 + z^2} \tag{27}$$

Let us define a function which returns the distance from the $Y$ camera's axis to the nearest point visible at angle $\alpha$ from the optical axis $Z$, *c.f.* Figure 9. This function is defined by Equation (28).

$$d_\alpha(\alpha) = \min(d_Y(x, y, z)); \text{ where } \tan\alpha = \frac{x_p}{f} = \frac{x}{z} \tag{28}$$

The data from the stereovision camera is digitized. The $(x_p, y_p)$ coordinates are quantized and converted to units called pixels. The transformed coordinates are denoted by $(x_{\text{pix}}, y_{\text{pix}})$. The conversion follows through Equation (29).

$$\begin{bmatrix} x_{\text{pix}} \\ y_{\text{pix}} \end{bmatrix} = \begin{bmatrix} \text{Round}(\frac{f_{\text{pix}}}{f} x_p) \\ \text{Round}(\frac{f_{\text{pix}}}{f} y_p) \end{bmatrix} \tag{29}$$

where the Round() operator denotes rounding to a nearest integer number, $f_{\text{pix}}$ is the focal length of the camera expressed in pixels and is provided by the manufacturer in the technical specification. The difference between the maximal and minimal value of $x_{\text{pix}}$ is called the horizontal resolution of the camera and is denoted by $x_{\text{res}}$. The vertical resolution $y_{\text{res}}$ is defined by analogy. A depth map or a depth picture **L** is a two dimensional map defined by Equation (30).

$$\mathbf{L}(x_{\text{pix}}, y_{\text{pix}}) = \min(d_p(x, y, z)); \; (x, y) \overset{(25),(29)}{\longrightarrow} (x_{\text{pix}}, y_{\text{pix}}) \tag{30}$$

The coordinates $(x, y)$ are related with $(x_{\text{pix}}, y_{\text{pix}})$ through transformations (25) and (29). Equation (30) formally says that a stereo camera provides the depth to nearest points that are not occluded. This is so-called 2.5 dimension image. Usually, the depth for a given point $(x_{\text{pix}}, y_{\text{pix}})$ is represented by a floating point-number. Using the identity (24), the disparity map **K** can be defined by Equation (31).

$$\mathbf{K}(x_{\text{pix}}, y_{\text{pix}}) = \frac{Bf}{\mathbf{L}(x_{\text{pix}}, y_{\text{pix}})} \tag{31}$$

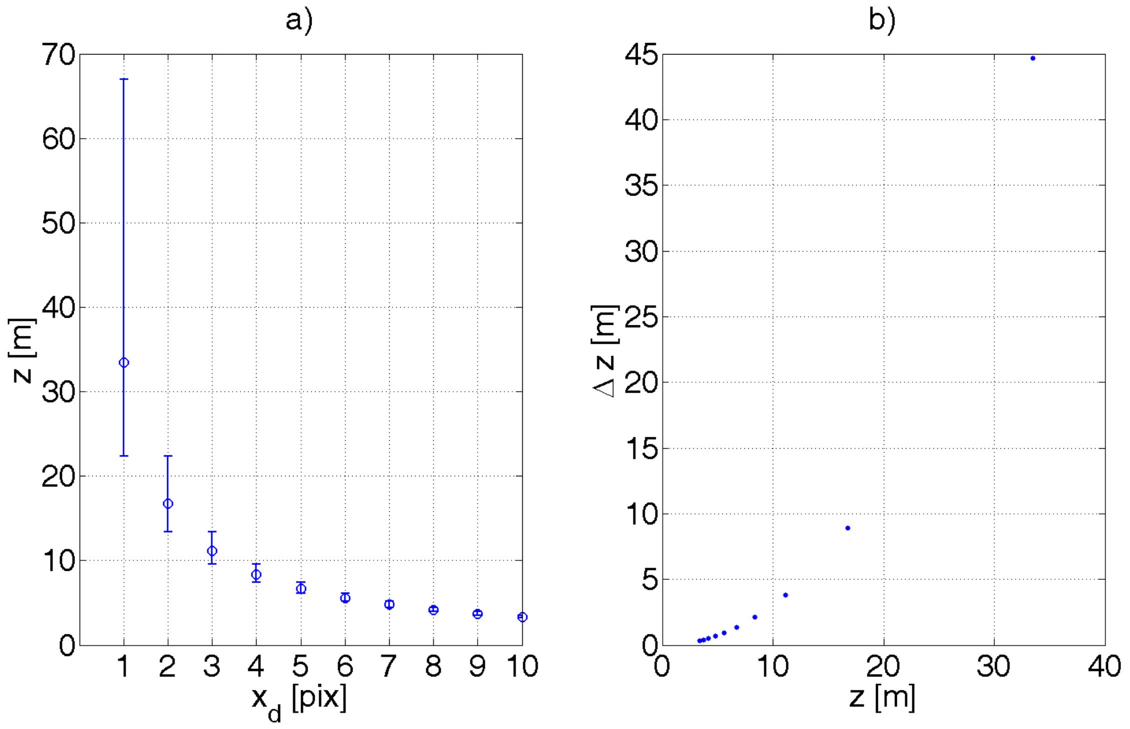
As a matter of fact, stereovision cameras first determine the disparity map **K** and then calculate the depth map **L**. The disparity $x_d$ is calculated in digital cameras in pixels, thus **K** should be a map containing integer numbers. An example of a depth map is presented in Figure 12(b). Advanced stereo cameras perform subpixel interpolation, e.g., the camera used in the project calculates the disparity with the resolution of 0.25 pixel and then performs filtering and interpolation on the depth map.

### 3.6.2. Error Analysis

As the object recedes from the camera, the disparity $x_d$ decreases. An error in calculating the shift $x_d$ becomes more significant. Calculating the derivative of $z$ with respect to $x_d$ in Equation (24) gives Equation (32). The accuracy of determining the distance $z$ decreases with its squared value. As the stereo-camera is digital, the values of $x_L$, $x_R$ are quantized and so is $x_d$. It means that $z$ manifests discontinuities as $x_d$ changes to an adjacent quantum. Let $\epsilon$ denote the size of the pixel in metrical units. For the sake of simplicity, $x_d$ will have units of metres or pixels, whereby the appropriate unit will be clear from the context or mentioned implicitly. Let us assume at this stage that the error of determining $x_d$ can be $\pm\frac{\epsilon}{2}$ or $\pm\frac{\text{pixel}}{2}$. Figure 10 exemplifies the effect of such quantization.

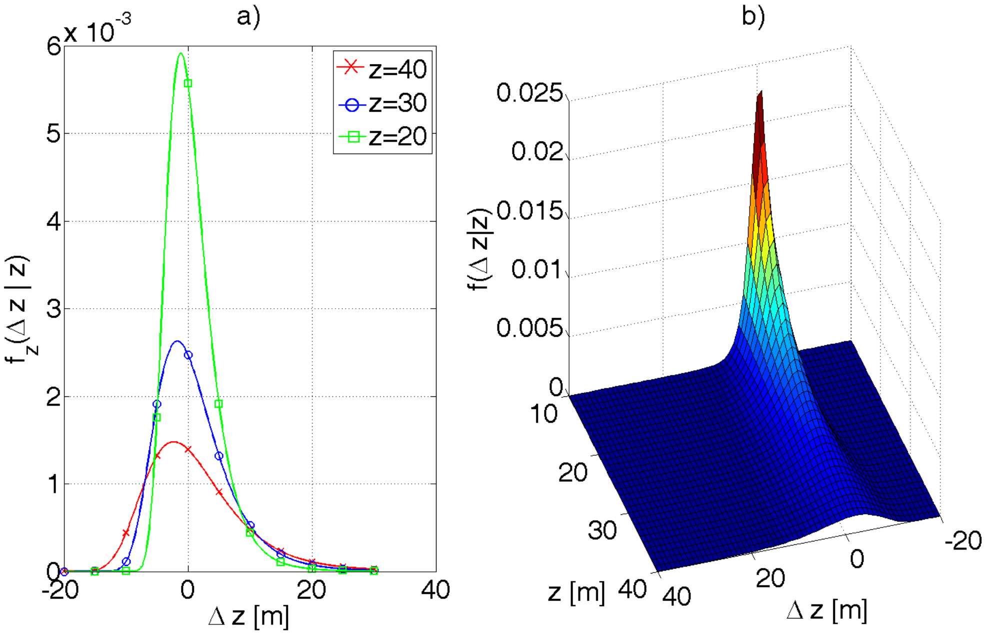
$$\Delta z \approx -\frac{Bf}{x_d^2}\Delta x_d \approx -\frac{z^2}{Bf}\Delta x_d \tag{32}$$

**Figure 10.** An illustration of the quantization effect on estimating the $z$ coordinate. (**a**) The plot of $z$ and errors thereof as a function of disparity $x_d$. Dots symbolize the depth $z$ calculated from $z = Bf/x_d$. Bars symbolizes errors of $z$ when $x_d$ is encumbered with an error of $\pm\frac{\epsilon}{2}$ or $\pm\frac{\text{pixel}}{2}$, (**b**) The absolute error of $z$ as a function of the depth $z$.



The error of determining the disparity does not only come from quantizing $x_d$, in which case, the disparity maps presented in Section 3.8 would be devoid of any artifacts or falsely calculated disparities—see [36] for full clarification.

**Figure 11.** (**a**) A plot of $f_z(\Delta z|z)$ for different values of $z$. As $z$ increases the conditional probability density function widens. (**b**) A 3D plot of $f_z$. For illustration purposes, $\sigma_d$ was assumed 6 pix, whereby, to account for the units, $f$ = 279 pix was substituted.



The camera used in this work calculates disparity by correlating small blocks (e.g., 7 × 7 pixels) and looking for best match. The probability density function of the disparity error is disputable in this case. Let it be the Gaussian distribution, expressed by Equation (33), where $\Delta x_d$ denotes the disparity error.

$$f_d(\Delta x_d) = \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-\frac{\Delta x_d^2}{2\sigma_d^2}\right) \tag{33}$$

The probability density function of $\Delta z$ can be calculated from Equation (34). The function $f_z$ is conditioned on $z$, *i.e.*, the error $\Delta z$ depends on the value of $z$ as mentioned before.

$$f_z(\Delta z|z) = \left|\frac{\partial(\Delta x_d)}{\partial(\Delta z)}\right| f_d(x_d) \tag{34}$$

Equation (32) holds for small $x_d$. The exact value of $\Delta z$ can be computed from Equation (35). The formula necessary for the transformation is given by Equation (36).

$$\Delta z = \frac{Bf}{x_d + \Delta x_d} - z \tag{35}$$

$$\frac{\partial(\Delta x_d)}{\partial(\Delta z)} = -\frac{Bf}{(\Delta z + z)^2} \tag{36}$$

Thus, $f_z$ can be calculated from Equation (37).

$$f_z(\Delta z|z) = \frac{Bf}{(\Delta z + z)^2} \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-\frac{(Bf\Delta z)^2}{2\sigma_d^2(z + \Delta z)^2}\right) \tag{37}$$

The shape of the function $f_z(\Delta z|z)$ for different values of $z$ is presented in Figure 11. The function is skewed as one should have suspected and decays slower for $\Delta z$ greater than 0.

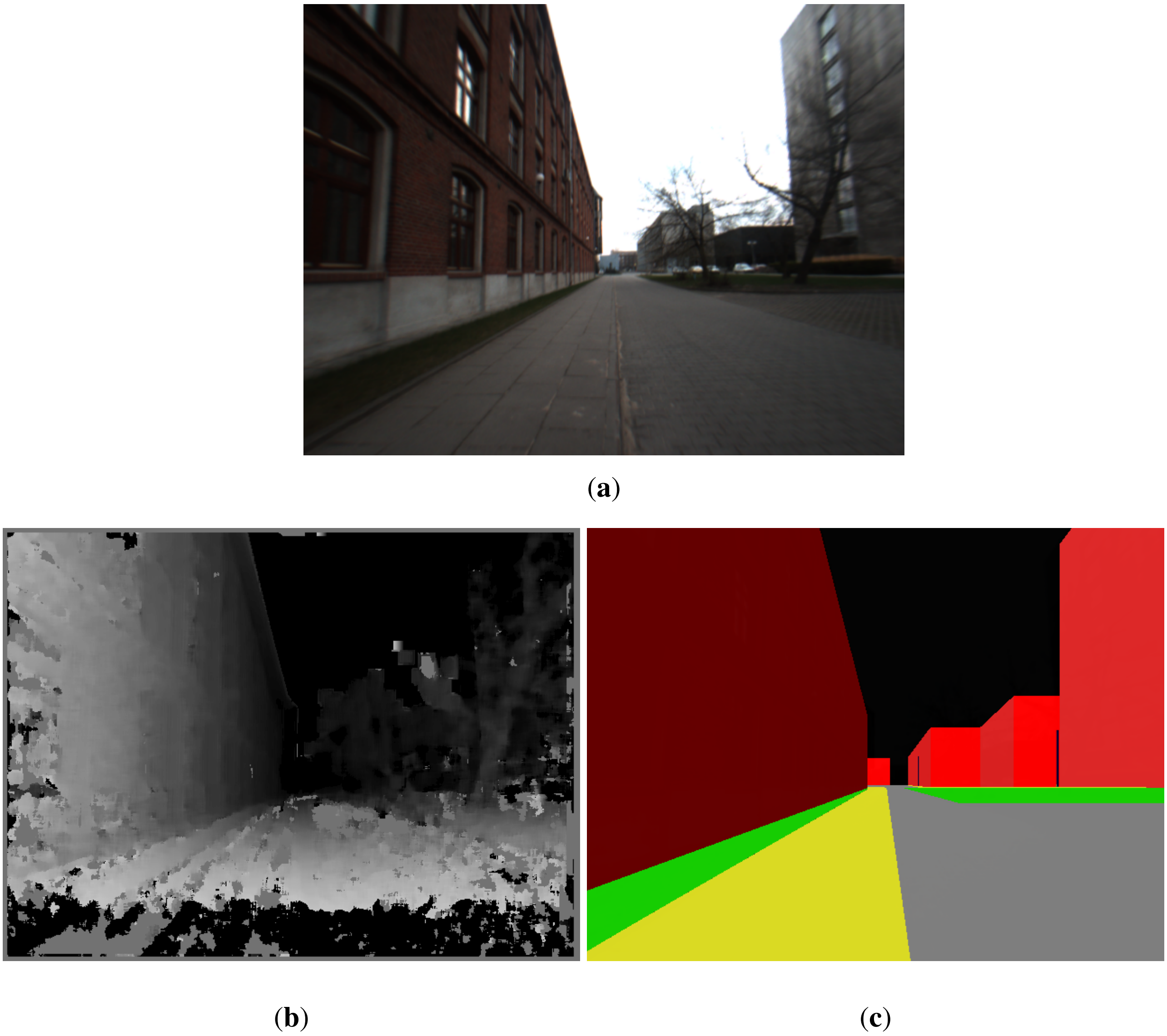
### *3.7. 3D Map of an Urban Terrain*

A precise 3D map of an urban terrain was provided by the local cartographic office of Lodz (MODGiK). The map is divided in layers containing: roads, buildings, entrances to buildings, steps, fences, lawns, trees, bushes, ponds, lamp posts and many other data not used in the project. The accuracy of building deployment is in an order of single centimetres and the buildings' heights were measured with a bit worse accuracy. The radius of trees' crowns was assumed 3 m and the radius of lamp posts 0.5 m.

### *3.8. 3D Matching Algorithm*

Estimated location based on the stepmeter, gyroscope and digital maps is accurate to within 15 m (a street with pavements on both sides) [3]. GPS errors in urban canyons being a several dozens of metres, practically both sides of a road are equally probable.

**Figure 12.** (**a**) TV camera image. (**b**) Disparity map from the stereo camera. Darker colours correspond to smaller disparity, thus bigger distance. (**c**) 3D model of the corresponding urban environment.



(**a**)

(**b**)

(**c**)

Figures 12(a–c) show a picture from a monocamera, a disparity map $\mathbf{K}$ and a part of the 3D model of the University Campus.

Let us use the local coordinate system defined in Section 3.1.2. Let the vector $[x'_S, y'_S, z'_S, \phi'_S]^T$ denote the Cartesian coordinates and the orientation of the stereo camera in the local coordinate system. $\phi'_S$ is called the azimuth of the stereo camera and it has the same interpretation as with a magnetic compass. It is assumed that the stereovision camera is worn by a pedestrian and it is attached to the chest, say at $z = h = 1.5$ m above the ground. The optical axis of the camera is directed roughly parallel to the ground plane. The positioning algorithm has a precise 3D model of the environment that is defined for the local coordinate system—see Figure 12(c). Let us put a virtual camera in the $(x'_V, y'_V, z'_V)$ coordinates and set its orientation to $\phi'_V$. Then, let us render the virtual environment based on the 3D model. The virtual camera should see the picture presented in Figure 12(c). On the other hand, the stereovision camera provides a depth image of the environment. Both images can be compared. Let $\mathbf{L_S}$ and $\mathbf{L_V}$ denote the depth image returned by the stereovision camera and retrieved from the virtual environment, correspondingly. $f_L(\mathbf{L_S}, \mathbf{L_V})$ is some cost function comparing the two depth images and returning comparison error. The function is defined by Equation (38).

$$f_L : (\mathbf{L}, \mathbf{L}) \to \mathbb{R}^1 \tag{38}$$

Formally, the problem comes down to finding the vector $[x'_V, y'_V, z'_V, \phi'_V]^T$ where the function $f_L(\mathbf{L_S}, \mathbf{L_V})$ returns a minimum value, that is, the difference $||\mathbf{L_S} - \mathbf{L_V}||$ is minimal according to some criterion. Then the estimated coordinates of the stereo camera (that is the pedestrian location) are given by Equation (39). This task is handled by the particle filtering algorithm.

$$[x'_S, y'_S, z'_S, \phi_S]^T = [x'_V, y'_V, z'_V, \phi'_V]^T \tag{39}$$

Assuming $z_S = z_V = h$ reduces the dimension of the problem. The function $f_L$ is still not trivial. The depth picture $\mathbf{L_S}$ is encumbered with many errors, see Figure 12(b). For example, sun illumination causes some parts of the depth map to have falsely determined values (see Figure 13). The function $f_L$ should be immune to these undesirable effects that can cause random bias errors, most troublesome errors to correct by any filter, including a particle filter which is used in the presented project. A random bias error is understood as an error whose expected value differs considerably from 0 for a significant period of time, e.g., 30 seconds. There can be also objects that can compromise the comparison between $\mathbf{L_S}$ and $\mathbf{L_V}$ and cannot be filtered by any methods, *i.e.*, parked cars, trucks, bushes, posters, garbage bins, *etc*. Their influence can be minimized to some extent by excluding from comparison that part of $\mathbf{L_S}$ which is below the optical axis of the camera. Hence the ground surface is also excluded from comparison. First $\mathbf{L_S}$ is analysed to seek fragments for comparison. The method of analysing $\mathbf{L_S}$ depends on many empirical factors that were verified during many trials. The idea is following:

1. Reject the part of the depth map $\mathbf{L_S}$ that is below the optical axis of the camera.
2. Divide vertically the upper part of $\mathbf{L_S}$ into $n_S = 5$ equal regions $\mathbf{S}_j$, $j = 1 \ldots n_S$—see Figure 13(a). This forces to compare the depth maps $L_S$ and $L_V$ in different regions, which, to some extend, decorrelate random bias errors.
3. for $j = 1$ to $n_S$

(a) From the area $\mathbf{S}_j$, pick at random a small square window $\mathbf{W}_j$ of $W_W$ by $W_W$ measurements, where $W_W = 24$ pix.

(b) Check if $\mathbf{W}_j$ is devoid of artifacts like sun illumination, falsely determined disparity *etc.*—*c.f.* Figure 13(b).

(c) Calculate the angle $\alpha_j$ at which the camera sees the object pointed by $\mathbf{W}_j$ (compare with the angle $\alpha$ in Figure 9).

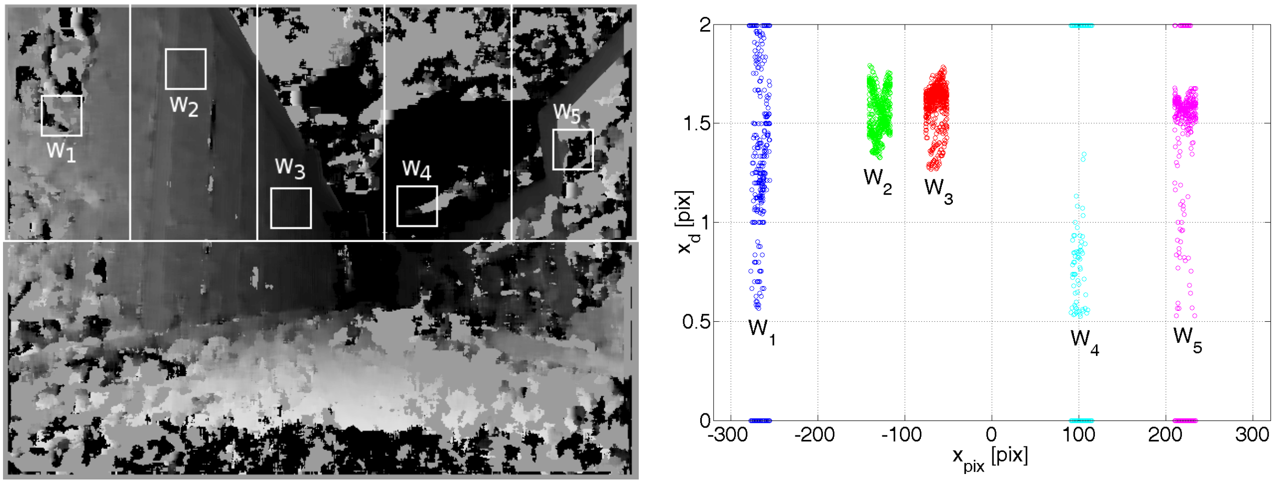
(d) Calculate the distance $d_{Sj}$ to an object encompassed by $\mathbf{W}_j$.

(e) Check what the virtual camera sees at the angle $\alpha_j$ from coordinates $[x'_V, y'_V, z'_V, \phi'_V]^T$. Calculate in this direction the distance to the nearest object $d_{Vj}$.

(f) Compare the distances $d_{Sj}$ and $d_{Vj}$ by the distance error function $f_{Lj}$.

4. Return the comparison error between $\mathbf{L_S}$ and $\mathbf{L_V}$ as $f_L(\mathbf{L_S}, \mathbf{L_V}) = f_{L1} \cdot f_{L2} \cdot f_{L3} \cdot f_{L4} \cdot f_{L5}$.

**Figure 13.** (**a**) An example of a disparity map $\mathbf{K}$. Darker colours correspond to smaller values of disparity, thus larger distances. Example windows for each region $\mathbf{S}_j$ $(j = 1 \ldots 5)$ are denoted by $\mathbf{W}_1$ through $\mathbf{W}_5$. One can see that for windows $\mathbf{W}_2$ and $\mathbf{W}_3$ the depth map was determined correctly and these windows should be compared with corresponding windows in the $\mathbf{L}_V$ depth map. (**b**) The disparity values $x_d$ for each window $\mathbf{W}_j$ as a function of the $x_{\text{pix}}$ coordinate in the disparity picture. For $\mathbf{W}_1$, $\mathbf{W}_4$ and $\mathbf{W}_5$ windows, the disparity $x_d$ changes from 0 pix to ≈2 pix which corresponds to distances ∞ to ≈16 m. These windows should be rejected from further analysis.



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

**Ad. 1, 2.** The areas $\mathbf{S}_j$ are shown in Figure 13(a).

**Ad. 3a.** The window $\mathbf{W}_j$ is picked at random from the corresponding area $\mathbf{S}_j$, according to Figure 13(a).

**Ad. 3b.** By trial and error, the empirical measure (40) was proposed to check if the disparity map for $\mathbf{W}_j$ is devoid of artifacts. $x_d^{\max}$ (41) and $x_d^{\min}$ (42) are the maximum and minimum values of disparities for a given window $\mathbf{W}_j$

$$k_j = \frac{x_d^{\max}(\mathbf{W}_j) - x_d^{\min}(\mathbf{W}_j)}{x_d^{\max}(\mathbf{W}_j)} \tag{40}$$

$$x_d^{\max}(\mathbf{W}_j) = \max(\mathbf{K}(x_{\text{pix}}, y_{\text{pix}})); \forall(x_{\text{pix}}, y_{\text{pix}}) \in \mathbf{W}_j \tag{41}$$

$$x_d^{\min}(\mathbf{W}_j) = \min(\mathbf{K}(x_{\text{pix}}, y_{\text{pix}})); \forall(x_{\text{pix}}, y_{\text{pix}}) \in \mathbf{W}_j \tag{42}$$

Window $\mathbf{W}_j$ is stated to be devoid of artifacts if $k_j > k_{\text{thr}}$, where an empirical value of $k_{\text{thr}} \approx 0.8$.

**Ad. 3c.** The angle $\alpha_j$ at which the stereovision camera sees the object encompassed by the $\mathbf{W}_j$ window can be calculated from Equation (43), where $x^{(c)}$ denotes the $x$ coordinate of the centre of $\mathbf{W}_j$.

$$\alpha_j = \arctan \frac{x_j^{(c)}}{f_{\text{pix}}} \tag{43}$$

**Ad. 3d.** The distance $d_{Sj}$ to the object pointed by $\mathbf{W}_j$ can be calculated from Equation (44). This is an arithmetical average of depths divided by the $\cos \alpha_j$ factor to obtain the Euclidean distance—*c.f.*, Equations (26) and (27) that show the difference between the depth of a point and distance to the camera's $Y$ axis.

$$d_{Sj}(\alpha_j) = \frac{1}{\cos \alpha_j} \frac{1}{W_W^2} \sum_{x_{\text{pix}}} \sum_{y_{\text{pix}}} \mathbf{L_S}(x_{\text{pix}}, y_{\text{pix}});\ (x_{\text{pix}}, y_{\text{pix}}) \in \mathbf{W}_j \tag{44}$$

**Ad. 3e.** Given the coordinates and orientation of a virtual camera in the 3D environment, the $d_{Vj}$ distance from the virtual camera to the nearest object at $\alpha_j$ angle can be easily obtained.

**Ad. 3f.** The corresponding distances $d_{Sj}$ and $d_{Vj}$ are compared by the error function $f_{Lj}$. For the integrity sake of Equation (47), the function $f_{Lj}$ should return 1 if window $\mathbf{W}_j$ was rejected from comparison, *i.e.*, $k_j \leq k_{\text{thr}}$. The function $f_{Lj}$ reflects the correspondence between $d_{Sj}$ and $d_{Vj}$, whereby the comparison should be more tolerant for greater distances, according to formula (32)—the accuracy of determining $d_{Sj}$ drops with its square value. Following the reasoning in the previous chapter, the function $f_{Lj}$, expressed by Equation (45), was proposed. The parameter $a$ introduces some tolerance in case of unexpected objects registered by the stereo camera (e.g., cars, humans), *c.f.* Figure 14. Experimentally, $a \in< 0.95, 0.98 >$ gives desired results.

$$f_{Lj}(k_j, d_{Sj}, d_{Vj}) = \begin{cases} 1 & k_j \leq k_{\text{thr}} \\ a + (1-a)\exp\left(-\frac{(Bf\Delta z)^2}{2\sigma_L^2(d_{Lj}+\Delta z)^2}\right) & k_j > k_{\text{thr}} \end{cases} \tag{45}$$

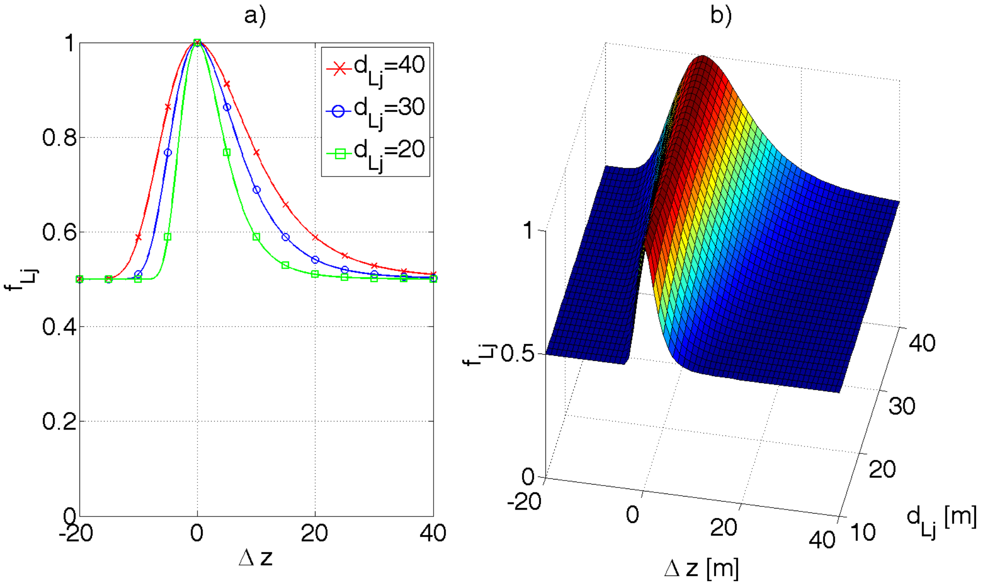
$$\Delta z = d_{Sj} - d_{Lj} \tag{46}$$

**Ad. 4.** Finally, the function $f_L$ comparing the two environments, *i.e.*, real and virtual, can be defined by Equation (47).

$$f_L(\mathbf{L_S}, \mathbf{L_V}) = \prod_{j=1}^{n_S} f_{Lj}(k_j, d_{Sj}, d_{Vj}) \tag{47}$$

The 3D match algorithm is evoked 10 times a second.

**Figure 14.** (**a**) An example of the $f_{Lj}$ function from Equation (45) where $k_j > k_{\text{thr}}$. Example values of $a = 0.5$ and $\sigma_L$ = 6 pix were used (to account for the units, $f$ = 279pix was substituted). (**b**) A 3D plot of $f_{Lj}$ function where $k_j > k_{\text{thr}}$ and $\Delta z = d_{Sj} - d_{Lj}$. The base of the function widens as the compared distances increase.



### 3.9. *Particle Filtering*

#### 3.9.1. Introduction

The particle filter is a sequential version of the Monte Carlo method [37] which enables to find solutions of a multidimensional problem by generating a large number of possible solutions, so-called particles, and then verifying these solutions by a given criterion, *i.e.*, an error function. The particle filter can be looked at from the statistical point of view, whereby the solution is represented by a probability density function. The particle filter improves upon Kalman filter in case of multi-modal distributions and non-linear dynamics at the cost of high computation burden. Let $i$ denote the number of a particle and $L$ the total number of particles, thus $i \in< 1, L >$. A particle at a given time instant $k$ is represented by vector $\mathbf{c}_i(k)$ given by Equation (48).

$$\mathbf{c}_i(k) = \begin{bmatrix} \mathbf{x}_i(k) \\ w_i(k) \end{bmatrix} \tag{48}$$

where $\mathbf{x}_i(k)$ is a possible system state at time instant $k$ and $w_i(k)$ is the weight associated with $i$-th particle. A possible system state is understood as a possible solution of the problem in question. In our case $\mathbf{x}_i(k)$ represents a hypothetical user location and orientation, $\mathbf{x}_i(k) = [x_i, y_i, \phi_i]^T$, $x_i, y_i$ denote coordinates in the local coordinate system defined in Section 3.1.2. $\phi_i$ is the azimuth orientation. The aim of this simulation is to to find the most probable user location. Given the series of measurements $\mathbf{y}(1)\dots\mathbf{y}(k)$ the pedestrian's location is given by the probability density function that can be approximated by the probability mass function (49).

$$p(\mathbf{x}(k)|\mathbf{y}(1:k)) \approx \sum_{i=1}^{L} \delta(\mathbf{x}-\mathbf{x_i}(k)) \cdot w_i(k) \tag{49}$$

where $\delta(\cdot)$ is the Dirac delta function. Weights $w_i(k)$ of all generated particles should sum up to unity.

### 3.9.2. Implementation of the Particle Filtering Algorithm

The implementation of the particle filtering algorithm for processing the positioning data is as follows:

1. **Initialization**. At the algorithm outset for $k$=1, all the particle states $\mathbf{x_i}(1)$ are randomly initialized according to a given distribution. The weights $w_i(1)$ are assigned equal values of $\frac{1}{L}$.
2. **Prediction**. Based on the transition Equation (50), new particle states are predicted.

   $$\mathbf{x}_i(k) = \mathbf{f}(\mathbf{x}_i(k-1), \mathbf{u}(k-1), \mathbf{v}_i(k-1)) \tag{50}$$

   $\mathbf{u}(k-1)$ is a driving vector, $\mathbf{v}_i(k-1)$ is a noise vector introduced to the state due to the error of $\mathbf{u}(k-1)$, where each particle is perturbed with an individually generated vector $\mathbf{v}_i(k-1)$, $\mathbf{f}(\cdot)$ is a transition function that calculates a new state based on the previous one, driving signals and errors thereof.
3. **Measurement update**. Each measurement $\mathbf{y}(k)$ updates the weights of the particles by Equation (51).

   $$w_i(k) = w_i(k-1) \cdot p(\mathbf{y}(k)|\mathbf{x}_i(k)) \tag{51}$$

   $p(\mathbf{y}(k)|\mathbf{x}_i(k)$ is a conditional probability density of measuring $\mathbf{y}(k)$, given the particle state $\mathbf{x}_i(k)$. In other words, $p(\mathbf{y}(k)|\mathbf{x}_i(k)$ characterizes error distribution of a given data source. Particles that diverge in the long run from measurements will assume small weights $w_i(k)$.
4. **Weights' normalization**. The weights are normalized so that they sum up to 1—Equation (52).

   $$w_i(k) := \frac{w_i(k)}{\sum\limits_{i=1}^{L} w_i(k)} \tag{52}$$

5. **State estimation**. The most probable system state is estimated, e.g., by the first moment (53):

   $$\overline{\mathbf{x}}(k) = \sum_{i=1}^{L} \mathbf{x}_i(k) \cdot w_i(k) \tag{53}$$

6. **Resampling**. After a number of algorithm iterations, all but a few particles have negligible weights and therefore do not participate in the simulation effectively. This situation is detected by

calculating the so-called degeneration indicator, expressed by Equation (54), which is an inverse of the second moment reflecting the dispersion of the weights.

$$d(k) = \frac{1}{L\sum\limits_{i=1}^{L} w_i^2(k)} \tag{54}$$

If all particles have the same weights of $L^{-1}$ then $d(k)$ reaches its maximum value of 1. As the weights start to differ, the $d(k)$ indicator decreases. If $d(k)$ falls below a given threshold, then a process called resampling is introduced and a new set of particles is created. The probability of copying a particle to the new set is proportional to its weight $w_i(k)$. Therefore particles that poorly approximate the system state are superseded by more "accurate" particles. Then all particles are assigned the same weight $L^{-1}$.

7. **Go to point 2**.

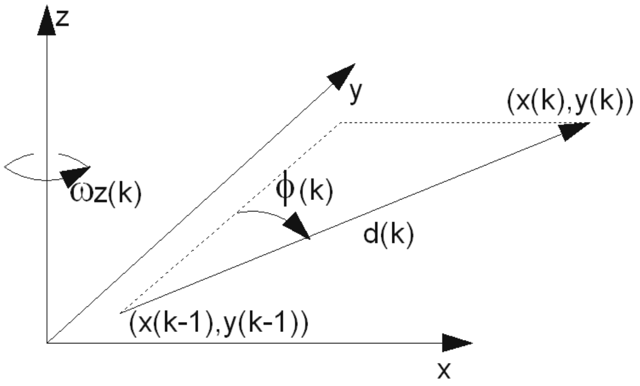
### 3.9.3. Implementation

**Ad. 1.** The particles can be initialized, e.g., around a first GPS readout.

**Ad. 2.** Based on Figure 15, the transition equation for a particle $i$ reads (55).

$$\begin{bmatrix} x_i(k) \\ y_i(k) \\ \phi_i(k) \end{bmatrix} = \begin{bmatrix} x_i(k-1) \\ y_i(k-1) \\ \phi_i(k-1) \end{bmatrix} + \begin{bmatrix} (d(k)+v_i(k))\sin(\phi_i(k-1)-(\omega_z(k)+w_i(k))dT) \\ (d(k)+v_i(k))\cos(\phi_i(k-1)-(\omega_z(k)+w_i(k))dT) \\ -(\omega_z(k)+w_i(k))dT \end{bmatrix} \tag{55}$$

**Figure 15.** An explanation of the unperturbed state transition: $d(k)$ is the length of a step estimated by the pedometer. $\phi(k)$ is the azimuth orientation. $\omega_z(k)$ is the angular velocity around the gravity axis. $\omega_z$ is estimated by the gyroscope and the gravity estimation algorithm. $(x(k-1), y(k-1))$, $(x(k), y(k))$ are the coordinates in the previous and current time instant.



$v_i(k)$ and $w_i(k)$ are random realizations of the pedometer and gyroscope noise respectively. Each particle is perturbed individually. $k$ is the time instant, where $t = k \cdot dT$.

**Ad. 3.** The following sources are used to update the particle weights:

- GPS. The measurement update for the GPS readouts are given by Equations (56) through (58) based on the Equation (7)

$$w_i(k) = w_i(k-1) \cdot p_{\mathrm{GPS}}(r_i, \sigma_{\mathrm{GPS}}) \tag{56}$$

$$r_i = \sqrt{((x_{\mathrm{GPS}}(k) - x_i(k))^2 + (y_{\mathrm{GPS}}(k) - y_i(k))^2)} \tag{57}$$

$$\sigma_{\mathrm{GPS}} = \beta_{\mathrm{GPS}}\mathrm{HDOP} \tag{58}$$

$\beta_{\mathrm{GPS}}$ is a coefficient which enables to weight GPS readouts with appropriate importance. Larger values of $\beta_{\mathrm{GPS}}$ makes the $p_{\mathrm{GPS}}()$ function more tolerant to GPS errors.

$$w_i(k) = w_i(k-1) \cdot w_{\mathrm{map}}(x_i(k), y_i(k)) \tag{59}$$

- comparison between the 3D virtual environment model and depth picture retrieved from the stereo camera. The measurement update equation is given by Equation (60).

$$w_i(k) = w_i(k-1) \cdot f_{L,i}(\mathbf{L_S}, \mathbf{L_{V,i}}) \tag{60}$$

For each particle, a virtual environment is generated individually. The coordinates of the virtual camera are set to $[x_i, y_i, h, \phi_i]$. $h$ is the height at which the stereo camera is attached. Then the depth map $L_{V,i}$ is generated. The function $f_L(\mathbf{L_S}, \mathbf{L_{V,i}})$ is evoked for every particle. This is a computation demanding procedure. Note that, if an unexpected object appears in front of the stereo camera for 10 seconds, given particles' weights will be decreased by $a^{10/s \cdot 10s}$, *c.f.* Equation (45) (3D matching algorithm is run 10 times a second). This might introduce a large error in the position estimation. Bigger values of $a$ diminish the benefit of using a stereo camera and 3D model of the urban terrain.

## 4. Trials

The trials were carried out at the campus of Lodz University of Technology, in an area of *ca.* 500 m by 500 m. The length of the trial path was *ca.* 2.6 km. The reference path was restored off-line by analysing images recorded by the camera. The average error of determining the reference path was *ca.* 0.5 m on average. The maximum error might have been 2–3 m. The delineated path is composed of straight lines. The location error is calculated as the distance between the appropriate path segment and the estimated location by a GPS receiver or the particle filter. Thus, actual positioning errors are larger. On the other hand, GPS errors assume dominant values in the direction perpendicular to a street. In the direction parallel to a street there are no buildings to occlude the sky. Hence, the signal from GPS satellites is not occluded, which results in a small error in the direction along a street [2].

## 5. Results

The results are presented by plots of paths, a histogram and a cumulated histogram of errors and a table that summarizes the errors. An interesting question is how much a given data source improves the accuracy of the positioning. Table 1 provides the answer. Therefore, several off-line tests were run with different combination of data sources. It was though assumed that the starting location is known.

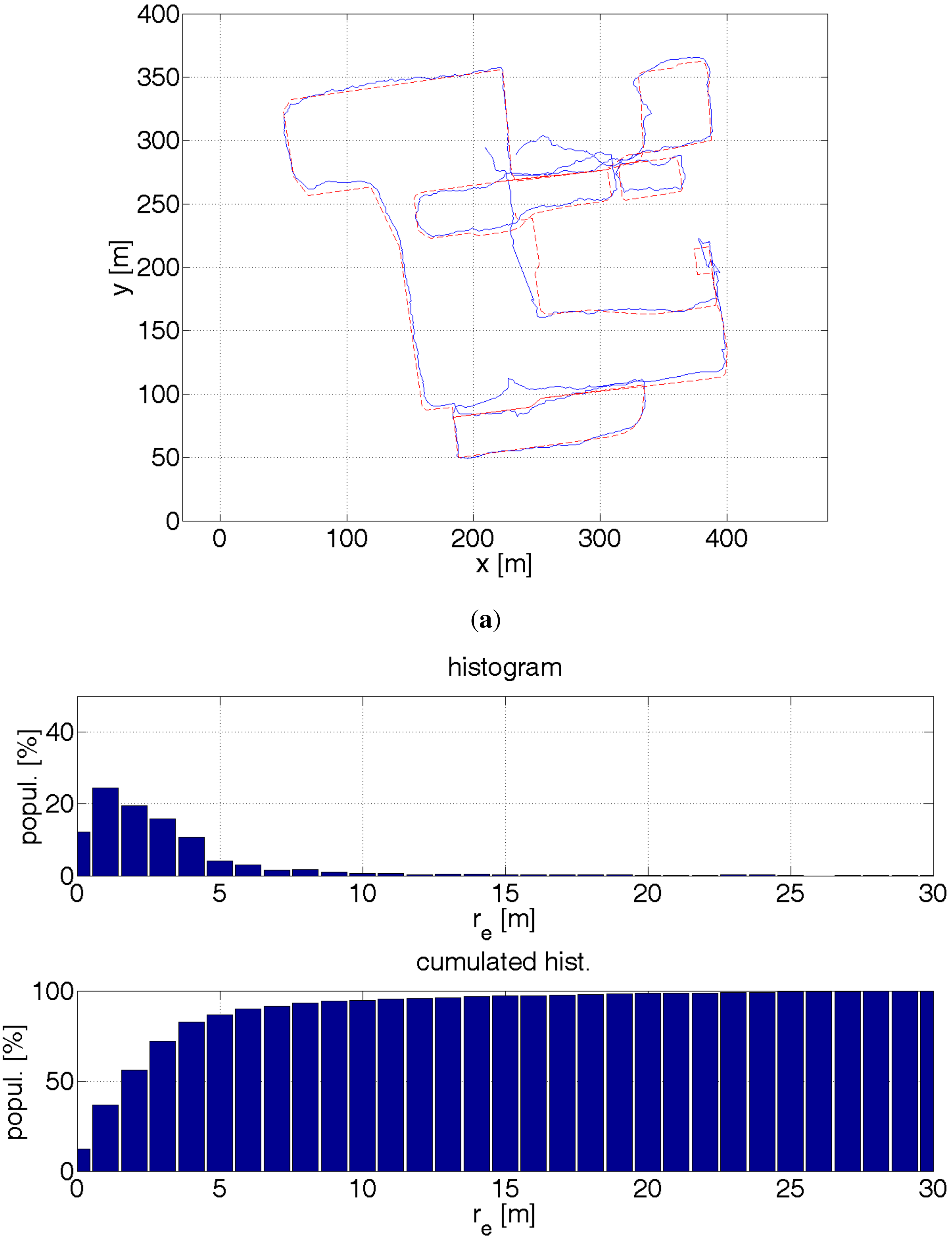
Otherwise, a bigger number of particles would have been necessary to find a first approximate location. Once, a rough location is found the filter is convergent. The initial errors might have been large and therefore eclipse the key point of measurements. On one side, this assumption might be justified. A pedestrian, e.g., a blind user goes out from his home, so he knows his location. On the other hand, a blind pedestrian might use the system when lost. The aforementioned distance error $r_e$ is used as a comparison criterion. The results are evaluated by the following measures:

1. Root-mean-square value, denoted by RMS$_e$,
2. Mean error, denoted by $\mu_e$,
3. CEP50—Circle of Error Probability 50%—the radius of a circle encompassing 50% of errors, *i.e.*, a median value,
4. CEP90—the radius of a circle encompassing 90% of errors, *i.e.*, ninth decile,
5. CEP95—the radius of a circle encompassing 95% of errors, *i.e.*, 95th percentile,
6. CEP99—the radius of a circle encompassing 99% of errors, *i.e.*, 99th percentile,
7. Maximal error—the maximum value of an error recorded during a trial, denoted by $\max(r_e)$

**Table 1.** Results of the trial at the University Campus. Seven setups of data sources were simulated. The sign '+' in a column denotes that a given data source was included in the simulation. Otherwise the data source is not used. Setup number 4 was carried out with all data sources on. The results for this setup are best.

| Setup | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---:|:-:|:-:|:-:|:-:|:-:|:-:|:-:|
| GPS | + | + | + | + | | | |
| Inertialsensors | | + | + | + | + | + | + |
| Probability map | | | + | + | + | + | |
| 3D matching algorithm | | | | + | + | | |
| RMS$_e$ [m] | 5.23 | 4.96 | 2.10 | 1.26 | 1.65 | 4.39 | 107.5 |
| $\mu_e$ [m] | 3.27 | 3.45 | 1.56 | 0.87 | 0.95 | 2.94 | 86.4 |
| CEP50 [m] | 2.19 | 2.47 | 1.24 | 0.57 | 0.55 | 1.89 | 76.4 |
| CEP90 [m] | 6.54 | 6.79 | 3.39 | 2.03 | 2.13 | 6.92 | 182 |
| CEP95 [m] | 10.7 | 12.0 | 4.28 | 2.90 | 3.19 | 9.09 | 210 |
| CEP99 [m] | 23.3 | 18.2 | 6.41 | 4.50 | 7.02 | 16.0 | 261 |
| $\max(r_e)$ [m] | 30.6 | 20.6 | 8.64 | 5.24 | 12.8 | 25.5 | 265 |
| Figure | 16 | 17 | 18 | 19 | 20 | 21 | 22 |

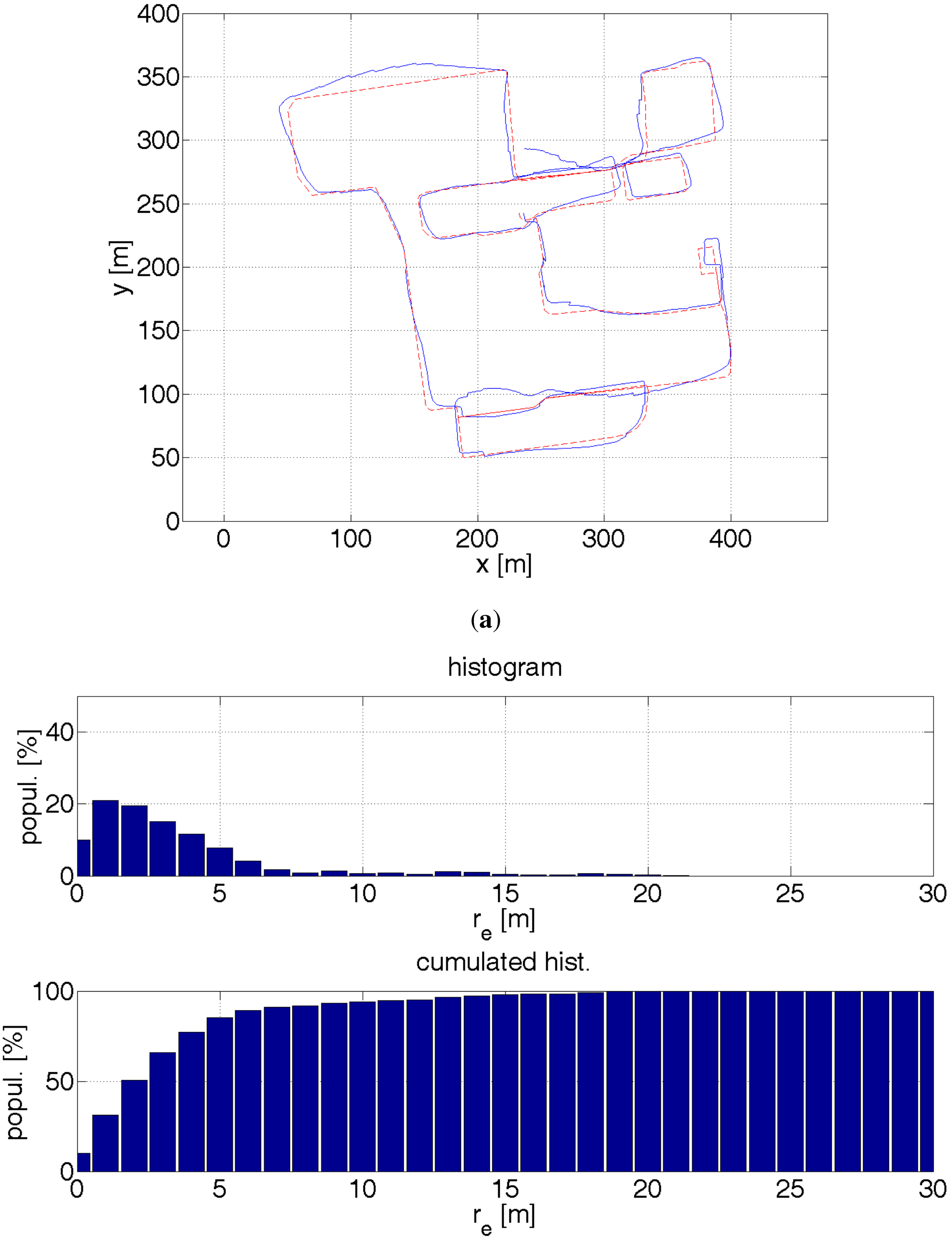
**Figure 16.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of a GPS receiver. In the presence of buildings, the GPS readouts are ≈30 $\mathrm{m}$ inaccurate. The errors depends also on the satellites configuration against a building. This can be noticed for coordinates $(220\ \mathrm{m}, 80\ \mathrm{m})$. The same place was revisited after a couple of minutes and the positioning error was very different. (**b**) Histograms of distance errors.



(**a**)

(**b**)

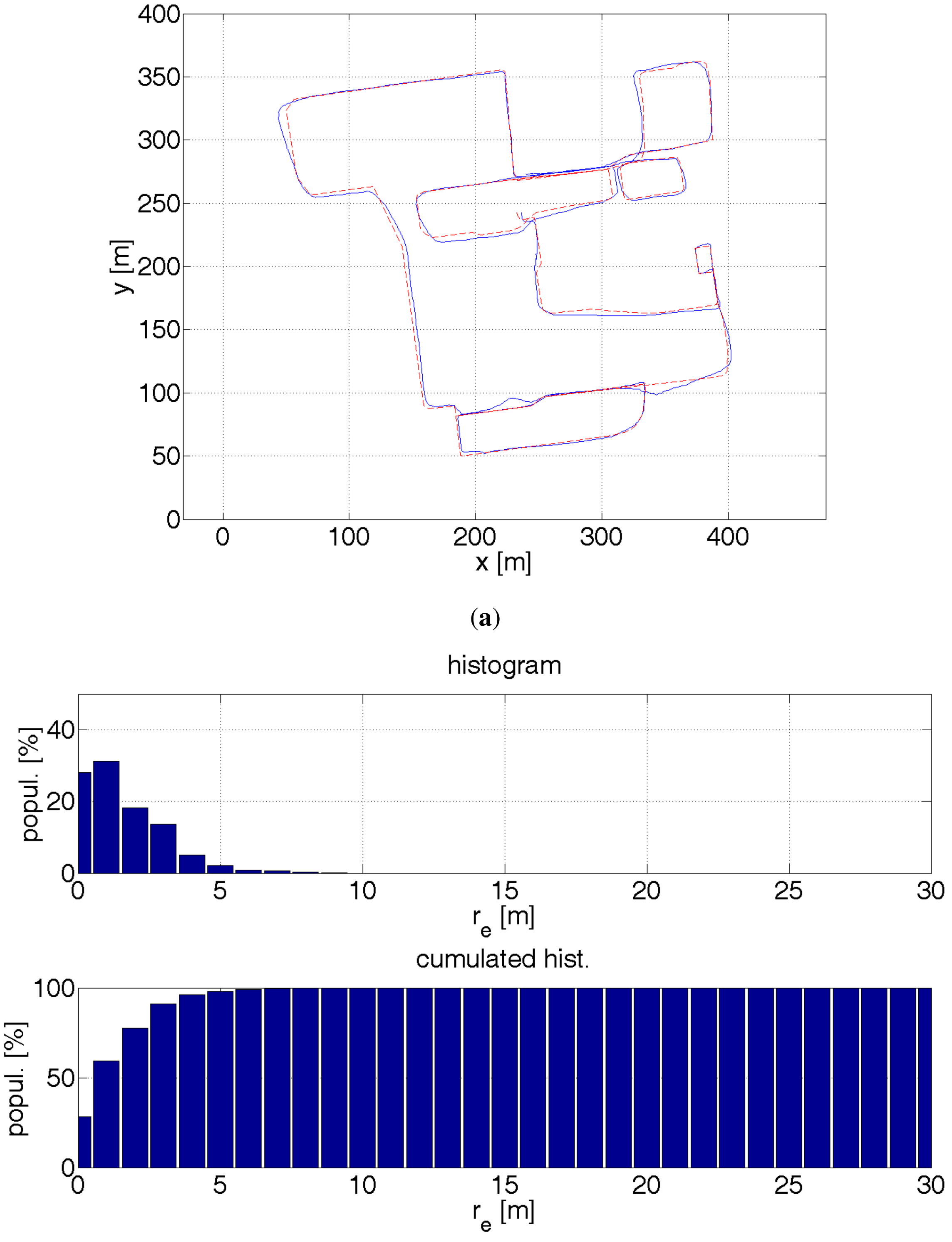
**Figure 17.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of GPS receiver and inertial sensors. The readouts from the inertial sensors enable to eliminate big errors. The accuracy improvement is not significant due to errors in estimating direction. (**b**) Histograms of distance errors.



(**a**)

(**b**)

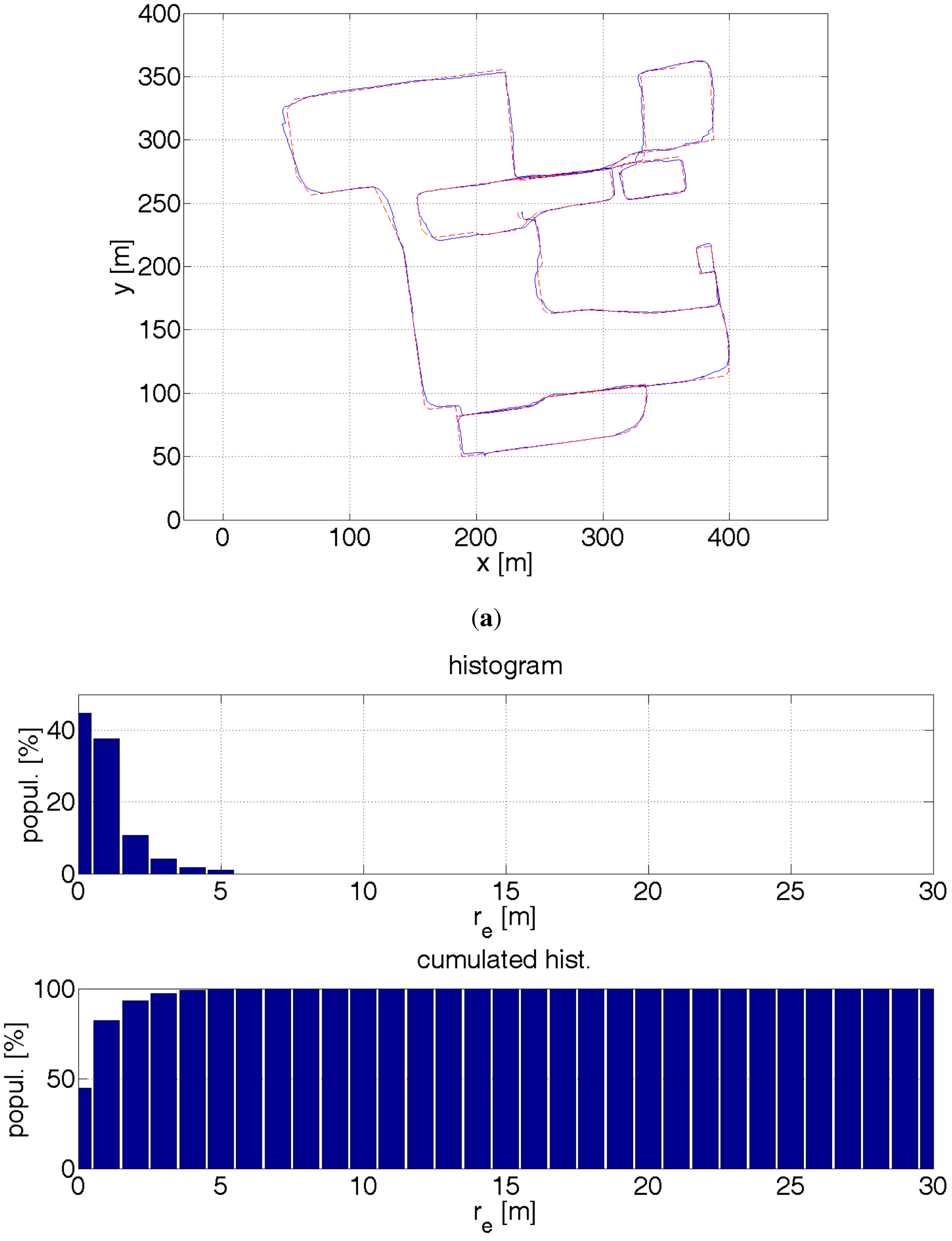
**Figure 18.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of GPS receiver, inertial sensors, probability map. The probability map eliminates direction errors, by pruning particles that diverge from a reference direction. Therefore errors are instantly eliminated and do not propagate. The positioning errors are then described by random walk with bounds. (**b**) Histograms of distance errors.



(**a**)

(**b**)

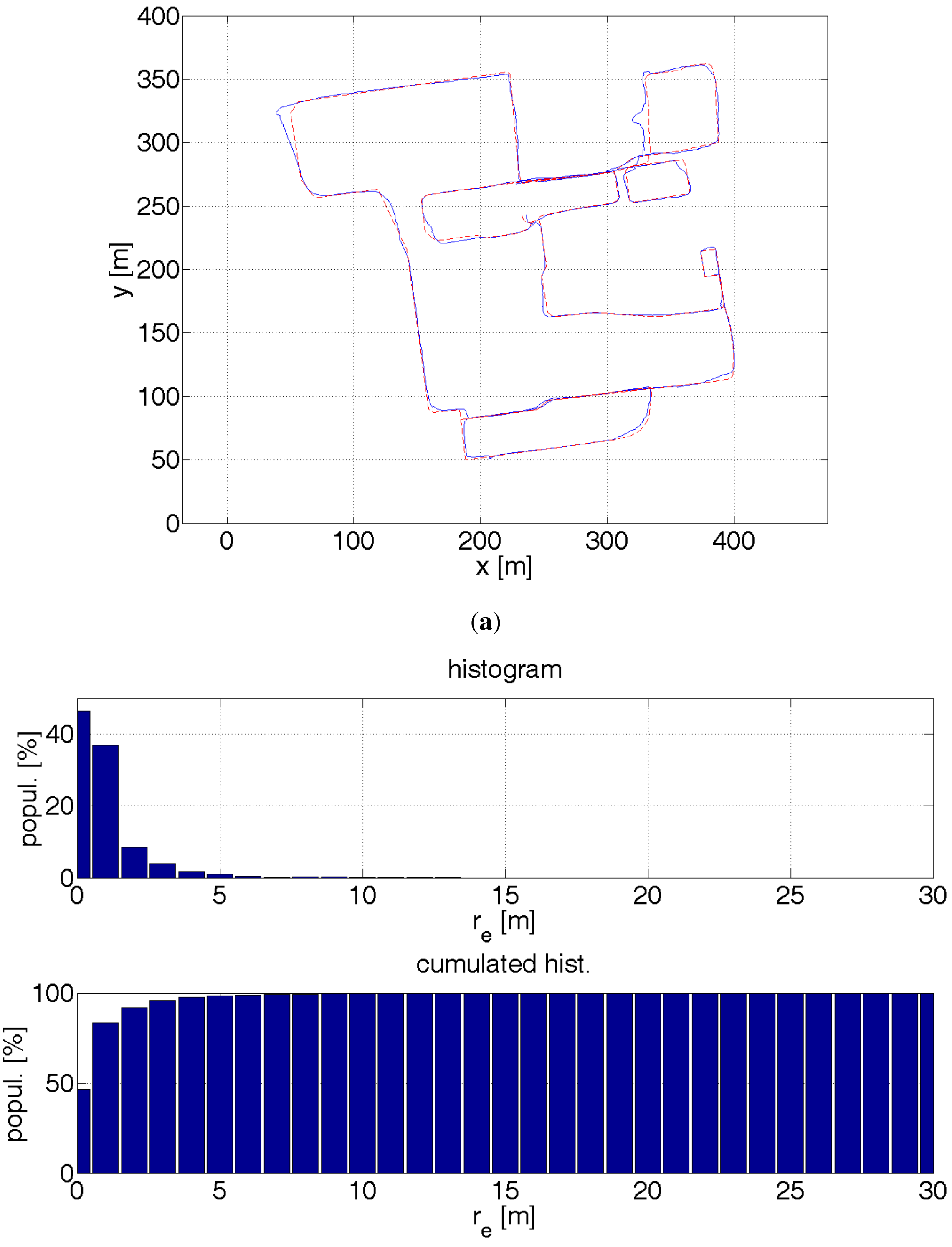
**Figure 19.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of GPS receiver, inertial sensors, probability map, stereo camera and 3D model of the environment. The positioning is quite precise. The errors are mainly introduced by the error in estimating the heading direction. (**b**) Histograms of distance errors.



(**a**)

(**b**)

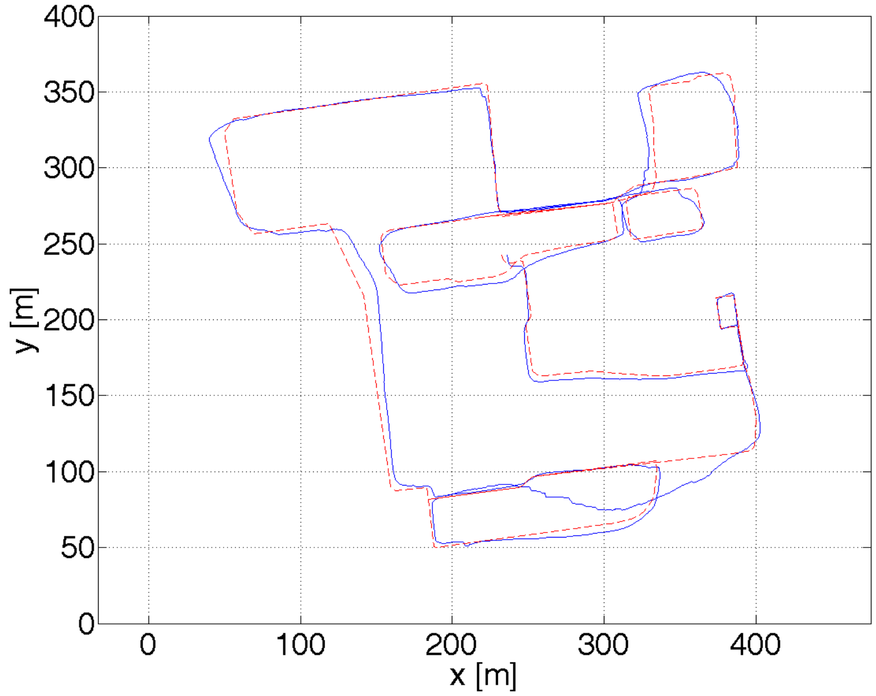
**Figure 20.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of inertial sensors, probability map, stereo camera and 3D model of the environment. When the initial position is known, the filter can estimate the pedestrian location with smaller errors than the GPS receiver. (**b**) Histograms of distance errors.



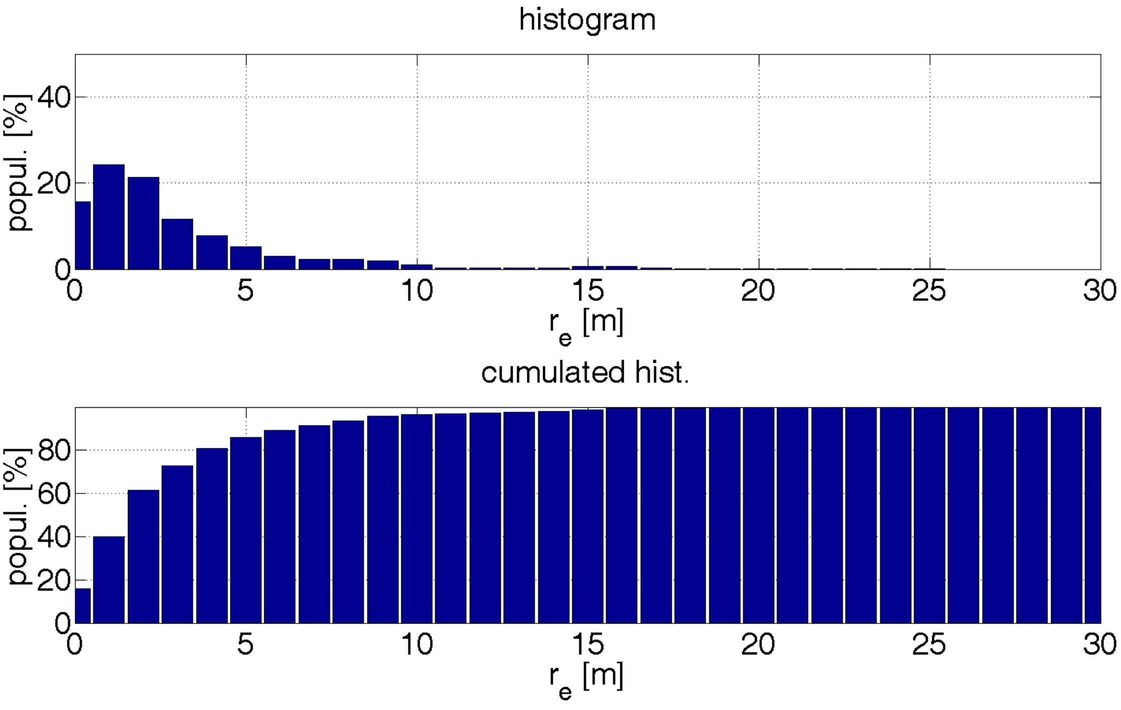
(**a**)

(**b**)

**Figure 21.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of GPS receiver, inertial sensors, probability map, stereo camera and 3D model of the environment. The error of estimating direction at (400 m, 120 m) introduces big errors later on. The filter managed to recover at (240 m, 80 m). (**b**) Histograms of distance errors.

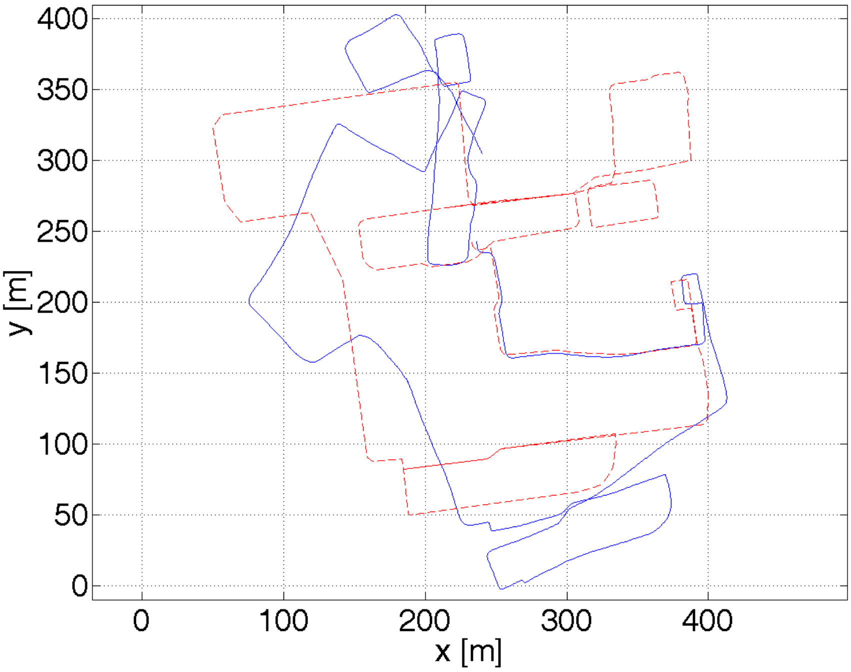


(**a**)

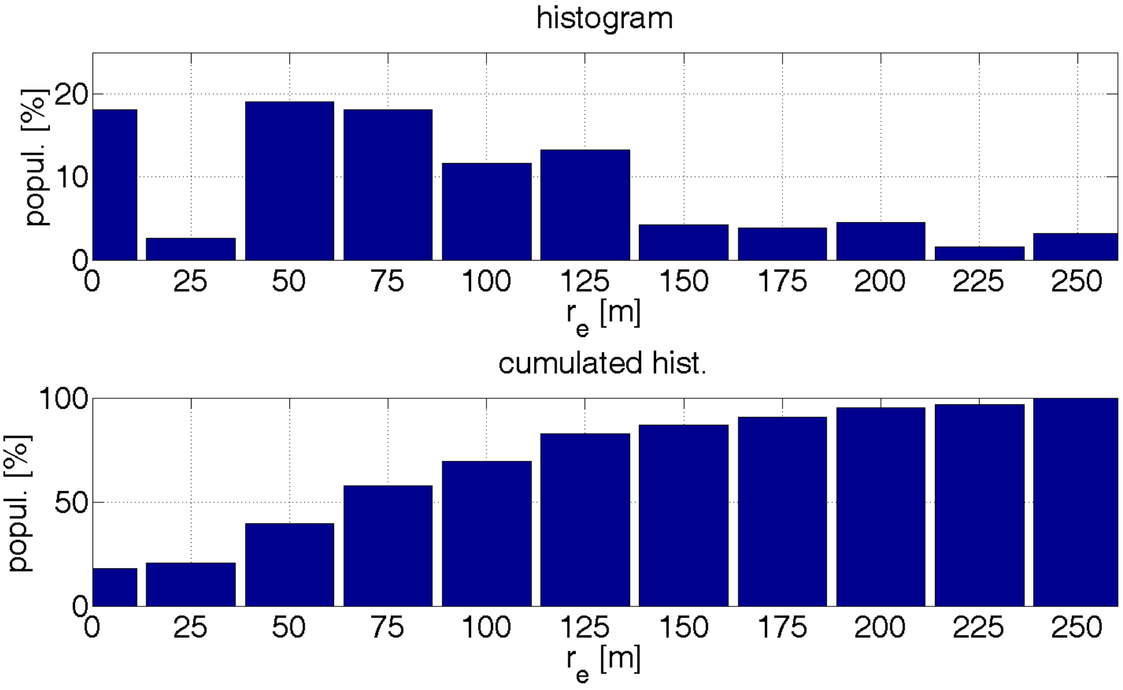


(**b**)

**Figure 22.** (**a**) Red dashed line—reference path. Blue solid line—estimated path with a help of the inertial sensors, *i.e.*, gyroscope for estimating direction change and accelerometers for counting steps and their lengths. As times goes on, the errors accumulate due to angular random walk. (**b**) Histograms of distance errors.



(**a**)



(**b**)

## 6. Conclusions

The results were calculated based on *ca.* 1,900 GPS readouts, 3,400 particle filter estimates and 19,000 pictures from the stereo camera. The particle filter using all available data sources provides by far the best accuracy. For 99% of the time the accuracy is better than 4.5 m. This is still too large an error for safe navigation of visually impaired pedestrians. The main source of errors were introduced by misestimated direction, which is visible in Figure 22 at coordinates (410 $\mathrm{m}$, 125 $\mathrm{m}$). The turn is misestimated by $\approx 20^{\circ}$. This error is too large for the non-linearity error mentioned in Section 3.1.1 or any other gyroscope error. Possibly the sensors came adrift and did not change its direction accordingly. This error, when not corrected immediately, has a ripple effect. This can be, however, taken care of. Also, more accurate sensors are available on the market. It is surmised that an error of 3 m for 99% of time is achievable.

It is interesting to note that when the initial position is known, the location of a pedestrian can be estimated without having to use a GPS receiver. The setup number 5 (see Table 1) gives best results after the setup number 4 and it is better than sole GPS readouts. Needless to say, this approach can be used in limited areas, e.g., inside buildings where GPS readouts are not available. The probability map concept limits the positioning error to the width of a street, *ca.* 10 m. Due to the imposed constraints on the pedestrian kinematics, the error dropped 3 times in terms of a root-mean-square error, which poses the best improvement from all other additional sources.

As mentioned before, on the bases of GPS readouts and the probability map, a user's location can be estimated within *ca.* 10 m. A stereo camera can refine the user's position down to 2 m, providing there are no unexpected objects, or difficult objects like trees, whose crown's radius could differ from the assumed 2 m. GPS readouts complement nicely with the algorithm in which the depth images derived from terrain model and stereoscopy are compared. In the presence of high buildings, GPS readouts are compromised. Under these circumstances the image matching algorithm performs well as the walls of buildings are confidently detectable by a stereo camera. On the other hand, in an open space GPS readouts are accurate down to, *ca.* 2–3 m. Then in turn, the image matching algorithm has no reference points to perform a viable comparison.

The presented algorithm works off-line. It is quite computationally demanding. The particle filtering algorithm requires generation of *ca.* 500 particles for good positioning accuracy. However, the task can be handled on-line without any optimization techniques by a dual core, 2.5 GHz standard notebook.

### Acknowledgements

The project is financed by the National Centre of Research and Development of Poland in years 2010–2013 under the grant NR02-0083-10.

### References and Notes

1. Official U.S. Information Website about the GPS System. Available online: http://www.gps.gov (accessed on 14 February 2012).

2. Modsching, M.; Kramer, R.; Hagen, K. Field Trial on GPS Accuracy in a Medium Size City: The Influence of Built-up. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication*, Hannover, Germany, 16 March 2006; pp. 209-121.
3. Baranski, P.; Strumillo, P. Fusion of Data From Inertial Sensors, Raster Maps and GPS for Estimation of Pedestrian Geographic Location in Urban Terrain. *Metrol. Measure. Syst.* **2011**, *1*, 145-158.
4. Ong, R.B.; Petovello, M.G.; Lachapelle, G. Assessment of GPS/GLONASS RTK under Various Operational Conditions. In *Proceedings of the 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2009)*, Savannah, GA, USA, 22–25 September 2009; pp. 3297-3308.
5. Trajkowski, K. K.; Sterle, O.; Stopar, B. Sturdy Positioning with High Sensitivity GPS Sensors under Adverse Conditions. *Sensors* **2010**, *10*, 8332-8347.
6. Diggelen, F. Generating Assisted Data. In *A-GPS: Assisted GPS, GNSS, and SBAS*; Artech House: Boston, MA, USA, 2009.
7. Krakiwsky, E.; Harris, C.; Wong, R. A Kalman Filter for Integrating Dead Reckoning, Map Matching and GPS Positioning. In *Proceedings of Position Location and Navigation Symposium, Navigation into the 21st Century*, Orlando, FL, USA, 29 November–2 December 1988; pp. 39-46.
8. Taghipour, S.; Meybodi, M.; Taghipour, A. An Algorithm for Map Matching for Car Navigation System. In *Proceedings of the 3rd International Conference on Information and Communication Technologies From Theory to Applications*, Damascus, Syria, 7–11 April 2008; pp. 1-5.
9. Mezentsev, O.; Lachapelle G. Pedestrian Dead Reckoning—A Solution to Navigation in GPS Signal Degraded Areas. *Geomatica* **2005**, *59*, 145-182.
10. Fang, L.; Panos, J. Design of a Wireless Assisted Pedestrian Dead Reckoning System—The NavMote EXperience. *IEEE Trans. Instrume. Measur.* **2005**, *50*, 2342-2359.
11. van der Spek, S.; van Schaick, J.; de Bois, P.; de Haan, R. Sensing Human Activity: GPS Tracking. *Sensors* **2009**, *9*, 3033-3055.
12. Woodman, O. *An Introduction to Inertial Navigation*; Technical Report Number 696; University of Cambridge: Cambridge, UK, 2007. Available oneline: http://www.cl.cam.ac.uk/techreports/ (accessed on 1 February 2012).
13. Jirawimut, R.; Ptasinski, P.; Garaj, V. A Method for Dead Reckoning Parameter Correction in Pedestrian Navigation System. *IEEE Trans. Instrum. Measur.* **2003**, *52*, 209-215.
14. Godha, S.; Lachapelle, G.; Cannon, M. Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment. In *Proceedings of the 19th International Technical Meeting (ION GNSS 2006)*, Fort Worth, TX, USA, 26–29 September 2006; pp. 2151-2164.
15. Ladetto, Q.; Merminod, B. An Alternative Approach to Vision Techniques: Pedestrian Navigation System Based on Digital Magnetic Compass and Gyroscope Integration. In *Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, FL, USA, 14–18 July 2002; pp. 145-150.
16. Seung-Hyun, K. Statistical Analysis of Urban GPS Multipaths and Pseudo-Range Measurement Errors. *IEEE Trans. Aerospace Electr. Syst.* **2011**, *47*, 1101-1113.

17. Bancroft, J.B.; Lachapelle, G. Data Fusion Algorithms for Multiple Inertial Measurement Units. *Sensors* **2011**, *11*, 6771-6798.
18. Jirawimut, R.; Shah, M. A.; Ptasinski, P.; Cecelja, F.; Balachandran, W. Integrated DGPS and Dead Reckoning for A Pedestrian Navigation System in Signal Blocked Environments. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Salt Lake City, UT, USA, 19–22 September 2000; pp. 1741-1747.
19. Williamson, J.; Strachan, S.; Murray-Smith R. It's a Long Way to Monte Carlo: Probabilistic Display in GPS Navigation. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services*, Espoo, Finland, 12–15 September 2006; pp. 89-96.
20. Ceranka, S.; Niedzwiecki, M. Application of Particle Filtering in Navigation System for Blind. In *Proceedings of Signal Processing and its Applications*, Paris, France, 1–4 July 2003; pp. 495-498.
21. Widyawan; Klepal, M.; Beauregard, S. A Backtracking Particle Filter for Fusing Building Plans with PDR Displacement Estimates. In *Proceedings of Positioning, Navigation and Communication*, Hannover, Germany, 27 March 2008; pp. 207-212.
22. Hesch, J.A.; Roumeliotis, S.I. An Indoor Localization Aid for the Visually Impaired. In *2007 IEEE International Conference on Robotics and Automation (ICRA'07)*, Roma, Italy, 10–14 April 2007; pp. 3545-3551.
23. Girard, G.; Cote, S.; Zlatanova, S.; Barette, Y.; St-Pierre, J.; Van Oosterom, P. Indoor Pedestrian Navigation Using Foot-Mounted IMU and Portable Ultrasound Range Sensors. *Sensors* **2011**, *11*, 7606-7624.
24. Thrun, S.; Montemerlo, M. FastSLAM. In *A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*; Siciliano, B.; Khatib, O., Groen, F., Eds.; Springer: Berlin, Heidelberg, Germany, 2007.
25. Kleiner, A.; Dali, S. Decentralized SLAM for Pedestrians without Direct Communication. In *Proceedings of the International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, 29 October–2 November 2007; pp. 1461-1466.
26. Cecelja, F.; Balachandran, W.; Jirawimut, R. A Stereo Vision System for Pedestrian Navigation. In *Proceedings of the 17th International Conference on Applied Electromagnetics and Communications*, Dubrovnik, Croatia, 1–3 October 2003; pp. 30-33.
27. Burguera, A.; Gonzlez, Y.; Oliver, G. Sonar Sensor Models and Their Application to Mobile Robot Localization. *Sensors* **2009**, *9*, 10217-10243.
28. Brown, M.Z., Burschka, D., Hager, G.D. Advances in Computational Stereo. *IEEE Trans. Patt. Anal. Mach. Intell.* **2003**, *25*, 9931008.
29. Szeliski, R.; Scharstein, D. Symmetric Sub-Pixel Stereo Matching. In *Proceedings of the 7th European Conference on Computer Vision*, Copenhagen, Denmark, 27 May–2 June 2002; pp. 525-540.
30. Howe, D.A.; Allan, D.W.; Barnes, J.A. Properties of Signal Sources and Measurement Methods. In *Proceedings of the 35th Annual Symposium on Frequency Control*, Philadelphia, PA, USA, 27–29 May 1981.
31. *IEEE Standard for Inertial Sensor Terminology*; IEEE Std 528-2001 (Revision of IEEE Std 528-1994); The Institute of Electrical and Electronics Engineers, Inc.: New York, NY, USA, 2001.

32. International Hydrographic Organization. *User's Handbook On Datum Transformations Involving WGS 84*. Available online: www.iho.int/iho_pubs/standard/S60_Ed3Eng.pdf (accessed on 10 February 2012).
33. Brown, R., Hwang, P. Transformation of Random Variables. In *Introduction to Random Signals and Applied Kalman Filtering*; John Willey & Sons: New York, NY, USA, 1997; pp. 42-49.
34. Park, S.A.; Suh, Y.S. A Zero Velocity Detection Algorithm Using Inertial Sensors for Pedestrian Navigation Systems. *Sensors* **2010**, *10*, 9163-9178.
35. Anderson, B.D.O.; Moore, J.B. *Optimal Filtering*; Prentice Hall: Englewood Cliffs, NJ, USA, 1972.
36. Rodriguez, J.; Aggarwal, J. Quantization Error in Stereo Imaging. In *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, USA, 5–9 June 1988; pp. 153-158.
37. Cappe, O.; Godsill, S.J.; Moulines, E. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proc. IEEE* **2007**, *95*, 899-924.