BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Mechanical Engineering

PHD THESIS

Brno, 2018 Ing. Jan Hrbáček

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS
ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

NAVIGATION AND INFORMATION SYSTEM FOR VISUALLY IMPAIRED

NAVIGAČNÍ A INFORMAČNÍ SYSTÉM PRO NEVIDOMÉ

PHD THESIS
DIZERTAČNÍ PRÁCE

AUTHOR **Ing. JAN HRBÁČEK**
AUTOR PRÁCE

SUPERVISOR **doc. Ing. VLADISLAV SINGULE, CSc.**
VEDOUCÍ PRÁCE

BRNO 2018

Abstract

Visual impairment is one of the most widespread physical handicaps – as much as 3 % of the world's population suffers from visual impairment or complete blindness. Vision loss substantially worsens one's ability to orient in the environment – without the knowledge of spatial arrangement, normally acquired predominantly by one's vision, the impaired simply does not know which way to go. The most usual solution is an accompanying person; this service is very demanding though and the sightless have to fully rely on the accompaniment.

This thesis explores ways to support visually impaired users' orientation in space by employment of existing sensory technology and application of appropriate processing methods. The subject is researched through an analogy to mobile robotics, by partitioning the subject to the separate problems of localization and path planning. While the methods of path planning are generally existing, pedestrian localization often suffers from major inaccuracies and complicates usage of standard navigation devices by the visually impaired users.

An improvement of pose estimate quality can be accomplished by a multitude of approaches studied by the analytical section. In the first stage, the thesis proposes data fusion between an ordinary GPS receiver and a pedestrian dead reckoning unit, leading to preserving local trajectory shape feature faithfully. To mitigate remaining offset errors, design of a globally referenced mechanism of natural landmark detection and matching is provided.

Building on the existing graph search formalism, the path planning part of the thesis examines optimality criteria suitable for navigating the visually impaired user through urban terrain. A human-oriented fuzzy logic driven high-level instructions generator is then devised together with a real-time haptic feedback delivering heading directions.

The performance of the proposed techniques was evaluated in real-world scenarios, aiming to capture the particularities of the target urban environment. The outcomes have shown considerable improvements in both maximum and mean positioning errors.

Keywords

Pedestrian Localization, Data Fusion, Unscented Kalman Filter, Particle Filter, Image Processing, Natural Landmark Detection, Path Planning, Fuzzy Logic

Citation

Jan Hrbáček. *Navigation and Information System for Visually Impaired*. Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2018. 112 p. PhD thesis supervisor doc. Ing. Vladislav Singule, CSc..

Abstrakt

Poškození zraku je jedním z nejčastějších tělesných postižení – udává se, že až 3 % populace trpí vážným poškozením nebo ztrátou zraku. Oslepnutí výrazně zhoršuje schopnost orientace a pohybu v okolním prostředí – bez znalosti uspořádání prostoru, jinak získané převážně pomocí zraku, postižený zkrátka neví, kudy se pohybovat ke svému cíli. Obvyklým řešením problému orientace v neznámých prostředích je doprovod nevidomého osobou se zdravým zrakem; tato služba je však velmi náročná a nevidomý se musí plně spolehnout na doprovod.

Tato práce zkoumá možnosti, kterými by bylo možné postiženým ulehčit orientaci v prostoru, a to využitím existujících senzorických prostředků a vhodného zpracování jejich dat. Téma je zpracováno skrze analogii s mobilní robotikou, v jejímž duchu je rozděleno na část lokalizace a plánování cesty. Zatímco metody plánování cesty jsou vesměs k dispozici, lokalizace chodce často trpí značnými nepřesnostmi určení polohy a komplikuje tak využití standardních navigačních přístrojů nevidomými uživateli.

Zlepšení odhadu polohy může být dosaženo vícero cestami, zkoumanými analytickou kapitolou. Předložená práce prvně navrhuje fúzi obvyklého přijímače systému GPS s chodeckou odometrickou jednotkou, což vede k zachování věrného tvaru trajektorie na lokální úrovni. Pro zmírnění zbývající chyby posunu odhadu je proveden návrh využití přirozených význačných bodů prostředí, které jsou vztaženy ke globální referenci polohy.

Na základě existujících formalismů vyhledávání v grafu jsou zkoumána kritéria optimality vhodná pro volbu cesty nevidomého skrz městské prostředí. Generátor vysokoúrovňových instrukcí založený na fuzzy logice je potom budován s motivací uživatelského rozhraní působícího lidsky; doplňkem je okamžitý haptický výstup korigující odchylku směru.

Chování navržených principů bylo vyhodnoceno na základě realistických experimentů zachycujících specifika cílového městského prostředí. Výsledky vykazují značná zlepšení jak maximálních, tak středních ukazatelů chyby určení polohy.

Klíčová slova

pěší lokalizace, fúze dat, unscented Kalmánův filtr, particle filtr, zpracování obrazu, detekce význačných bodů prostředí, plánování trasy, fuzzy logika

Citace

Jan Hrbáček. *Navigační a informační systém pro nevidomé*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 112 p. Vedoucí dizertační práce doc. Ing. Vladislav Singule, CSc..

Navigation and Information System for Visually Impaired

Declaration

I declare that this thesis has been composed solely by myself under supervision of doc. Ing. Vladislav Sigule, CSc., and using the sources listed in references.

.....

Jan Hrbáček

2018-08-30

Acknowledgements

I would like to express my gratitude to my adviser doc. Ing. Vladislav Singule, CSc. for leading my steps through the PhD study and towards this dissertation. Furthermore, I greatly appreciate discussions on the presented topics we had with doc. Ing. Stanislav Věchet, Ph.D. and doc. Ing. Jiří Krejsa, Ph.D. Last but not the least, I would like to thank my parents for showing me the importance of education; I am also immensely grateful to my wife and daughter for their patience during last stages of my efforts.

Contents

Chapter 1

Introduction

Visual impairment is one of the most widespread physical handicaps – according to World Health Organization, as much as 3 % of the world's population suffers from visual impairment or complete blindness [1]. This number may even increase in the long term due to life prolonging and development of vision defects strongly bound to old age with 81 % of the affected people aged 50 years and above [1] (similarly to age-induced degradation of the other senses). However, visual impairment affects also people in productive or even pre-productive age, either due to a congenital defect, a disease or an injury. In such cases the handicap impacts life of the disabled incomparably more.

Compared to deficiency in any other of the human senses, vision loss substantially worsens one's ability to orient in both known and unknown environments. The sightless is usually able to learn how to navigate through common spaces (home, workplace etc.), although any change in spatial arrangement can cause potentially fatal problems. The biggest challenge, however, is the "outer world" – otherwise simple tasks (as a journey to a previously unvisited part of town) become difficult and even dangerous. Without the knowledge of spatial arrangement, normally acquired predominantly by one's vision, the impaired simply does not know which way to go.

1.1 Motivation

The most usual solution to the problem of orientation in unknown environments is an accompanying person that helps the handicapped to reach their goal and provides them with information compensating their handicap. This service is very demanding though and the sightless have to fully rely on the accompaniment (at least until they safely learn the new route or place). As the common outer environment is unlikely to become simple enough to allow independent initial navigation of the visually impaired, the only realistic option to facilitate self-reliance to sightless people is presumably through an electronic apparatus compensating for non-functionality of their vision by artificial means.

Technical solutions supporting pedestrian orientation and navigation through unknown environment are known and existing to some extent, but this research topic seems to be less popular in literature compared to other areas of interest with similar aims and techniques (sensory signal processing, data fusion, state space searching etc.). One of the motivations of this work is thus extending the published knowledge of methods usable for application in this field and experimental results of their employment.

1.2 Research objectives

The aim of this thesis is to research, select and develop a set of methods that together form the core of the navigation and information system for visually impaired. The functionality enabled by these methods should allow its user to traverse through previously unknown outdoor terrain with focus on urban areas with their specifics.

The solution can be broken down into the following partial goals:

1. *localization method* development that provides position estimation accurate and repeatable enough to support robust operation of the whole system;
2. *path planning method* development that is optimal for selection of the global and local route of a visually impaired user mainly through urban environment, optionally with an *obstacle detection* subsystem as a potential white cane complement or replacement;
3. *evaluation* of the proposed methods in realistic environments.

1.3 Thesis organization

In order to get deeper background on the topics of visual impairment, the presented thesis starts in Chapter 2 with an analysis of the state of the art on the subject of helping sightless people overcome their handicap, focusing on the technical solutions available.

Based on the aforementioned research objectives and analogies provided by Section 2.3, the main theoretical content of the thesis is divided into two thematic chapters: Chapter 3 Localization and Chapter 4 Path planning. Each of them is divided to an analytical and design part; the former provides literature research and theoretical background, while the latter builds on it and derives one or more potential solutions.

Chapter 5 then provides details on the realized implementations. Proper evaluation on real-world experimental data and in representative environments is an important part of the work and is provided in Chapter 6, as well as discussion of the the results. Finally, the conclusions are given in Chapter 7.

Chapter 2

State of the Art

Since the thesis aims to provide methods that could help relieve a medical condition, this chapter provides an overview of visual impairment consequences and existing aids related to spatial orientation. In the last section, a conceptual link to the field of autonomous mobile robotics is presented.

2.1 Visual impairment and orientation in space

The extent of visual impairment (VI) is, naturally, individual – as is the practical impact on each impaired person. Depending on a multitude of factors, including both objective parameters as sight loss percentage or age and subjective ones as openness to information technology, the travel self-reliance ranges into both extremes.

Spatial orientation of people with VI has been studied mostly within the fields of geography, behavioral science or sociology. The research objective usually relates to urban environment arrangement and mobility with respect to VI. For one, [2] provides commuting experience of VI individuals in the San Francisco Bay Area categorized according to the three vital resources: transportation, assistive technology, and mobile technology; other sources [3, 4] provide similar insights. For the purposes of this state of the art analysis, even such localized studies provide relatively generalizable outcomes, commented in the following paragraphs.

Transportation An effective way to delegate the burden of spatial orientation to a different (sighted) person, in many cases enabling the VI passenger to cover the majority of the distance. Particularly in the case of paratransit[1] or private car options, the impaired can be almost completely spared the pitfalls of outdoor environment traversal. Public transport systems, while imposing greater demands upon the passenger, still provide considerable benefits to the user.

[1] A door-to-door transportation service usually provided by municipalities to physically impaired clients.

A limitation of confined urban spaces is discussed in [2] – public transport stops often combine several lines which makes it difficult for the impaired to choose the right vehicle. The solution, if available at all, is highly locally dependent; in the Czech Republic, for instance, the public transport vehicles carry an identification speaker that can be remotely activated by an appropriate device – see Section 2.2.2 below.

Assistive technology Assistive technology is by definition any technology that improves individuals' functional capabilities [2] – white cane being one of the most frequent tools. Usage of these means can be perceived by the users in different ways – by making them discernibly impaired, they are likely to expect greater carefulness of their surrounding, which some reportedly do not appreciate:

> "I don't like the fact that it makes you, it brings you more attention than other people.
>
> (...) Everyone says that everybody else uses the cane and all that stuff. But when I went to my [specialist school visit], they say the majority of people don't like using one." [4]

The above quotation formulates a soft requirement on the assistive tools, to be adopted by a broad audience: their employment should not make the user embarrassed; this aspects seems greatly relevant in the younger part of the VI population. On the other hand, even when a tool cannot be executed in an unobtrusive form, it may be still adopted if its benefits clearly prevail.

Influence of urban architecture Several of the above and below cited papers notice the role urban architecture plays in accessibility of city centers for the VI persons. There are both safety and orientation aspects present on this topic; providing a dedicated pedestrian corridor (ideally equipped with tactile pavement or other structural aids discussed in the following section) may simplify traversal of the VI pedestrians significantly.

Despite the generally growing public attention to disabilities, there are also recent examples of architecture trying to improve public spaces, but doing so with very little respect to impaired users. One of the widely discussed modern urban concepts, by some uncritically perceived as "a civilised, communal approach to city living that clearly works" [5], yet in fact creating chaotic environments confusing and endangering the VI pedestrians [6], are the shared surfaces:

> Shared surface streets (sometimes called a level surface) are where the road and pavement are built at the same level, removing the kerb so that cars, buses, cyclists and pedestrians share the same surface. In some cases, controlled crossings (pelican crossings) are also removed.
>
> Shared surface streets are dangerous for people with sight loss, who rely upon the presence of the kerb to know they are on the pavement and not in the road.

> (...) Guide Dogs has been campaigning against the use of shared surface streets as part of our Streets Ahead campaign, supported by organisations representing disabled people across the disability sector, older people and other groups. [7]

It is apparent that lack of natural spatial orientation aids needs to be compensated for somehow; modern technologies seem to fit this application. It is, however, shown just below that the existing approaches do not seem to make that much of a difference with regard to the spatial orientation.

Mobile technology and global navigation systems usage The functionality of the set of methods proposed in this thesis has similar outcomes as the ubiquitous "satnav" or "GPS" devices, used by drivers on a daily basis. There is a fundamental difference though; unlike the driver, the VI user has a very limited ability to validate and correct the generated navigation cues. The demands placed on the described system – mainly the localization module – are for this reason qualitatively much more stringent. While the mean performance under favorable conditions may be comparable, the real difference lies in worst-case behavior; a 10-meter localization error commonly observed on "satnav" may cause serious problems to the blind user. This kind of a frustrating experience was tellingly expressed by one of the aforementioned interviews:

> "I turned on my GPS... and I asked Siri to give me directions from my current location to [an address]. And that went well but it told me to walk to the route. And it's like, well, I don't know where the route is. And so I canceled that. I then asked it for current location to [the address] again and that's when it said to continue on [Street Name]. And it's like okay, good, I know I'm on [Street Name]... I was walking and I was walking and I thought this is really far... I also knew it was a totally different route from what I'd been told... I was definitely late so I went a couple more blocks and finally my GPS said that I had arrived at my destination, it was on the right. Well, of course I couldn't find the entrance. So then I called... and so they came out and found me." [2]

The outcome of this experience is clear and supports motivation foundations of this thesis – even though recent technical advances have already changed life of many people for the better, users are still compensating many imperfections by their own perception and intelligence. There are, however, people, whose perception is damaged and intelligence itself cannot help sufficiently; the technology still needs to evolve.

2.2 Existing aids for visually impaired

For reference on the existing aids, their merits for the user and involved technology, this section lists both non-electronic and electronic means that can be utilized by a VI user to improve their orientation in (mainly urban) space.

2.2.1 Non-electronic aids

Purely mechanical or constructional/structural tools have played the primary role in self-reliant movement of VI individuals; this is obviously partially given by their earlier invention compared to electronic aids.

White cane Perhaps the most widely used and best-known aid; besides the obvious tactile-sensory role, it also serves as a distinct marker making its blind user immediately identifiable by other traffic participants and thus helps prevent situations endangering the user.

There are various techniques of cane manipulation dependent on the traversed terrain or environment [8]. Its universality makes the cane a very effective tool of near-space exploration, covering the area of the lower part of the body. For experienced user with proper technique, even detailed features of the ground can be sensed, preventing stumbling or falling down. This sensitivity is very hard to achieve by other, no matter how "high-tech", sensory principles.

Structural aids The white cane is very useful not only for detection of obstacles, but also in utilization of inherent city street features – building walls, sidewalks and other detectable shapes. By swinging their cane side-to-side, the impaired is able to follow such a feature very easily.

Tactile paving, as illustrated in Figure 2.1, is a modern supplement to the inherent urban tactile features. Generally, they are intended to be sensed by one's feet even through shoes, for which they are very distinctively shaped. There are the following common types of the pavement [9]:

- *Guidance path surface* is the most interesting surface type from the point of view of this thesis. It is designed to directionally lead the VI pedestrians, either due to lack of inherent tactile features or because it is needed to guide the impaired along a specific path (e.g. to find an elevator or avoid fixed obstacles).

- *Blister surface* acts as a delimiter of pedestrian path segments, usually as a substitution of detectable height change where the curbs are not pronounced enough or where an explicit warning to the VI pedestrian needs to be given. Naturally, this is mostly often the case of crosswalks. The blister surface is also used to denote guidance path junctions.

- Other special types include on- and off-street *platform edge warning surfaces (lozenge)*, *hazard warning surface (corduroy)*, shared cycle track/footway *delimitation strips* or *information surface*.

Figure 2.1: Tactile pavement in Sylvia Park New Train Station (yellow color) [10]

Guide dogs Another well-known help for the impaired is provided by the specially trained guide dogs. Such a dog is able to lead its master safely through the urban environment – walk centrally along the pavement, avoid obstacles, stop at ground irregularities (curbs, steps) or find doors [11].

However, the utilization of a guide dog is still conditioned by the fact that its master at least schematically knows the arrangement of the environment and has a rough plan of such environment traversal – which is a task the dog cannot help with on new/unknown routes.

2.2.2 Electronic aids

Intuitively, electronic aids have much broader functional possibilities than the mechanical ones; ongoing miniaturization of electronics coupled with advances in computational methods enable improvements in achievable functionality every year. Implants can be perceived as the notional cutting edge of technology – even able to supersede non-functional vision by stimulating retinal nerves; this thesis is rather interested in external devices though.

Maps There are dedicated maps for visually impaired, based solely on the haptic principle and allowing the users to get a reasonable level of spatial awareness. Apart from the purely paper ones, there are projects that try to convert the usual street map into a haptic presentation usable for print on the specialized embossing printers. Covering Czech Re-

public, Mapy.cz[2] provide such service; worldwide, the Open Street Maps project launched "OSM for the blind"[3] aiming in a similar direction.

Acoustic beacons Acoustic signaling methods designed to inform blind citizens are well known and beneficial: for example, pedestrian traffic lights are often equipped with an auxiliary acoustic signal indicating green or red light. Furthermore, specialized signaling devices can be found in public transport (on-demand identification of vehicle number[4]), public service buildings and other places. An important aspect of general adoption is significant local dependence; particular implementations and suppliers of such solutions often differ even within a single country.

Range finders, obstacle detectors With the advent of various sensors used industrially or in robotics, it is understandable that there have been attempts of using such technology for human-oriented applications as well. The employment commonly takes the form of an obstacle detector, trying to replace or complement the white cane. Functionally, they are based on a single or multiple range-finding sensors attached to one of these locations:

- *hand-held* devices are designed either as an independent device or as an integral part of the white cane, such as the Ultracane instrument[5];
- *torso-mounted* contraptions typically cover the upper half of the body height where the white cane is not effective – e.g. the Czech project "Ven ze tmy"[6];
- *head-worn* gadgets provide ranging coaxial with one's face which, with proper technique, may be exploited very efficiently, such as the range-finding glasses Ambutech iGlasses[7].

Equally important as sensing quality stands the aid's ability to convey the obstacle distance information to the impaired. The often implicit choice of an audio interface may not constitute the best solution though; hearing as the remaining long-reaching sense is better left unburdened as much as possible. Haptic solutions appear as a less disturbing mechanism, although finding proper representation patterns for the distance value might be challenging, particularly in the case of a sensor array.

Specialized smart phone applications It has already been noted that general-purpose assisting services such as Siri or alike are often not usable enough for a visually impaired used – being based on a common smart phone, they are fairly limited in sensory perception.

[2] http://hapticke.mapy.cz/
[3] https://wiki.openstreetmap.org/wiki/OSM_for_the_blind
[4] http://www.ok.cz/elvos/Majacky.html
[5] http://www.ultracane.com/
[6] http://venzetmy.eblog.cz/
[7] https://ambutech.com/shop-online/iglasses%E2%84%A2-ultrasonic-mobility-aid

Currently, impaired user do not have many other choices of sensory augmentation though, so that even using a plain GNSS-aided smart phone is still better than no aid, even though it leads to unpleasant experience under poor conditions.

Even though the sensory aspect is not easily solvable without additional devices, there are smart phone applications at least trying to provide navigation instructions tailored for VI users, no matter how flimsy the GNSS "undercarriage" may be. As an example, the app DotWalker[8] implements a blind-friendly point-to-point navigation with user interface based on voice synthesis and recognition.

2.3 Mobile robotics as a related field

As indicated in the introductory chapter, this thesis explores ways to support visually impaired users' orientation in space through use of existing sensory technology and application of appropriate processing methods. Coincidentally, *localization*, *navigation* and *obstacle avoidance* are probably the most important topics addressed in the field of (autonomous) mobile robotics. This fact leads to interesting analogies, thoroughly elaborated in analytical sections of the next chapters; for each of the solved problems, the following can be summed up:

- *Localization* plays similarly critical role as in mobile robotics; poor position estimate leads to poor performance of all dependent subsystems, including navigation.
- *Navigation* shares core algorithmization techniques, but differs significantly in the action step. While robot chassis only understands low-level commands (e.g. angular speed of each wheel), the human user can perform a variety of high-level operations: go along a wall, cross a street etc.
- *Obstacle avoidance* is vital in mobile robotics – at least some primitive form of it is present in majority of applications. The VI user can be, however, expected to operate their white cane – obstacle detection can thus be left as an optional, non-critical subsystem.

In literature, this subject is not covered particularly widely; the following papers are known to the author which approach the subject in a comparable manner. In [12], RFID tags are proposed for pedestrian localization; [13] concentrates primarily on stereo vision usage for localization by comparing the scene to a 3D model of urban environment. [14] proposes feature extraction from camera image; [15] then exploits space topology to build a probabilistic motion model.

[8] `http://www.dotwalkerpro.com/dw/dotwalker_en.html`

Chapter 3

Localization

Very often, the localization subsystem serves as foundation for a multitude of higher layers, ranging from navigation and obstacle avoidance to specific mission execution tasks – their performance then usually strongly depends on quality of the pose estimates. No matter how advanced the navigation algorithms are, they cannot perform well with a flawed estimate.

This chapter will progress from foundation principles of representing a *pose*, through an overview of technologies and suitable data processing methods, finally to an essential, implementation-independent description of the proposed localization arrangement.

3.1 Analysis

The goal of this section is to introduce the range of methods and technologies that lie within the fields of interest. Brief characteristics, strong and weak points are given to prepare for the particular employment proposals in Section 3.2.

3.1.1 Used notation conventions

The notation of important quantities varies widely in the literature – for example, among the authoritative sources on application of probabilistic principles in robotics, there is even no unity in notation of as fundamental quantity as covariance matrix: while Thrun et al. in [16] and Siegwart & Nourbakhsh in [17] use Σ, Choset et al. in [18] use $\mathbf{P}$ for the same purpose. Individual papers then often introduce further custom conventions.

This thesis follows the conventions used within the mobile robotics team at Faculty of Mechanical Engineering of Brno University of Technology; the following table sums up the major ones.

$\mathbf{x}_k$	state vector at a discrete time step k
$\mathbf{x}_k^\theta$	θ component of the state vector at a discrete time step k
$\hat{\mathbf{x}}_{k+1\|k}$	estimate of state vector at time step $k+1$ based on state from step k
$\boldsymbol{\mathcal{X}}$	sigma points generated around state $\mathbf{x}$
$\mathbf{A}^\top$	transpose of matrix $\mathbf{A}$
$\mathbf{P}$	covariance matrix
$\mathbf{P}^{x,z}$	cross-covariance matrix between system state and system output

3.1.2 Representations of attitude

With respect to the motion of humans, only direct or proper motion[1] can be safely assumed. As such, proper motion in Euclidean space consists of two components: *translation* and *rotation*. The state quantity resulting from motion is the *pose* of a body – its position information combined with attitude.

While change in position can be unambiguously represented by a translation vector, rotation is inherently more complex. Firstly, attitude as an angular position is a cyclic quantity and when normalized to the usual range of $(-\pi;\pi\rangle$ or $\langle 0;2\pi)$ it evinces a discontinuity complicating computational operations. Secondly, rotations in three-dimensional space are generally not commutative and the order of applying individual rotations is important.

A great reference and comparison of the attitude representations is given in [19]. In the paper [20], a thorough discussion of various operations on the group of rotations is given.

Euler angles, Tait–Bryan angles The most intuitive approach to express attitude in three dimensions is a sequence of consecutive elemental rotations about three axes. Depending on the axes selection, the sequences can be divided into two groups, as illustrated in Figure 3.1:

- *Euler angles* that use the same axis for the first and last rotation in the sequence. The most common of such sequences is *z-x-z* and the angles are referred to as *spin*, *nutation*, and *precession*, denoted by ϕ, θ and ψ respectively, or α, β and γ.
- *Tait–Bryan angles* whose sequence is formed by rotations about all three axes. The well-known triplet of *roll*, *pitch* and *yaw* is again denoted by ϕ, θ and ψ and corresponds to the sequence *x-y-z*.

Despite this classification, both groups are sometimes commonly called Euler angles; to distinguish the first group, it is in such case referred to as the *proper Euler angles*.

[1] As opposed to indirect motions like reflection or improper rotation that invert the chirality – handedness – of the body.

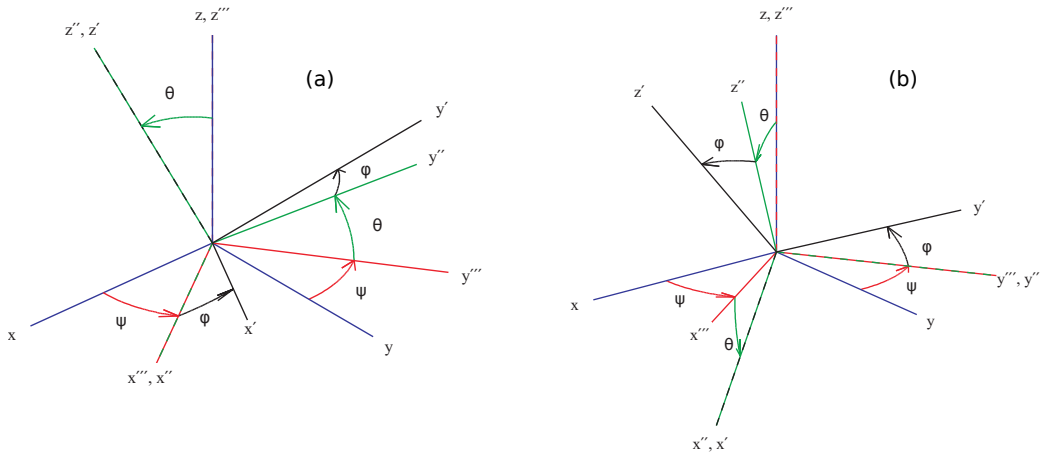


Figure 3.1: Euler $(3,1,3)$ (a) versus Tait–Bryan $(1,2,3)$ (b) angles; adopted from [19]

There is a major drawback of using any of the Euler angles sequence – the *gimbal lock*. It stands for a singularity case in which two of the axes are aligned and thence the system loses one degree of freedom.

Rotation matrices Matrices of dimension 3×3 whose multiplication with a vector leads to rotation of the vector without change of its length are called rotation matrices and form the *special orthogonal group SO(3)*. A rotation matrix is often coined *direction cosine matrix* (DCM), since its elements can be expressed as cosines of angles included between each axis of the body $x^{\mathrm{b}}, y^{\mathrm{b}}, z^{\mathrm{b}}$ and each axis of the reference system x, y, z:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta_{x^{\mathrm{b}},x}) & \cos(\theta_{x^{\mathrm{b}},y}) & \cos(\theta_{x^{\mathrm{b}},z}) \\ \cos(\theta_{y^{\mathrm{b}},x}) & \cos(\theta_{y^{\mathrm{b}},y}) & \cos(\theta_{y^{\mathrm{b}},z}) \\ \cos(\theta_{z^{\mathrm{b}},x}) & \cos(\theta_{z^{\mathrm{b}},y}) & \cos(\theta_{z^{\mathrm{b}},z}) \end{bmatrix} \tag{3.1}$$

Applying the rotation from the reference to the body system to a vector $\mathbf{x}$ then means performing the following matrix-vector multiplication:

$$\mathbf{x}^{\mathrm{b}} = \mathbf{R}\mathbf{x} \tag{3.2}$$

The DCM does not suffer from the singularity cases as Euler angles do; it represents rotation redundantly, in case of 3D using 9 coefficients to express 3 degrees of freedom. Performing mathematic operations on DCM introduces accumulation of numeric errors which, because of this redundancy, leads to deformations of the matrix. It then no longer represents a rotation "cleanly", i.e. loses orthogonality and unit determinant. For this reason, re-normalization is an important step of the DCM-based algorithms [21].

Quaternions A four-dimensional hypercomplex number – extension to the concept of complex numbers – composed of *real part* and *imaginary part* that uses three *quaternion units* i, j, k instead of a single imaginary unit:

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{3.3}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{3.4}$$

Similarly to rotation matrices, quaternions overcome the gimbal lock problem by a redundancy – the one degree of freedom, superfluous to represent a 3D rotation, is removed by introducing unit norm [18]:

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{3.5}$$

Such a unit quaternion is then often called a *rotation quaternion* or a *versor*; in this thesis, quaternions are used solely to represent rotation, so that all these terms may be used interchangeably.

There are equations and computation algorithms available for all operations quaternion usage as an orientation descriptor needs [22, 23]. There are also formulae converting quaternion notation to DCM or Euler/Tait–Bryan angles [19]:

$$\mathbf{R}(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2q_3 + 2q_0q_1 \\ 2q_1q_3 + 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{3.6}$$

$$\begin{bmatrix} \phi_{xyz} \\ \theta_{xyz} \\ \phi_{xyz} \end{bmatrix} = \begin{bmatrix} \operatorname{arctan2}(2q_2q_3 + 2q_0q_1, q_3^2 - q_2^2 - q_1^2 + q_0^2) \\ -\arcsin(2q_1q_3 - 2q_0q_2) \\ \operatorname{arctan2}(2q_1q_2 + 2q_0q_3, q_0^2 + q_1^2 - q_2^2 - q_3^2) \end{bmatrix} \tag{3.7}$$

In many practical respects, quaternions can be likened to the rotation matrices – not suffering from singularity nor discontinuities, composite rotations are available, and re-normalization is needed to recover rounding errors. However, all these operations are simpler in case of quaternions, not only due to the more compact representation, but also thanks to different internal structure (re-orthogonalization not needed).

3.1.3 Global localization – GNSS

Global Navigation Satellite System (GNSS) is a common designation for systems capable of outputting Earth-referenced receiver position estimate based on processing of radio signals transmitted by a constellation of satellites (orbiting mostly roughly 20 000 km above sea

level). There are four GNSS operational currently: the well-known GPS established by the United States, European Galileo system, Russian GLONASS and Chinese BeiDou-2 (BDS).

Although there are technology differences among the different GNSS networks[2], the principle is similar enough on the level needed for the following evaluation. Each satellite carries precise atomic clock that is furthermore synchronized among the whole satellite network; it transmits the precise time together with information about the satellite's current orbital position.

By combining pseudorange measurements from multiple satellites, the navigation equation system is formed with four unknown parameters: receiver location in three dimensions and clock offset[3] [25]. To solve the system and obtain a so-called 3D fix, data from four satellites are needed – in such case the system of equations has a single solution. When there are more satellites in view, the systems becomes overdetermined and needs to be solved using one of the optimization methods; it, however, generally helps compensating for individual pseudorange measurement errors.

Error sources

To evaluate the limitations and consequent usability of GNSS as a pedestrian localization sensory input, it is important to understand its error sources. The majority of them are inherent to operating principle, i.e. radio signals traveling from the satellite through layers of atmosphere of the Earth towards receiver, and are listed in the following paragraphs [26].

Atmospheric effects Individual satellites within the receiver's line of sight are generally visible in different angles of elevations above horizon, i.e. their radio signals are traveling through various thickness of the atmosphere's layers.

Ionospheric delay is a frequency-dependent delay caused by charged particles of higher layers of atmosphere. Because of this dependency, dual-frequency receivers are able to remove ionosphere influence almost completely. For single-frequency users, there are compensation models available, predicting the effects of ionosphere: the Klobuchar Model [27] used in GPS with roughly 50% delay reduction, and the Galileo NeQuick-G ionospheric electron density model which performs slightly better [28].

Tropospheric delay does not depend on frequency and therefore cannot be compensated by dual-frequency measurements as the ionospheric delays [29]. There are two contributors: the predictable hydrostatic component delay and fast varying wet component delay. While the former one can be modeled and its error contribution (in the order of meters) successfully removed, the latter is hard to compensate and remove its delay in the range of decimeters.

[2] A great source of in-depth information on various GNSS-related topics is established by the ESA-founded Navipedia [24]

[3] Since the satellites synchronize their clocks, the receiver's clock offset is common to all satellites and can thus be distinguished in the pseudorange measurement.

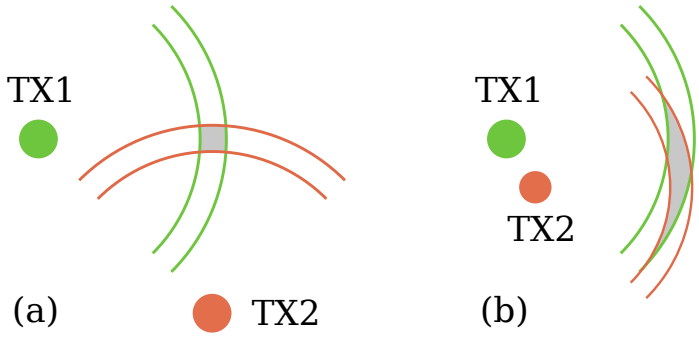


Figure 3.2: Dilution of Precision illustration; (a) favorable and (b) inferior arrangement.

DOP In the realm of the global navigation satellite systems, the predicted position estimation quality is quantified using the Dilution of Precision (DOP) values [30]. The concept has been introduced to express the influence of various satellite position configurations on the achievable state estimate; DOP is a scaling factor of positioning variance to measurement variance.

In Figure 3.2, geometric configuration influence on position estimate precision is illustrated on a simple 2D case. In 3.2 (a), the transmitters are in a favorable position to the receiver. In 3.2 (b), both transmitters are oriented to the receiver in a similar direction which leads to larger uncertainty.

The main descriptor is the Geometric Dilution of Precision (GDOP) devised from variances of the east, north and up components of the position estimate and variance of the receiver clock offset estimate. This figure can be broken down into Position DOP and Time DOP; PDOP is further composed of Horizontal DOP and Vertical DOP.

Receiver errors Mostly related to cost-sensitive applications, the user segment may play a significant role among the error sources. The apparent factors include antenna quality and receiver front-end sensitivity and noise figure; by improving these parameters one can expect better reception of weak satellite signals, either obstructed by foliage or in unfavorable orbital position with regard to the receiver.

The other source of errors on the end of the chain is the actual computational processing of the measured pseudoranges – solution of the navigation equations. Lack of numerical accuracy leads to quantization errors; for instance, 8 significant figures for latitude and 9 for longitude of the Navilock NL-402U leads to a quantization grid of 14 cm $\times$ 18 cm [31]. Employment of a filter with kinematic model, processing the quantized GNSS data, is commonly able to moderate the quantization errors impact.

Environment influence In addition to limiting the direct line of sight ideal for weak GNSS signal reception, tall buildings in urban areas ("urban canyons") also introduce the second source of signal degradation – reflections [32]. The mitigation techniques include signal quality aids (choke ring antenna), signal processing techniques and behavior patterns

(avoiding tall buildings). Recently, with the cloud computing advance, there have been attempts of ray tracing techniques utilization in prediction of the multipath cases – using 3D city maps and actual satellite orbital positions, one can calculate path of the signal to an estimated receiver position [33, 34]

Usability assessment

For the pedestrian application involving visually impaired user, the usability of a sensory data source is not only given by its mean performance, but rather by the worst-case behavior that may even relate to user's safety.

The pedestrian application implies that only simpler wearable GNSS receivers are usable. Advanced techniques as Real-Time Kinematics (RTK), using a reference receiver on known location, real-time wireless point-to-point and considerable computation efforts, are not suitable either for their logistic inflexibility and poor scalability.

In terms of the basic service accessible to general public, there is a promise of Galileo GNSS accuracy in the range of 1 m once the Galileo satellite network is finished and fully operational. There is, however, a legitimate doubt that this can be met in the demanding urban conditions.

GNSS localization data are for these reasons safe to be regarded approximate only; there need to be other sensory means in the system able to improve the pose estimates and reduce the worst-case error magnitude.

3.1.4 Inertial measurements usage

In an ideal world, inertial measurements could be usable for relative tracking of pose changes – simply by integrating the measured quantities (acceleration and angular velocity) once or twice to obtain relative pose over time. In reality, however, measurement errors with non-zero mean value (i.e. evincing a bias) lead to divergence in integrated quantities – this is even amplified by the dual integration needed to process accelerometer signals into translation.

Without a compensation based on an external quantity, every inertial measurement would diverge eventually. There are applications though whose requirements can be met with proper selection of sensor technology; for example, tactical-class sensors achieving gyro bias of $< 10\,°/h$ can be a sufficient choice for an air-to-air missile with expected operation lifetime measured in minutes. Or, a navigation-class system with bias $< 0.01\,°/h$ can fit requirements of a passenger plane or a cruise missile with flight times in range of hours [35].

Such sensory system classes are, however, not applicable to the studied class of problems: both for excessive size/weight and financial exclusivity. Only the commercial-grade inertial sensors, usually designed as MEMS[4], are within reach for this application. Their inherent

[4]Micro Electro-Mechanical System

restrictions need to be considered design parameters and bypassed by selecting a favorable measurement arrangement.

There are two basic applications a MEMS inertial measurement unit (IMU) can be successfully employed for – AHRS and PDR.

Attitude and heading reference systems (AHRS)

The first approach of reducing drift impact on integrated IMU data is actually limiting integration to the bare minimum. Instead of producing pose change estimates, the AHRS only works with the angular part – attitude.

In its simplest form, a 3D accelerometer can be used to estimate *roll* and *pitch* by determining the vector of local acceleration of free fall; of course, this only works faithfully in static scenarios, as additional forces cannot be discerned from the weight vector. Adding a gyroscope to the setup extends the usefulness to dynamic scenarios as well – it also permits estimation of *locally-referenced yaw* by integration of the measured angular velocity. Such IMU configuration is often referred to as 6-DOF.

In case the application needs to estimate *yaw* that is globally referenced, the sensory equipment needs to be augmented again; in majority of cases, a 3D magnetometer is used, leading to a 9-DOF IMU configuration [36]. Earth's magnetic field is a well-known source of heading reference, for which the processing methodology has been, obviously, known for a long time (field strength models, magnetic declination correction etc.). In the urban environment, however, the conditions are not ideal for this measurement type – massive structures from materials influencing magnetic field are practically inevitable; similarly, wearable platform contains metallic parts near to the sensor. The influence can be classified as a hard- or soft-iron effect; hard iron affects the readings as a direction-independent additive error, while soft-iron distortions are not constant and have to be compensated by a complex calibration [37].

The problem of AHRS data fusion is then one of the most often to be solved nowadays, presumably due to widespread adoption of small to mid-sized autonomous aerial vehicles (drones) – nearly all of them are equipped with flight stabilization systems depending on fast attitude estimation. For this reason, there is a multitude of fusion approaches, ranging from a simple complementary filter to advanced variants of Bayesian filters. An excellent write-up comparing the different approaches can be found in [38].

Pedestrian dead reckoning (PDR)

The mechanism enabling drastic reduction of IMU drift is provided inherently by the human bipedal locomotion. Zero-Velocity Update (ZUPT) takes advantage of the stationary phase of a stride during which the foot is in contact with the ground; velocities integrated from IMU accelerometry data can be zeroed at such moment, limiting unbounded divergence of biased double integration.

The only mounting place providing real zero-velocity phase are feet [39, 40, 41]. Approaches of mounting the IMU to body parts other than feet can be found in the literature as well, using periodic nature of pedestrian motion without the real zero-velocity phase – the average relative position errors, however, tend to be an order of magnitude worse in case of such solutions [42, 43, 44]. The majority of humans also dispose of two feet, which can be utilized to mount a couple of PDR units – the improvements achieved by dual-foot configuration cooperation have been described in [45].

The major imperfection of the step-wise PDR is residual drift in attitude; when the PDR system is used without correction, attitude gradually drifts which deforms the integrated trajectory shape. Although there are some infrastructure-free drift limiting techniques employing constraints of human motion other than the zero-velocity phase [46], globally referenced correction mechanisms are generally needed to prevent the errors grow unbounded in the long term. Usage of magnetic field has been explored by [47] with moderately good results in favorable conditions, but the non-Gaussian, impulse character of the disturbances is hard to account for in standard filter frameworks and can bring more detriment than benefit.

Usability assessment

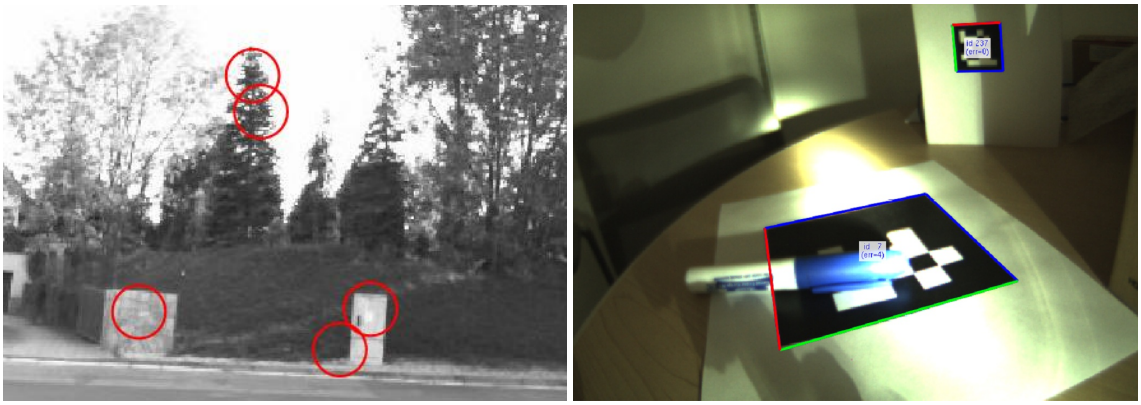
Meaningful utilization in the proposed pedestrian navigation system can be expected from both above mentioned IMU usage paradigms: AHRS is a vital component of environment-observing subsystems, while PDR contribution can improve local trajectory features estimation. In the urban environment, provisions need to be taken to prepare for deviations in AHRS-determined globally referenced yaw though: as outlined above, the magnetic field is likely to be distorted by the environment and by the wearable platform itself.

3.1.5 Environmental observations

Landmarks are distinct, stationary features of the environment that can be recognized reliably [16]. Based on their origin they can be divided into two basic groups – natural and artificial landmarks, as illustrated in Fig. 3.3. While natural landmarks are constituted by the environment itself and its interaction with light, artificial ones see their employment mainly in controlled environments; their preparation and maintenance is sensible in limited scopes only.

Natural landmarks

Natural landmarks are formed by environment's inherent features – in the most strict form only by nature's inanimate and living objects such as rocks, trees, skyline and their combinations. Logically, also *urban* landmarks belong into a wider understanding of this group – distinct features of the architecture and other signs of human activity – since they are not primarily designated as landmarks (as opposed to artificial ones below). Both of



(a) (b)

Figure 3.3: Examples of detected landmarks in camera image: (a) natural and urban; (b) artificial [48].

these are of interest for outdoor localization in uncontrolled environments, i.e. without the need or even chance to prepare the surrounding in a way that would simplify orientation in it.

The major perception method usable for landmark detection is obviously (computer) vision; there are three steps in which the image processing progresses [49], in principle applicable to all below mentioned methods, even to the artificial landmarks case[5]:

- *Detection* – finding the distinctive points of interest; usually, detectors work with high-contrast intensity patterns of gray-scale image and are based on Difference of Gaussian[6] (DoG, used by SIFT), Determinant of Hessian (DoH, SURF) or corner detection (FAST [51]).
- *Description* – assignment of a feature vector representing the particular point. Details strongly vary with each method, but the aim is common – be able to distinguish among the points with as much reliability as possible while exhibiting sensible storage and computational requirements.
- *Matching* – finding pairs of corresponding features in subsequent images.

SIFT *Scale-invariant feature transform*, introduced in [52], was one of the first "packages" of standardized feature detection and extraction functionality. It has been patented by the author, which makes commercial usage quite pricey and also limits support in FOSS libraries.

[5]Each method also contains proper image preprocessing pipeline [50], so that additional preprocessing is normally not needed – except for rescaling the source image, which determines the physical detail level on which the features will be extracted.

[6]Mostly as a resource-friendlier approximation of the Laplacian of Gaussian (LoG). The functional principle is essentially a band-pass filter preserving only some of the spatial frequencies, usually edges.

The detection step is based on DoG as already noted; the descriptors are composed of 128 elements and determined by binning 8 major directions of feature neighborhood gradient samples of 4×4 px size in 4×4 descriptor arrangement [53].

Among the real-number descriptor methods, SIFT is considered the most faithful, i.e. providing best feature matches. This is "compensated" by relatively high computational demands of the method though.

SURF *Speeded up robust features* was keyed in [54] and is possibly the most popular general image feature detector nowadays. Similarly to SIFT, it is scale invariant – and additionally also rotation invariant. However, it is also patented and not cheap for commercial usage.

Detection step is designed as DoH with second-order Gaussian derivatives approximated by box filters, improving the performance over SIFT [55]. Each feature's descriptor based on Haar wavelet responses in circular neighborhood around the detected feature is composed of 64 items given by 4×4 sub-regions (denoted as SURF-64 or just SURF since it is the most often case), 36 items for 3×3 sub-regions (SURF-36) or even 128 items for 4×4 sub-regions with extended descriptors (SURF-128) [54].

Overall, SURF has been shown to evince a better compromise between faithful matching and performance than SIFT [55], given by its order of magnitude faster execution and slightly lower quality of feature matches. In applications featuring translation-only transformation among the images, the Upright SURF (U-SURF, not invariant to feature rotation) flavor may bring further improvements in performance and also improve match success rate (by reducing false match probability).

Other options There is quite a wide range of different solutions with varying parameters: e.g. LoG-based detectors, corner detectors (Harris–Stephens, FAST); GLOH, HOG, FREAK descriptors. Also, different descriptors can be used with SURF than the SURF descriptor – combinations of SURF detector and SIFT or BRIEF descriptor can be found applied with the motivation of improving descriptor quality with compromise performance [56].

Recently, most promising is the binary-descriptor-based BRISK method [57], which outperforms SURF with roughly order of magnitude lower computational costs – and with comparable quality as SURF; importantly, it comes without patent restrictions. However, although binary descriptors may work well for initial tracking of landmarks in the mapping phase, subsequent matching phase is more likely to work well with real-number descriptors as used by SIFT or SURF.

Feature matching For real-number descriptors, the most commonly nearest neighbor: given two sets of features A and B, $b \in B$ is called the nearest neighbor of $a \in A$ if $b = \mathrm{argmin}_{b' \in B} \|a - b'\|$ (i.e. using Euclidean distance) [56]. The criterion of acceptation

or rejection of the feature pair as matched is the relative distance to the second nearest neighbor; if the second one is sufficiently *worse* a match than the winner, the two features are pronounced matched. The relative distance is then one of the parameters of the matching process: value of 0.8 is proposed by [53] for SIFT, discarding 90 % of false matches together with less than 5 % correct matches. SURF authors in [54] use 0.7 distance for the nearest neighbor method. Besides the nearest neighbor method, similarity threshold matching is sometimes preferred, especially for matching on larger feature databases: it only compares the computed distance $\|a - b'\|$ to a threshold and potentially returns multiple match pairs [58].

Binary descriptors, on the other hand, use Hamming distance which is computationally advantageous – consists of applying the *XOR* operation to descriptors and summing the resulting number of bits set to 1, i.e. the whole distance computation of two feature descriptors can be accomplished by two efficient steps[7].

Localization usage There are multiple documented approaches on utilization of environment visual features for localization purposes. The general direction can be seen toward the Simultaneous Localization And Mapping (SLAM) mechanisms building from huge quantities of identified landmarks and resulting in significantly sized data representations of even relatively small maps.

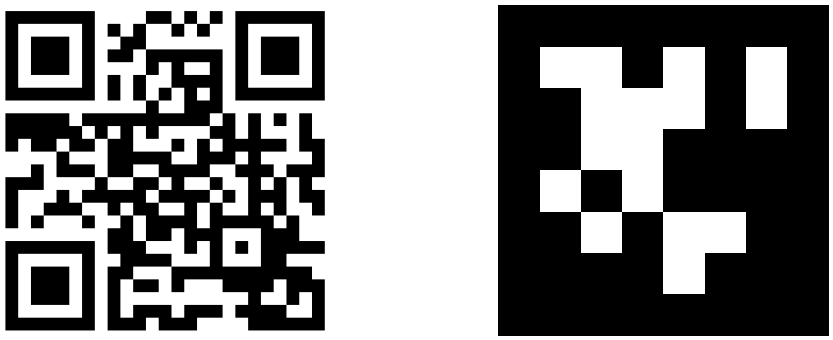
An indoor case is given in [59], featuring a robot with a stereovision system; the map is composed based on SIFT features detected in the stereo images and thus inherently localized in space. Loop closure working with sub-maps is employed to guarantee consistent map building; map-based localization is then proposed in two approaches, either using Hough transform applied to candidate poses gained from landmark matches, or alternatively employing a RANSAC-based optimization method to select the best pose hypothesis.

In [60], a topological outdoor map building process is proposed, again utilizing the SIFT detector/descriptor and association of extracted features with poses from which they are visible. The localization is then based on weighted linear combination of nearest neighbors' pose determined for landmarks found in image under investigation and matched to the map. Big importance is attributed to proper landmark selection during the mapping phase.

An interesting view on the topic is presented by [61], which describes map building based on a large number of geo-tagged photos from social networks depicting the same object (building, statue etc.). Bundle adjustment is employed to integrate all snapshots of each object, together with inherent correction of the geo tags.

Triangulation as a method of either landmark gathering or subsequent localization is published in several literature sources, each dealing with a slightly different aspect of it [62, 63, 64, 65]. Similarly to the DOP figures introduced above for GNSS, precision dilution is present for such triangulation methods as well; an analysis is given in [66].

[7] On SSE4-enabled CPUs, bits set to one can be counted by a specialized instruction `POPCNT` with constant cycle latency which is very efficient compared to Euclidean distance computation.



(a) (b)

Figure 3.4: (a) general-purpose QR Code tag; (b) specialized fiducial mark AprilTag.

Artificial landmarks

Artificial landmarks are, as opposed to natural landmarks, placed into the environment specifically with the intention of creating points of orientation, designed to be easily detectable and distinguishable by visual or other sensors. Compared to natural landmarks extracted by means of SIFT/SURF/alike, the uniqueness and positive matching of an artificial landmark can be trusted – contained error correction mechanisms and/or coding schemes effectively prevent misunderstanding.

In the context of this thesis we are only interested in visual (camera sensed) fiducial marks. An implicit choice would be some of the wide range of 2D codes, that started emerging in 1990s and allowed to contain much more information than its predecessor, the 1D barcode. Among the most used types we can find the Quick Response Code (QR Code, Fig. 3.4a), SnapTag, MaxiCode, DataMatrix etc., each providing a slightly different set of features (dot resolution, data capacity, robustness etc.). These types of visual coding see its utilization in robotics – mostly in cases when machine identification of manufactured parts, supply crates or simply other tracked items is needed. Practically all of these scenarios, however, typically rely on controlled sensing conditions – since these codes have relatively complex structure (to contain the information payload), they need to be sensed from rather shorter distance, with sufficient resolution etc.

On the other hand, there are fiducial systems created specifically for ease of detection at distance and extremely small false positive rate; we can find several packages among the representatives of this category – starting with the ARToolkit [67], through ARTag [68] or the most recent AprilTag / AprilTag 2 [48, 69]. As can be seen from the lower complexity in Fig. 3.4b, robust detection at distance in turn limits the type and quantity of information contained to an integer identifier with a couple of thousand unique codes (for the most usual configuration of 6×6 "pixels").

Usability assessment

For the purposes of this thesis, artificial landmarks can be beneficial for creating calibration scenarios or similar tasks. Outdoor utilization of the feature detection/extraction methods and usage of the surrounding environment as the geo-reference inevitably brings one question: how robust is it in terms of short- and long-term repeatability? The answer is severalfold:

- *Short term* repeatability does not suffer from transformations of the environment; the only practical exception are cars and other transportation customary in urban environments. Major influencing factors are lighting conditions – varying brightness of particular surfaces may change the scene significantly.
- *24-hour cycle* is a special case. Conditions of camera-based recognition change from impossible (night) through difficult (gloomy light or on the contrary sharp light blinding the cameras) to advantageous.
- *Performance among seasons* – strictly urban landscape may be affected in a lesser extent by spring and fall (since buildings would generate most of the extracted landmarks), but snow changes perception of generally all kinds of outdoor environment. [49] compares SIFT, SURF, U-SURF and SURF-128 among all four seasons on a representative mixed environment. Generally, the results are favorable, with majority of cases evincing 60 % and more correct matches among all seasons.

Overall, while inherently limited by the daily light cycle, the usefulness of natural features coupled with suitable processing methods is obvious. The limitation to bright part of the day may seem severe; however, the wearable platform constraints discussed above do not permit employment of many better-equipped sensing principles.

3.1.6 Map utilization

One of the important information sources related to the localization task are obviously maps. The usual sense of human-readable map as a two-dimensional plot of path segments and other objects is a subset of the functionality a map can provide – this may range to capturing of complex topological structures, feature descriptors and other data resulting from the mapping process. In this sense, a proper feature map is necessary for the landmarks utilization discussed above.

In terms of map as a graph structure of inter-connected path segments, there are two possible uses:

- *Path planning* is the obvious subsystem relying on segment maps; Chapter 4 deals with this topic thoroughly.

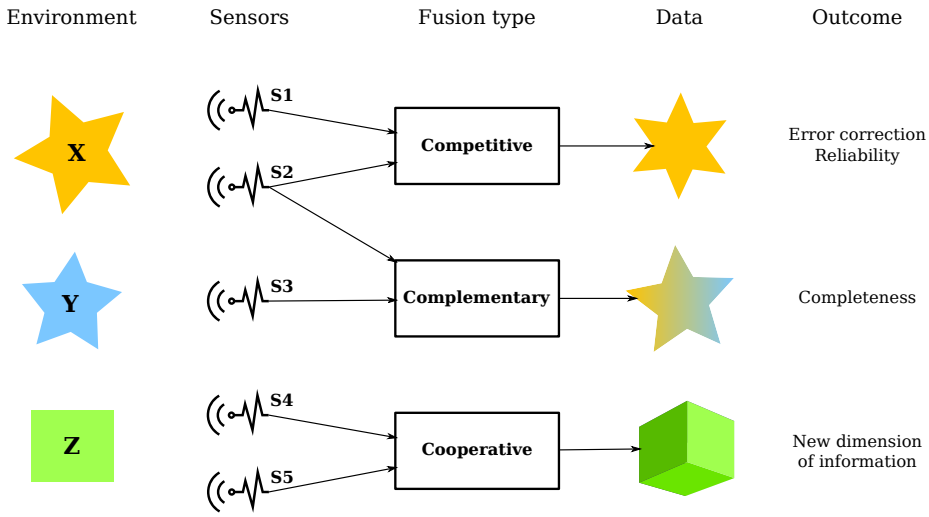


Figure 3.5: Notion of competitive, complementary and cooperative data fusion.

- *Localization* may benefit from an information about environment arrangement that the pedestrian should conform to as well, see the discussion below this paragraph.

Using a segment map to increase probability of pedestrian's position situated within sensible parts of the map (i.e. primarily sidewalks and other pedestrian-designated areas) may seem tempting – after all, sticking to current/nearest road is the strategy most of the car satellite navigation devices apply to overcome the GNSS position inaccuracy. However, there are several reasons why it may not be advisable to be used in the pedestrian case, at least not restricted to sidewalks alone[8]. First and foremost, it may more often lead the system to a false belief that the user is still on sidewalk, while in fact being on road or in other potentially dangerous area – which may be catastrophic in case of a visually impaired user. Second, a pedestrian can inherently traverse wider range of surface types, so that almost all areas neighboring the sidewalk would have to be given enough probability, limiting impacts of the method.

3.1.7 Fusing data from multiple sources

The analytical section has so far introduced a range of sensory data sources usable for the pedestrian localization. An important task of creating a pose estimate based on them naturally arises: in an optimal case, only sensible contributions converging the estimate to the ground truth would be extracted from the noisy observations.

A basic introduction to data fusion is given in [70], including the overview classification below; the topics are also comprehensibly and in-depth elaborated in the mobile robotics sources as [16], [18] or [17].

[8]This strategy would be otherwise elegantly implementable within the particle filter framework, by generation of new particles preferably on segment.

- *Competitive fusion.* Whenever multiple sensors observe the same state, they will in real-world conditions often come to different results – a competition or mutual correction based one the redundant observation may then drastically improve behavior with corrupted signal. This thesis describes several competitive mechanisms, including the cases of GNSS + PDR position estimation or PDR + AHRS heading estimation.

- *Cooperative fusion.* Fusion of otherwise independent observations of different features often brings more benefit than using the observations separately. An example can be found in fusion of GNSS + AHRS or using multiple landmarks found in camera image for position triangulation.

- *Complementary fusion.* Using independent sensors observing different parts of the environment is probably the easiest of the tasks. The multi-camera setup of the wearable platform, as described in Chapter 5, is an illustrative case of such approach.

Key principle in the sensor fusion endeavor is the notion of *uncertainty* – accepting that all data gained from real world are inherently inaccurate due to many non-predictable processes occurring on the path along the observation. In the realm of mobile robotics, the approach of *probabilistic robotics* has been keyed by [16], building on employment of mathematical apparatus supporting the probability theory.

Bayes theorem can be found at the foundations of vast majority of the probabilistic mechanisms utilized by this thesis. It is in its discrete form provided on the last line below and is based on the preceding formulas of *conditional probability* $p(x|y) = p(X = x|Y = y)$ and *total probability* $p(x)$ [18], where X is a random variable with value of x. The main notion of the Bayes theorem is its ability to express the conditional probability $p(x|y)$ by its "inverse" $p(y|x)$ – for example, by knowing the probability of sensor reading y observing feature x (which can be usually determined for each sensor), we can then infer x from sensor data y:

$$p(x|y) = \frac{p(x,y)}{p(y)} \tag{3.8a}$$

$$p(x) = \sum_{y} p(x|y)p(y) \tag{3.8b}$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \tag{3.8c}$$

Normal (Gaussian) distribution is then one of the most useful probability distributions that a certain class of random variables can be described with. In plays an important role in probabilistic principles – according to the *central limit theorem*, normalized sum of independent random variables tends toward normal distribution, although the variables may not be normally distributed [71]. The consequence is then good applicability of statistical

tools devised for normal distribution also to problems where the distribution is not well known or differs from the normal distribution. Random variable conforming to normal distribution X with value of x, mean μ and variance σ^2 is often denoted as $\mathcal{N}(x; \mu, \sigma^2)$. Its probability density function (PDF) for one dimension is then given by the Gaussian "bell curve" function:

$$p(x) = p(X = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.9}$$

Recursive state estimation

The two mechanisms introduced below, Kalman filter and Particle filter, both fall into the category of recursive probabilistic state estimators. They build upon system state description in the following form (for a linear system):

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{v}_k \tag{3.10a}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{D}_k\mathbf{u}_k + \mathbf{w}_k \tag{3.10b}$$

where $\mathbf{x}_k$ denotes state vector, $\mathbf{u}_k$ is input vector and $\mathbf{y}_k$ is output vector. Noise vectors $\mathbf{v}_k$ and $\mathbf{w}_k$ introduce the process and measurement noise – typically, Gaussian distribution with zero mean is presumed: $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{V}_k)$, $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{W}_k)$.

Since most of the useful systems evince non-linear behavior, more practical (or even necessary) is to assume the system state model in a different form:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{v}_k \tag{3.11a}$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, k) + \mathbf{w}_k \tag{3.11b}$$

A frequent simplification of the description both in linear and non-linear domain is removal of the direct link from input to output, i.e. omission of term $\mathbf{D}_k\mathbf{u}_k$ of Eq. 3.10b and parameter $\mathbf{u}_k$ of function h from Eq. 3.11b; the text below will do so as well. If needed, such linkage can be expressed by a suitable state vector extension.

The merit of this arrangement is that although the state vector $\mathbf{x}_k$ items often cannot be measured directly, they may be *observable* from system output $\mathbf{y}_k$ [72]. Recursive state estimation is a mechanism used in great many application dealing with uncertain inputs; out of the wider range of Bayesian methods, Kalman filter and Particle filter frameworks have been selected for a performance comparison and will be briefly introduced below. Dear reader is encouraged to find complete derivation of the briefed methods in the cited sources.

Kalman filtering

First presented by [73] in 1960, the Kalman filter was conceived and applied to help solving trajectory estimation problems for the Apollo program. Since then, it and its non-linear flavors have dominated in various state estimation and data fusion employments in practically all branches of engineering.

The classic Kalman filter is introduced for linear systems with structure according to Eq. 3.10 (without the term $\mathbf{D}_k\mathbf{u}_k$ as discussed). Together with its non-linear variants focused on below, the whole Kalman filter family shares the representation of belief and uncertainty in system state: normally distributed variables with mean value μ and variance σ^2, or, considering the whole state vector, the mean vector $\boldsymbol{\mu}$ and its *covariance matrix* $\mathbf{P}$[9]. This parametric arrangement means that the filter can only have a single hypothesis on the system state, i.e. is *unimodal*, which has important consequences in real-world applications.

Generally, all Kalman filter variants work in two stages; the linear Kalman filter equations are used here for illustration purposes due to their simplicity, even though the problems solved by this theses need a non-linear estimator.

Prediction is the first step in which the dynamic model is used to progress the system to state $\hat{\mathbf{x}}_{k+1|k}$ based on previous state $\hat{\mathbf{x}}_{k|k}$ and control input $\mathbf{u}_k$. System covariance expressed by the matrix $\mathbf{P}_{k+1|k}$ is *increased* during prediction due to process noise $\mathbf{V}_k$:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k\hat{\mathbf{x}}_{k|k} + \mathbf{B}_k\mathbf{u}_k \tag{3.12a}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^\top + \mathbf{V}_k \tag{3.12b}$$

Correction then typically incorporates measurement updates $\mathbf{y}_{k+1}$ with their noise matrix $\mathbf{W}_{k+1}$ that allow convergence of modeled system state towards the observed system. It is done through measurement innovation $\tilde{\mathbf{y}}_{k+1}$ "transformed" by Kalman gain matrix $\mathbf{K}_{k+1}$; the system covariance $\mathbf{P}_{k+1|k+1}$ is here *decreased*, confirming the intuitive understanding that measurement has improved our knowledge of the true system state:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\tilde{\mathbf{y}}_{k+1} \tag{3.13a}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\mathbf{P}_{k+1|k} \tag{3.13b}$$

where measurement innovation $\tilde{\mathbf{y}}_{k+1} = \mathbf{y}_{k+1} - \mathbf{C}_{k+1}\hat{\mathbf{x}}_{k+1|k}$, correction covariance matrix $\mathbf{S}_{k+1} = \mathbf{C}_{k+1}\mathbf{P}_{k+1|k}\mathbf{C}_{k+1}^\top + \mathbf{W}_{k+1}$ and finally Kalman gain $\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{C}_{k+1}^\top\mathbf{S}_{k+1}^{-1}$.

[9] The covariance matrix is a square symmetric semidefinite matrix with principal diagonal elements expressing the respective state variable variance, while the other elements form covariances among the state variable.

Extended Kalman filter The straight-forward attempt to apply Kalman filtering to non-linear systems is formed by the extended form (EKF) – the non-linear functions f and h from Eq. 3.11 are replaced by linearized Jacobian matrices $\mathbf{F}$ and $\mathbf{H}$ [74]:

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \tag{3.14a}$$

$$\mathbf{H}_{k+1} = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \tag{3.14b}$$

The rest of the Kalman filter stays identical, only using the Jacobian matrices $\mathbf{F}$ and $\mathbf{H}$ instead of the original linear matrices $\mathbf{A}$ and $\mathbf{C}$.

Among advantages of EKF is the structural similarity to plain KF; out of the non-linear flavors, it is also one of the most computationally efficient. The imperfections may be prevalent though: the Jacobian matrices are often non-trivial to derive and unstable in performance[10]. Divergence issues for range and bearing type of sensor data are discussed in [76], comparing EKF with UKF and other filters; it has been found that, although being under circumstances also prone to divergence, UKF performs better.

Unscented Kalman filter The Unscented Kalman Filter (UKF) belongs to the Sigma-point Kalman Filters family and has been proposed to mitigate limitations of the EKF given by its main principle, Jacobian-based linearization. The principle responsible for the non-linear probability distributions processing is the Unscented Transform (UT), keyed together with UKF in [77, 78, 79].

The main notion of UT can be perceived as lying half-way between the classical KF (working on all levels with a mean and (co)variance) and Monte Carlo style of computation (using a big quantity of independent points to represent the probability distribution); UT deterministically selects a set of so-called sigma points to express the mean value and covariance of system state, which is both given by the points' position in the state space. This elegantly enables processing of the state distribution by a non-linear function which generally transforms the distribution into a different one; mean and covariance measures are determined from the processed points afterward.

As illustrated by Fig. 3.6 working with the demonstrative "banana" shape inherent to the distance & heading style of models, UKF/UT estimation method converges to the "true" mean and covariance established by Monte Carlo much better than the EKF approach.

Initial estimate of the system state $\hat{\mathbf{x}}_0$ and its covariance $\mathbf{P}_0$ is determined based on best available a-priori knowledge as the first step at the beginning of the state estimation process. System state representation may, besides its main "payload", contains also helper

[10] The performance is inferior mainly near pronounced non-linearities, where the first-order Taylor expansion cannot faithfully approximate the original function. EKF variations with second-order Taylor expansion through Hessian matrix [75]. Still, UKF is found a better non-liner estimator.

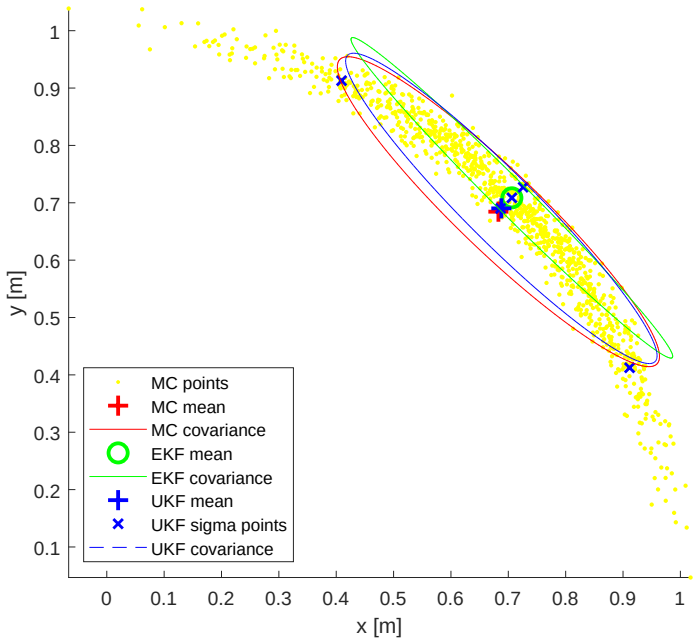


Figure 3.6: Comparison of mean and covariance estimation through 1000-samples Monte Carlo as ground truth, EKF and UKF mechanisms. The non-linear function is a simple distance/bearing beacon model given by $[x; y] = [r\cos(\phi); r\sin(\phi)]$; true covariance of the r and ϕ inputs is $P = [0.0004\,0; 0\,0.068]$. Covariance ellipses denote 68% confidence interval.

items – for UKF, the technique of state vector *augmentation* is particularly suitable. In such case, the augmented state $\hat{\mathbf{x}}^a_{k|k}$ holds also mean values of process and measurement noises and the augmented system covariance matrix receives the process and measurement noise matrices:

$$\hat{\mathbf{x}}^a_{k|k} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ [\underbrace{0\,0\,\cdots\,0}_{n \text{ elements}}]^\top \\ [\underbrace{0\,0\,\cdots\,0}_{m \text{ elements}}]^\top \end{bmatrix} \tag{3.15}$$

$$\mathbf{P}^a_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k} & 0 & 0 \\ 0 & \mathbf{V}_k & 0 \\ 0 & 0 & \mathbf{W}_k \end{bmatrix} \tag{3.16}$$

It is apparent that the noise mean values are zero in accordance with Gaussian noise characteristic assumptions introduced above. State augmentation then leads to two beneficial consequences:

- the number of sigma points is increased, allowing to approximate the state probability distribution more faithfully,
- the sigma point "spread" around the central is driven not only by the system state covariance, but also by the process and measurement covariances, allowing the estimator to react to the instantaneous uncertainty levels and position the sigma points properly.

The recursive estimation runs iteratively in steps described below:

1. Sigma points generation is based on the initial state estimate $\hat{\mathbf{x}}_{k|k}$ and its covariance matrix $\mathbf{P}_{k|k}$ which determines how far the other $2n$ sigma points are spread from the central one. At the same time, weight constants $W_i^{(m)}$ for mean calculations and $W_i^{(m)}$ for covariance calculations are established:

$$\boldsymbol{\mathcal{X}}_{k|k} = \left[\hat{\mathbf{x}}_{k|k}, \hat{\mathbf{x}}_{k|k} \pm \sqrt{(n+\lambda)\,\mathbf{P}_{k|k}}\right] \tag{3.17}$$

$$W_i^{(m)} = \begin{cases} \frac{\lambda}{(n+\lambda)} & \text{when i} = 0 \\ \frac{1}{2(n+\lambda)} & \text{when i} = 1 \ldots 2\text{n} \end{cases} \tag{3.18}$$

$$W_i^{(c)} = \begin{cases} \frac{\lambda}{(n+\lambda)} + (1 - \alpha^2 + \beta) & \text{when i} = 0 \\ \frac{1}{2(n+\lambda)} & \text{when i} = 1 \ldots 2\text{n} \end{cases} \tag{3.19}$$

where α, κ, β and $\lambda = \alpha^2(n+\kappa) - n$ are scaling parameters directing sigma points spread around the central point $\hat{\mathbf{x}}_{k|k}$ and $\sqrt{\mathbf{P}_{k|k}}$ is a matrix square root obtained by Cholesky decomposition from matrix $\mathbf{P}_{k|k}$.

2. Process function execution (motion model) f transforms the sigma points $\boldsymbol{\mathcal{X}}_{k|k}$ into the predicted sigma points $\overline{\boldsymbol{\mathcal{X}}}_{k+1|k}$ based on the control input $\mathbf{u}_k$:

$$\overline{\boldsymbol{\mathcal{X}}}_{k+1|k} = f\left(\boldsymbol{\mathcal{X}}_{k|k}, \mathbf{u}_k\right) \tag{3.20}$$

3. Predicted state vector $\hat{\mathbf{x}}_{k+1|k}$ is determined as weighted sum of the sigma points. Computation of its covariance $\mathbf{P}_{k+1|k}$ iterates the definition formula $P(X_i, X_j) = \mathrm{E}[(X_i - \mu_i)(X_j - \mu_j)]$ over all sigma points and weights the contributions similarly:

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n} W_i^{(m)} \overline{\boldsymbol{\mathcal{X}}}_{i,k+1|k} \tag{3.21}$$

$$\mathbf{P}_{k+1|k} = \sum_{i=0}^{2n} W_i^{(c)} \left(\overline{\boldsymbol{\mathcal{X}}}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k}\right)\left(\overline{\boldsymbol{\mathcal{X}}}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k}\right)^{\top} + \mathbf{V}_k \tag{3.22}$$

Equation (3.22) calculates the system covariance matrix $\mathbf{P}_{k+1|k}$ after running the sigma points through the process function f. In cases process noise $\mathbf{V}_k$ is significantly lower than the system state covariance, its contribution to the sigma points spread of the noise-augmented state vector is not pronounced and the process noise can be simply added to the covariance instead [80]. Otherwise, usage of the augmented state may be beneficial.

4. Re-generation of sigma points in case of the non-augmented state vector is on the other hand necessary to take the additional process noise $\mathbf{V}_k$ into account.

$$\boldsymbol{\mathcal{Z}}_{k+1|k} = \left[\hat{\mathbf{x}}_{k+1|k}, \hat{\mathbf{x}}_{k+1|k} \pm \sqrt{(n+\lambda)\,\mathbf{P}_{k+1|k}}\right] \tag{3.23}$$

5. Measurement model h can be executed, resulting in the output sigma points $\overline{\boldsymbol{\mathcal{Z}}}_{k+1|k}$:

$$\overline{\boldsymbol{\mathcal{Z}}}_{k+1|k} = h\left(\boldsymbol{\mathcal{Z}}_{k+1|k}\right) \tag{3.24}$$

6. Mean and covariance of output is determined analogically to Eq. 3.21 and Eq. 3.22. Similarly to $\mathbf{P}_{k+1|k}$, measurement noise $\mathbf{W}_{k+1}$ is simply added, with the same reasoning. Additionally, the state-output cross-covariance matrix $\mathbf{P}^{x,z}_{k+1|k}$ is quantified:

$$\hat{\mathbf{z}}_{k+1|k} = \sum_{i=0}^{2n} W_i^{(m)} \overline{\boldsymbol{\mathcal{Z}}}_{i,k+1|k} \tag{3.25}$$

$$\mathbf{S}_{k+1} = \sum_{i=0}^{2n} W_i^{(c)} \left(\overline{\boldsymbol{\mathcal{Z}}}_{i,k+1|k} - \hat{\mathbf{z}}_{k+1|k}\right) \left(\overline{\boldsymbol{\mathcal{Z}}}_{i,k+1|k} - \hat{\mathbf{z}}_{k+1|k}\right)^\top + \mathbf{W}_{k+1} \tag{3.26}$$

$$\mathbf{P}^{x,z}_{k+1|k} = \sum_{i=0}^{2n} W_i^{(c)} \left(\boldsymbol{\mathcal{X}}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k}\right) \left(\overline{\boldsymbol{\mathcal{Z}}}_{i,k+1|k} - \hat{\mathbf{z}}_{k+1|k}\right)^\top \tag{3.27}$$

7. Kalman gain is evaluated by using the state-output cross-covariance and output covariance matrices – $\mathbf{P}^{x,z}_{k+1|k}$ directs influence of each output to each state member, while $\mathbf{S}_{k+1}$ reciprocally weighs the quality of each output/measurement and thus determines the extent of correction:

$$\mathbf{K}_{k+1} = \mathbf{P}^{x,z}_{k+1|k} \mathbf{S}^{-1}_{k+1} \tag{3.28}$$

8. Correcting the predicted state estimate $\hat{\mathbf{x}}_{k+1|k}$ and its covariance $\mathbf{P}_{k+1|k}$ produces the final results of the *k+1*-th iteration – Kalman gain $\mathbf{K}_{k+1}$ is first used to weigh measurement innovation $\tilde{\mathbf{y}}_{k+1}$ applied to the predicted state vector and then, together with output covariance matrix, determines *decrease* of the system covariance due to the applied measurement correction.

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1} \overbrace{\left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}_{k+1|k}\right)}^{\tilde{\mathbf{y}}_{k+1}} \tag{3.29}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{S}_{k+1}\mathbf{K}_{k+1}^{\top} \tag{3.30}$$

It should be noted that the process model f can be used for prediction even if there is no correction measurement available at the moment – the steps 4–8 are then skipped. In such case, however, it is apparent that the system covariance can only grow, which is in agreement with the intuitive understanding that without a correction, the belief in the predicted state can only get weaker.

Particle filtering

Particle filtering is a Bayesian non-parametric technique capable of working with arbitrary probability distribution functions. The principle is tightly bound to the Monte Carlo method class, for which it is also referred to as the Monte Carlo Localization; the system state belief is represented by a set of discrete *particles*, each possessing a hypothesis on the system state. The main advantage over the Kalman filter family can be seen in the ability to represent *multimodal* distributions, which is often more faithful then assuming a unimodal Gaussian distribution. One of more contemporary formulations of the topic is [81]; a comprehensive overview of the principles with application to robotics can be found in [16].

Initial particle set $\mathbf{x}_0$ is again generated based on the best available a-priori knowledge. In the case of PF it is, however, wise to not underestimate uncertainty of newly generated particles – in case of inappropriately positioned particles, the measurement model becomes virtually ineffective in converging. If there is no a-priori information, the particles need to be generated covering the whole state space.

The filtering process is illustrated by Algorithm 1; in each filter execution iteration, particles from the preceding pass $\mathbf{x}_{i,k|k}$ are first processed by the motion model. This transforms them based on sampled control input $\mathbf{u}_k$ according to its uncertainty measure $\mathbf{V}$. The measurement model then determines the importance of each particle by formulating the weight $w_{k+1|k}$ based on the measurement vector $\mathbf{y}_{k+1}$ and its uncertainty $\mathbf{W}$.

Algorithm 1 Particle filter / Monte Carlo Localization

1: **for** i = 0...N **do**
2: $\quad \mathbf{x}_{k+1|k} \leftarrow \mathbf{motionModel}(\mathbf{x}_{i,k|k}, \mathbf{u}_k, \mathbf{V})$
3: $\quad w_{k+1|k} \leftarrow \mathbf{measurementModel}(\mathbf{x}_{i,k+1|k}, \mathbf{y}_{k+1}, \mathbf{W})$
4: **end for**
5: $w_{:,k+1|k} \leftarrow w_{:,k+1|k} \,/\, \mathbf{sum}(w_{k+1|k})$
6: $\mathbf{x}_{k+1|k+1} \leftarrow \mathbf{resampling}(\mathbf{x}_{k+1|k}, w_{k+1|k})$

After all particles are processed, normalization of the assigned weights is performed to ensure a probability distribution function (PDF) with a unit sum on line 5. The final and most important step is resampling – it allows to re-generate the particles so that they represent distribution according to the computed weights. There are many possible resampling methods – e.g. residual, multinomial, stratified or systematic resampling.

3.2 Design

The following text aims to gradually guide the reader along the path of the pedestrian localization mechanism design. The individual steps are closely based on information given in Section 3.1 and use references to the equations and algorithms introduced there.

The overall situation is outlined in Fig. 3.7; it illustrates the building blocks, each of them providing a different type and quality of contribution to the localization estimation. The blocks will be described in-depth in subsections below – here is a short summary of them in the order of data fusion build-up:

- *GNSS* block is a rough source of geo-referenced localization data. Under good conditions it can be used as a relevant contributor to global location; when the estimation quality is determined poor, it merely reduces the set of landmarks to be matched, thus vastly improving computational performance of the initial estimate.
- *PDR* block provides high-quality step-wise position difference estimates. Their utilization is similar to any other odometry-like data source, but brings specifics such as sampling irregularly spaced in time (see below).
- *AHRS* block is mechanically coupled to the camera frame and reports its attitude with high frequency. The yaw/heading component of attitude can be deduced from the landmark subsystem, but is important during the mapping phase.
- *Image* block with subsequent landmark detection processes image streams from four cameras in order to find natural landmarks and extract their descriptor and angular position relative to the platform.

The landmark block inherently needs a map to be capable of geo-referenced operation. Landmark-enhanced localization then has a special mapping regime aiming to fabricate the landmark map; section 3.2.2 below provides all details. The design, however, starts without landmarks – by integration of the PDR step-wise pose change estimates.

3.2.1 Step-wise PDR data fusion into GNSS measurements

The currently operational global navigation satellite systems (GNSS) provide the most widespread absolutely referenced positioning service available, even if their performance is limited to known extents as analyzed in section 3.1.3. A GNSS receiver can be found in

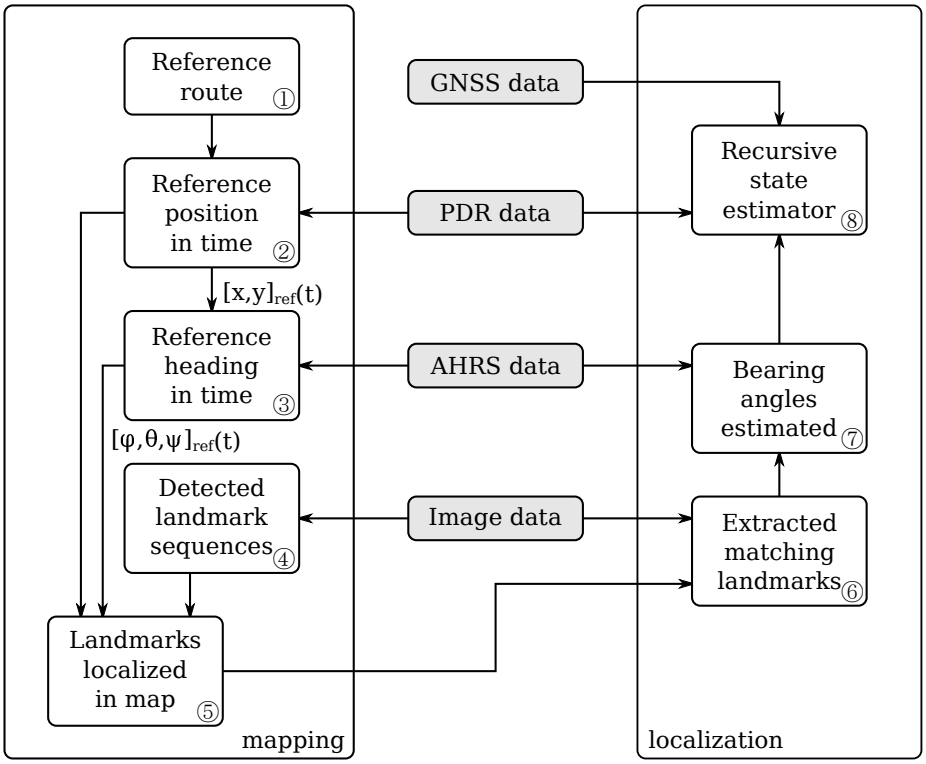


Figure 3.7: Overall structure of the natural landmark aided localization mechanism.

the majority of today's mobile communication devices. While there are techniques that allow significant improvements of GNSS position estimate dynamics by tightly coupling the pseudo-range navigation equations to IMU measurements [82, 83, 84], the necessary raw GNSS data output is only rarely available on wearable consumer-level devices.

The approach of the first step of GNSS position estimate improvement, necessary for successful pedestrian application, is focused on local behavior of the estimated pose. GNSS estimates without further processing are known to often deform the shape of the estimated trajectory; locally, they also tend to deviate significantly under a tree canopy or near buildings where satellite signal deterioration occurs. The fusion mechanism GNSS-PDR has been thus established, aiming to discipline the GNSS in short to mid term.

The global-level results will still largely depend on quality of the GNSS position estimates. This is, however, expectable and cannot be reliably rectified without additional use of a non-GNSS globally referenced positioning mechanism, to which the section 3.2.2 tries to find answers.

System state representation

In this first step of building the complete localization mechanism, system state $\mathbf{x}$ is comprised of only the pedestrian's pose $\mathbf{p}$:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.31a}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \end{bmatrix} \tag{3.31b}$$

Two-dimensional representation not considering altitude is sufficient in this application; the localization subsystem is only the first layer of knowledge, and additional information, necessary e.g. in case of traversal of multi-level pedestrian bridges/crossings, will be provided mainly by the high-level navigation subsystem's segment map. Also, faithful estimation of the third dimension – altitude – is nothing short of a technical challenge.

Unscented Kalman filter and Particle filter frameworks have both been used to assess and compare their performance on the GNSS-PDR fusion problem; refer to Chapter 6 for detailed results.

Unscented Kalman filter approach

Initial estimate of the system state $\hat{\mathbf{x}}_0$ is chosen as the first valid GNSS fix – and its covariance becomes the initial estimate of the system covariance $\mathbf{P}_0$.

The PDR step-wise estimates are fused inside the motion model (process function) f which transforms sigma points $\boldsymbol{\mathcal{X}}_{k|k}$ into the predicted sigma points $\overline{\boldsymbol{\mathcal{X}}}_{k+1|k}$. This technique is commonly used [16] to incorporate odometry data (intuitively belonging rather to the measurement model). The control input $\mathbf{u}_k$ is the PDR-reported step-wise pose difference – it consists of positions increments Δx_k, Δy_k and yaw increment $\Delta\theta_k$ relative to the start pose of the current step. In order to integrate these increments into the k-th state's coordinate frame, the control vector $\mathbf{u}_k$ needs to be first rotated into the k-th state heading direction $\boldsymbol{\mathcal{X}}^{\theta}_{k|k}$ using the transformation matrix $R(\theta)$:

$$\mathbf{u}_k = [\Delta x_k \ \Delta y_k \ \Delta\theta_k] \tag{3.32a}$$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.32b}$$

$$\overline{\boldsymbol{\mathcal{X}}}_{k+1|k} = f\left(\boldsymbol{\mathcal{X}}_{k|k}, \mathbf{u}_k\right) = \boldsymbol{\mathcal{X}}_{k|k} + \mathbf{R}(\boldsymbol{\mathcal{X}}^{\theta}_{k|k}) \ \mathbf{u}_k \tag{3.32c}$$

GNSS directly estimates position component of the system state (pose), which makes the measurement model h very simple – just selecting the relevant part of the sigma points:

$$\overline{\boldsymbol{\mathcal{Z}}}_{k+1|k} = h\left(\boldsymbol{\mathcal{Z}}_{k+1|k}\right) = \boldsymbol{\mathcal{Z}}^{x,y}_{k+1|k} \tag{3.33}$$

Particle filter approach

Initial particle set $\mathbf{x}_0$ is generated around the first valid GNSS fix. In each filter execution iteration, particles from the preceding pass are processed by the motion model which, similarly to the UKF approach above, integrates the PDR data. Algorithm 2 lists the model steps that are executed on each particle: first, the control vector is sampled with mean value $\mathbf{u}_k$ and covariance $\mathbf{V}$ given by the PDR data. The state is then extended with the control vector transformed into the k-th state's coordinate frame by the already familiar transformation matrix $R(\theta)$ defined in Eq. 3.32b.

Algorithm 2 PF / MCL motion model

1: **function** MOTIONMODEL($\mathbf{x}_{i,k|k}, \mathbf{u}_k, \mathbf{V}$)
2: $\quad\hat{\mathbf{u}}_k \leftarrow \mathbf{u}_k + \mathbf{sample}(\mathbf{V})$
3: $\quad\mathbf{x}_{i,k+1|k} \leftarrow \mathbf{x}_{i,k|k} + R(\mathbf{x}^{\theta}_{i,k|k})\ \hat{\mathbf{u}}_k$
4: $\quad$**return** $\mathbf{x}_{i,k+1|k}$
5: **end function**

The second system-dependent part of the PF is the measurement model (Algorithm 3) that produces a probability distribution function based on sensory information other than PDF, in our case GNSS.

Algorithm 3 PF / MCL measurement model

1: **function** MEASUREMENTMODEL($\mathbf{x}_{i,k+1|k}, \mathbf{y}_{k+1}, \mathbf{W}$)
2: $\quad w_{k+1|k} \leftarrow \mathbf{gaussian2D}(\mathbf{x}^{x,y}_{i,k+1|k} - \mathbf{y}_{k+1}, \mathbf{W})$
3: $\quad$**return** $w_{k+1|k}$
4: **end function**

Particle deprivation avoidance One of the pitfalls of the particle filter is deprivation – i.e. excessive reduction of particles spread within the state space. This leads to underestimation of the state uncertainty and effectively prevents measurement model from correcting the state distribution. Generally, deprivation is more likely to happen when the particle set is limited.

There are several options for particle deprivation prevention, ranging from simple to elaborate [16] – however, all of them in principle generate additional random particles to "inflate" the particle spread and allow the measurement corrections and resampling process to work. The easiest option is to simply exaggerate the process noise; this option may, however, unnecessarily distrust the state. A smarter way, often called *jittering* or

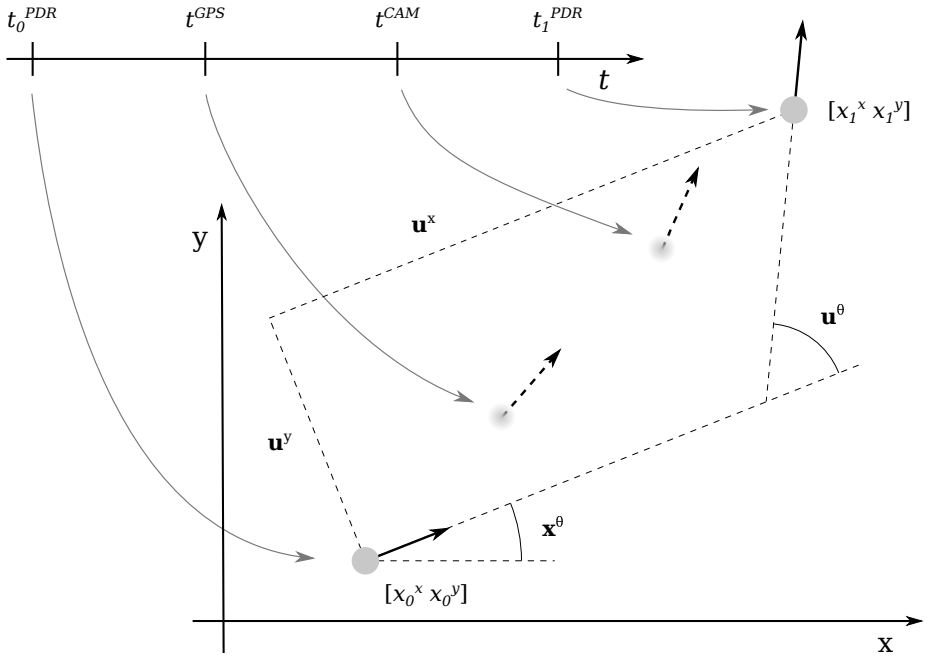


Figure 3.8: PDR data retrospective interpolation principle illustrated on two successive states $\mathbf{x}_0$ and $\mathbf{x}_1$, where the latter is given by applying PDR control input $\mathbf{u}$ in the reference frame of $\mathbf{x}_0$. Interpolated predictions for GPS and camera measurement integration are positioned between the two state linearly depending on time.

roughening, is to generate a fraction of the particle population randomly independently on resampling results. The amount of random particles can be then controlled by system's trust represented distribution health.

Systematic resampling has been selected – it performs well in terms of covering the particles weights uniformly, also helping prevent particle deprivation.

Multi-rate data sources integration

All sensor types intended for the localization subsystem are by their nature non-continuous, periodic – they publish measurement updates with either a fixed rate (GNSS, AHRS, cameras) or even irregularly (PDR). Both considered recursive state estimators, UKF and PF, however, need to apply the measurement update at the same time motion model has been predicted. With PDR used in the role of the prediction mechanism, the filter loses its flexibility of computing the motion model at any instant, whenever sensory data arrive and are to be fused into the state estimate.

In literature, there are various takes on the multi-rate / asynchronous sensors problem. In [85] and similarly also in [86], the authors propose usage of multiple filters from which the final estimate is then fused; [87] uses a two-stage filter for a dual sensor configuration. Finally, the solution described in [88] proposes a Bezier-curve based *filling* of gaps between precise, low-frequency measurements.

There are in principle two ways how this multi-rate synchronization problem can be solved in the realm of this thesis:

- *PDR extrapolation* at time of GPS arrival seems natural since PDR plays the role of prediction – this approach is often used when raw IMU data are to be fused [89].
- *PDR retrospective interpolation* to get $\Delta\mathbf{u}$ for any time instant between the last and the current PDR occurrence.

The second way has been chosen for the following reasons: first and foremost, PDR carries valuable information on the step-wise progress of pose; it would not be wise to compromise it by extrapolation. Second, a faithful dynamic motion model would be needed for the extrapolation, which is a complex task in case of the hardly-predictable human locomotion.

As illustrated by Fig. 3.8, the interpolation is driven by PDR data reception: state $\mathbf{x}_0$ is a result of PDR increment application at time t_0, state $\mathbf{x}_1$ will be predicted based on PDR data $\mathbf{u}$ that arrived at time t_1. In between, measurements are buffered together with their respective time stamps (i.e. t^{GPS} or t^{CAM}) and processed in a batch upon PDR data arrival. The total PDR pose increment $\mathbf{u}$, applied to state $\mathbf{x}_0$ in its reference frame, it then partitioned into multiple contributions $\Delta\mathbf{u}_i$ proportionally according to the time stamps of each measurement.

3.2.2 Natural landmarks for localization refinement

Building on the GNSS-PDR fusion mechanism described above, natural landmarks utilization adds an additional geo-referenced mechanism able to correct GNSS outages. Since there are no landmark maps readily available (as it is with the standard segment path maps), the landmark subsystem is divided into two phases, as illustrated in Fig. 3.7:

- *Mapping phase* happens typically once and is needed to be done for each part of the environment where the landmark augmentation should be available. It is done retrospectively off-line based on logged data and its output is a feature map allowing subsequent localization against it.
- *Localization phase* is than the real use of the landmark feature map to improve pose estimates – and is performed on-line within the recursive state estimation framework of choice.

Both the detector and descriptor based on SURF has been selected for the natural landmark aided localization application – providing the best compromise between feature matching and computational performance; patent limitations are not restricting the academic, non-commercial employment.

Mapping phase

The mapping phase is designed to be executed retrospectively on logged data to allow globally-optimal estimates to be established – opposed to real-time estimation working only with data known up to the specific instant. The structure si illustrated in the left part of Fig. 3.7; the process progressively goes through stages described below.

First, *reference route* trajectory is established as a position ground truth ①. The source of the reference data are coordinates of a path segment extracted from the segment map; local Cartesian coordinate system is assumed for all position-related operations. The representation is simply a vector of coordinates without heading or time information:

$$\mathbf{x}_{\text{route}} = \left[\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \cdots \begin{bmatrix} x_N \\ y_N \end{bmatrix} \right]^\top \tag{3.34}$$

This point-wise trajectory is then used to build a time-driven interpolation function producing *continuous position reference* based on log time ②. Fitting of reference positions to time axis is accomplished using the PDR data – the cumulative sum of distance increments is taken as progress measure within the reference trajectory, assuming the start and end positions are corresponding between the PDR and the reference route:

$$d^{\text{PDR}} = \sum_{i=0}^{N} \|\Delta \mathbf{x}_i^{\text{PDR}}\| \tag{3.35a}$$

$$d^{\text{ref}} = \sum_{i=1}^{M} \|\mathbf{x}_i^{\text{ref}} - \mathbf{x}_{i-1}^{\text{ref}}\| \tag{3.35b}$$

$$\mathbf{d}_{\text{cum}}^{\text{PDR}} = [d_0\ d_1\ \ldots\ d_N] \qquad \text{where} \quad d_k = \sum_{i=0}^{k} \|\Delta \mathbf{x}_i^{\text{PDR}}\| \cdot \frac{d^{\text{ref}}}{d^{\text{PDR}}} \tag{3.35c}$$

$$\mathbf{d}_{\text{cum}}^{\text{ref}} = [d_0\ d_1\ \ldots\ d_N] \qquad \text{where} \quad d_k = \sum_{i=0}^{k} \|\mathbf{x}_i^{\text{ref}} - \mathbf{x}_{i-1}^{\text{ref}}\| \tag{3.35d}$$

$$d_{cor}^{\text{PDR}}(t) = \text{interp}(\mathbf{t}^{\text{PDR}}, \mathbf{d}_{\text{cum}}^{\text{PDR}}, t) \tag{3.35e}$$

$$\mathbf{x}_{\text{ref}}(t) = \text{interp}(\mathbf{d}_{\text{cum}}^{\text{ref}}, \mathbf{x}^{\text{ref}}, d_{cor}^{\text{PDR}}(t)) \tag{3.35f}$$

First, the cumulative sum of PDR position increments $\mathbf{d}_{cor}$ is computed, assigning each PDR log time t^{PDR} a distance; correction based on total length of summed PDR and reference path is applied in Eq. 3.35c. A time-continuous distance function is built by a simple linear interpolation over points of $\mathbf{d}_{\text{cum}}^{\text{PDR}}$. This distance is then used as input in the second interpolation – "converting" reference distance to the final position reference $\mathbf{x}_{\text{ref}}$.

Similarly, reference of *roll*, *pitch* and *yaw* angles $\boldsymbol{\xi}_{\text{ref}}(t)$, where $\boldsymbol{\xi} = [\phi\ \theta\ \psi]$, is determined to provide time-continuous waveform ③. The same mechanism involving cumulated PDR position increments as in Eq. 3.35 is used; instead of using $\mathbf{d}_{\text{cum}}^{\text{ref}}$ and $\mathbf{x}^{\text{ref}}$ for the second

interpolation, AHRS-driven reference positions and attitude information are employed. Depending on the used AHRS sensor and its drivers, a conversion to Euler/Tait–Bryan angles according to Eq. 3.7 may be needed.

Landmark detection in camera image is then the next step to be performed ④. The aim is to find regions of interest in subsequent images that correspond to observations of the same natural landmark and group these into sequences for further processing. The process encapsulated within function `detectLandmarks` is captured in Algorithm 4; executing `runDetection` twice adds robustness by allowing up to one missing observation.

Algorithm 4 Natural landmark detection and grouping

```
1:  function detectLandmarks(images)
2:      for each camera image do
3:          landmarks ← runDetection(previous img, current img, landmarks)
4:          landmarks ← runDetection(2nd previous img, current img, landmarks)
5:      end for

6:      for each landmark do                ▷ prune landmarks with too few observations
7:          if landmark observations num < threshold then
8:              remove landmark
9:          end if
10:     end for

11:     return landmarks
12: end function

13: function runDetection(img1, img2, landmarks)
14:     f1 ← detectSURFFeatures(img1)
15:     f2 ← detectSURFFeatures(img2)
16:     matches ← matchFeatures(f1, f2, numBest)

17:     for each match do
18:         landmarks[match] ⁺← (feature, location, imageID)
19:     end for
20: end function
```

The final and most important step is *localization of detected landmarks* within a 2D map ⑤. The processing is done for each landmark observation sequence independently; operations for a single sequence are given by Eq. 3.37. Observations are formed by pixel position of the identified feature in particular image[11] – so that the first task is to convert the pixel position x to angular deviation from camera optical axis $\Delta\psi_\mathrm{l}$, using idealized pinhole camera model with focal length f expressed in pixels by Eq. 3.36a from camera horizontal resolution w and field of view ϕ_w:

[11] This section assumes the image distortions have already been rectified by intrinsic calibration chain described in section 5.1.3. Also, the camera images need to be corrected according to current roll or pitch estimate, depending on the camera placement.

$$f = \frac{\frac{w}{2}}{\tan(\frac{\phi_\mathrm{w}}{2})} \tag{3.36a}$$

$$\mathbf{c} = [\,0 \quad 0 \quad f\,] \tag{3.36b}$$

$$\mathbf{x} = \left[\, x - \frac{w}{2} \quad 0 \quad f \,\right] \tag{3.36c}$$

$$\Delta\psi_\mathrm{l} = \mathrm{sgn}\left(x - \frac{w}{2}\right) \arccos\left(\frac{\mathbf{c}\cdot\mathbf{x}}{\|c\|\|x\|}\right) \tag{3.36d}$$

$$\psi_\mathrm{l} = \boldsymbol{\xi}^{\psi}_{\mathrm{ref}}(t) + \Delta\psi_\mathrm{c} - \Delta\psi_\mathrm{l} \tag{3.36e}$$

where vector $\mathbf{c}$ connects focal point and center of image plane and vector $\mathbf{x}$ points from focal point to the landmark's projection to x axis. The angular deviation $\Delta\psi_\mathrm{l}$ expression is based on definition of dot product $\mathbf{a}\cdot\mathbf{b} = \|a\|\|b\|\cos\theta$, where θ is the angle between $\mathbf{a}$ and $\mathbf{b}$. Finally, the globally-referenced yaw angular position ψ_l of the landmark is determined by using the current yaw of the camera platform[12] $\boldsymbol{\xi}^{\phi}_{\mathrm{ref}}(t)$, relative heading of the particular camera[13] $\Delta\psi_\mathrm{c}$ and the landmark angular deviation from the camera image axis[14].

After yaw has been determined for each landmark, a measurement vector is assembled containing all observations and respective reference positions[15] from which the observation was made; refer to Fig. 3.9 for an example situation. In a real-world application, the observations will not be exact and the landmark position needs to be found by non-linear least-squares optimization – generally, this corresponds to the problem of finding the point $\hat{\mathbf{x}}$ of configuration space $\mathfrak{R}^n$, corresponding to the residual point $r(\hat{\mathbf{x}})$ in observation space $\mathfrak{R}^m$ closest to the origin [90]. In our case, position of landmark in 2D map is sought for, i.e. $n = 2$; the observation space dimension m is driven by the particular number of each landmark observations. The following equations lay foundations for the optimization process:

$$r_i(\mathbf{x}_\mathrm{l}) = \frac{\mathbf{x}^y_\mathrm{l} - y_i}{\mathbf{x}^x_\mathrm{l} - x_i} - \tan(\psi_{\mathrm{l},i}) \tag{3.37a}$$

$$r(\mathbf{x}_\mathrm{l}) = [r_1(\mathbf{x}_\mathrm{l})\; r_2(\mathbf{x}_\mathrm{l})\; \ldots\; r_m(\mathbf{x}_\mathrm{l})]^\top \tag{3.37b}$$

$$\hat{\mathbf{x}}_\mathrm{l} = \underset{\mathbf{x}_\mathrm{l}}{\mathrm{argmin}} \parallel r(\mathbf{x}_\mathrm{l}) \parallel^2 \tag{3.37c}$$

[12] Determined using corrected AHRS data, see the implementation chapter for details.

[13] Intuitively, front camera has zero relative heading, left camera is positioned at $+\frac{\pi}{2}$, right at $-\frac{\pi}{2}$ and rear at π radians. Consult section 5.1.3 for details on determining exact relative headings through the process of extrinsic calibration.

[14] The term $\Delta\psi_\mathrm{l}$ needs to be subtracted since the image coordinate system is defined counter to the oriented angle direction.

[15] Cameras are in fact positioned off the pedestrian's longitudinal axis, so that a time-invariant camera position compensation $\Delta\mathbf{x}_\mathrm{c}$ needs to be added to the reference position $\mathbf{x}_{\mathrm{ref}}(t)$ for each camera separately.

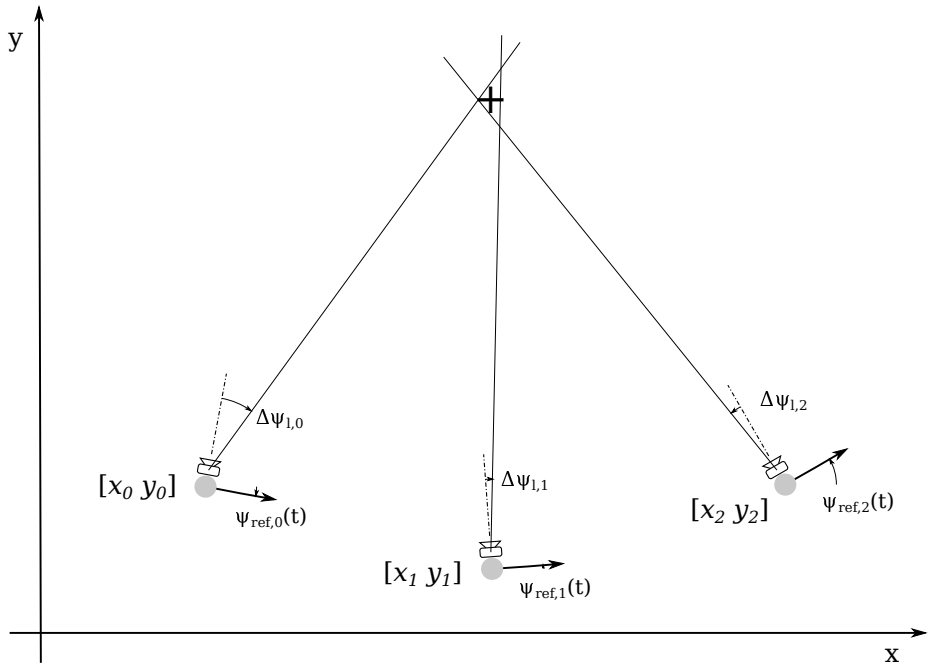


Figure 3.9: An example of three slightly inaccurate observations of a landmark denoted by the sign $+$, all using left camera so that $\Delta\psi_c = \frac{\pi}{2}$ for all points. Dash-and-dot lines are camera image axes, solid lines represent determined observation direction of the landmark.

where $\hat{\mathbf{x}}_l = [\hat{x}_l\ \hat{y}_l]$ is the estimated landmark position and residuals $r_i(\mathbf{x}_l)$ corresponding to each observation are assembled into the residual vector $r(\mathbf{x}_l)$. Important for successful convergence is the choice of the initial estimate; a simple yet robust solution has been chosen by intersecting the outermost observation directions – the resulting point is then used as the initial estimate. After the argmin search is done, the resulting value of the objective function $f(\hat{\mathbf{x}}_l) = \| r(\mathbf{x}_l) \|^2$ is evaluated as a measure of observation quality – in case of too high a value, the landmark is completely discarded.

After determining the landmark position for each of the observation sequences, a map is assembled from the "surviving" ones; each landmark is represented in the map by its *estimated position* $\hat{\mathbf{x}}_l$, the *estimation quality* $f(\hat{\mathbf{x}}_l)$ and a set of the *SURF features* for future matching[16].

Localization phase

Blocks of the "forward" use of natural landmarks identified in camera images constitute right part of Fig. 3.7 – it can be seen that sensory data sources are utilized to similar benefit in this phase, however, by using on-line recursive state estimation rather than retrospective evaluation.

[16]Each feature is accompanied by the position from which is was observed – this helps to filter visible landmarks in the localization phase, see below.

The first task is to extract natural landmarks from the current camera image and match them to the feature map ⑥. Algorithm 5 lists the operations needed; the line 3 is worth noting – a filtering of the map's features to be matched is performed, preferring near-observed ones. The result of function `extractLandmarks` is then a subset of the map with visible landmarks, and also a metric of how each match is believed to.

Algorithm 5 Natural landmark extraction and matching with map

```
1: function EXTRACTLANDMARKS(image, map)
2:     f1 ← detectSURFFeatures(image)
3:     f2 ← observable map features near current position
4:     matches ← matchFeatures(f1, f2, numBest)

5:     return map matches, match scale
6: end function
```

Once the landmarks have been matched to the map, relative observation angles towards them are determined similarly to the process described by Eq. 4 ⑦. The difference lies in the last step Eq. 3.36e – the term $\boldsymbol{\xi}^{\psi}_{\mathrm{ref}}(t)$ is not used since only the bearing within pedestrian-centric coordinate system is needed:

$$\Delta\psi^{\mathrm{o}}_{\mathrm{l}} = \Delta\psi_{\mathrm{c}} - \Delta\psi^{\mathrm{o}}_{\mathrm{l}} \tag{3.38}$$

Integration into the recursive state estimation framework is done intuitively through the measurement vector and appropriate measurement vector ⑧. The landmarks are considered independent, bearing-only and with known correspondence [16]; [74] describes a similar scenario with bearing-only infrared beacons fused into pose estimates by means of EKF framework. For further work with the landmarks, UKF approach has been selected – its simpler representation and deterministic behavior proved easier to design with.

The measurement vector is formed simply by stacking as many relative observation angles $\Delta\psi^{\mathrm{o}}_{\mathrm{l},i}$ as there are matched landmarks within the current camera image. Since the landmarks are considered independent, the respective covariance matrix is diagonal, with non-zero items based on inverse value of the map match scale:

$$\mathbf{y} = \begin{bmatrix} \Delta\psi^{\mathrm{o}}_{\mathrm{l},0} & \Delta\psi^{\mathrm{o}}_{\mathrm{l},1} & \dots & \Delta\psi^{\mathrm{o}}_{\mathrm{l},o} \end{bmatrix}^{\top} \tag{3.39a}$$

$$\mathbf{W} = \lambda \begin{bmatrix} \frac{1}{scale_0} & & \\ & \ddots & \\ & & \frac{1}{scale_o} \end{bmatrix} \tag{3.39b}$$

where o is the number of matched observations and λ is a scalar constant needed to scale the range of inverse match scale to the measurement uncertainty.

The last step is preparation of the measurement model – the non-linear function h that transforms the state into the output vector. In terms of UKF framework we need to transform the re-generated state sigma points $\boldsymbol{\mathcal{Z}}_{k+1|k}$ into the output sigma points $\overline{\boldsymbol{\mathcal{Z}}}_{k+1|k}$; compared to Eq. 3.24, a new parameter is the current map of matched landmarks corresponding to the measurement vector $\mathbf{y}$. The modeled relative observation angle to each visible landmark $\Delta\psi_{\mathrm{l},i}^{m}$ is computed from the current system state's position components and the landmark position gained from the map.

$$\Delta\psi_{\mathrm{l},i}^{m} = \operatorname{arctan2}(\hat{\mathbf{x}}_{\mathrm{l}}^{y} - \boldsymbol{\mathcal{Z}}_{k+1|k}^{y}, \hat{\mathbf{x}}_{\mathrm{l}}^{x} - \boldsymbol{\mathcal{Z}}_{k+1|k}^{x}) \tag{3.40a}$$

$$h(\boldsymbol{\mathcal{Z}}_{k+1|k}, \hat{\mathbf{X}}_{\mathrm{l}}) = \begin{bmatrix} \Delta\psi_{\mathrm{l},0}^{m} & \Delta\psi_{\mathrm{l},1}^{m} & \dots & \Delta\psi_{\mathrm{l},o}^{m} \end{bmatrix} \tag{3.40b}$$

where $\hat{\mathbf{X}}_{\mathrm{l}}$ is a matrix consisting of the landmark positions $\hat{\mathbf{x}}_{\mathrm{l}}$ for each of the o matched landmarks. Note that all operations working with sigma points $\boldsymbol{\mathcal{Z}}_{k+1|k}$ are in fact doing the operation on the $1 + 2n$ set of sigma points.

Chapter 4

Path Planning

The path planning and obstacle avoidance topic is rendered according to the very same principles as the localization subject was presented in the preceding chapter. Building on the overview given by an analytical introduction, particular solutions are proposed in the design section.

4.1 Analysis

Similarly to the application in mobile robotics, navigation subsystem is responsible for generation of motion instructions optimal for reaching the goal. Specialization to the "human chassis" means both drawbacks and advantages: although the human can only process a limited amount of data, is in return able to understand much more complex and meaningful instructions.

Information mediated to the user can be divided into two layers, alongside with the planner responsibility division introduced below:

- *High-level data.* Richer meaning issued at lower frequency – an output generated based on the global path plan and complex features of the map. Such is e.g. an instruction to cross a junction, go 100 m straight or follow tactile paving.
- *Low-level data.* Commands that need urgent reaction but do not carry complex information. The local path execution outputs fall into this category and are comprised of such primitives as straight motion, left or right turn.

The goal of this responsibility division is important: to convey enough instructions for successful and safe path plan execution, while leaving the VI users' senses as free as possible. This is greatly important in the case of the only long-range sense available to the sightless – the hearing – since it is their only mechanism of danger prediction.

4.1.1 Global path planning

The first task of the developed system requested by the user is an assessment of the route to be traversed. Based on the actual location and the chosen target to be reached, the path planning mechanisms are responsible for assembly of a path plan – an ordered sequence of path segments that connects the start and destination, chosen according to suitable optimality criteria. This kind of functionality is often referred to as discrete planning (the state space is not continuous).

The main input of global planning layer is intuitively a *map* – and, unlike the feature map introduces for landmark localization, this map is in essence very similar to what cartography deals with – a system of roads, sidewalks or generally *path segments* interconnected by junctions. Secondly, global planner may take advantage of other kind of information as well, which are likely to improve the path plan according to defined metrics, e.g. public transport timetables, list of road closures and so forth.

Map as a weighted graph

One of the objects of study in the field of discrete mathematics is the graph theory, which provides a useful formalism for topological map representation and work with. Computer science then builds on this foundation with the focus on graph-driven state space search, which is already directly applicable to path planning problems. Let us introduce necessary primitives first.

A graph $G = (N, E)$ is an ordered pair constituted by a set of nodes N and a set of edges E. Individual edges are then two-element subsets of N; in case the graph is defined as directed (edges have orientation, arrows) then the subsets are ordered, which is the usual case of map representations. In our case, a *weighted* graph is of particular interest: such graph introduces a numeric value $w\colon E \to \mathbb{R}^+$ to denote a certain quality of each edge; graph search methods then use the weights to find the optimal path.

Basic operations on a graph In terms of graph search, there is an initial node where the path starts and a target node which we want to reach; they are interconnected by a directed weighted graph. At the beginning, only the initial node has been visited; gradually, certain neighboring nodes (i.e. those directly connected through edges) are explored by *expanding* the nodes, i.e. extending the path to them.

To keep track on which node has already been rejected and which is still to be visited, two mechanisms are commonly used: first, *priority queue* is suitable for storing nodes that should be expanded next; second, *sets* without order information are used to store graph nodes that have already been evaluated to some extent.

Graph search algorithms

Graph or state space search has likely been one of the most popular algorithmic topics among CS/IT-related syllabi. Because of this omnipresence, the description hereinafter will be restricted to a few paragraphs only without reprinting complete algorithms; dear reader is kindly asked to find details in the cited sources.

Generally, the methods can be divided to *uninformed* and *informed* – the former group traversing the graph without any "navigation" towards the target node; such algorithms are of little interest to this thesis. The latter group of informed methods then generally work by evaluating *cost* function $f(n)$ on the last node n of each generated path which, depending on the particular algorithm, helps to find path close to optimal as efficiently as possible (expanding minimum number of nodes).

A* Introduced by [91], A* algorithms is the most flexible from the "traditional" deterministic informed methods – its cost function $f(n)$ that is minimized during path search is composed of the up-to-now price $g(n)$ and estimated cost $h(n)$ from node n to target. This arrangement allows to vary the proportion between path optimality and search efficiency, just by changing the proportion between the two components:

$$f(n) = g(n) + h(n) \tag{4.1}$$

The first part is deterministic – since we have already expanded the path from start to node n, we know the cost exactly. On the other hand, the second part is heuristic and needs to be estimated based on each particular application; in map planning, Euclidean or Manhattan metric from node n to target are typically used. An important aspect here is *admissibility* of the heuristic $h(n)$, meaning that it does not overestimate the cost – if satisfied, A* will always return optimal path.

Dijkstra, Greedy Best First Search These two algorithms can be perceived as a simplification of A*; the Dijkstra's algorithm [92] only uses the cost function $g(n)$ and thus always finds the optimal path, although with a big overhead of many expanded nodes. Greedy BFS by contrast works only with the heuristic $h(n)$ which pushes it aggressively towards the target, but it may lead to a sub-optimal path.

High-level navigation instructions

Once a path plan is prepared, the pedestrian needs to be properly instructed to be able to follow it. With respect to the common sat-nav devices, timing and contents of the instructions determine usability and safety substantially more — the VI person does not have means to correct potential shortcomings of the instructions.

Human-oriented design is an apt candidate for soft computing methods utilization, including fuzzy logic [93], neural networks [94] or probabilistic logic [95], mostly sharing their roots in the Dempster–Shafer theory [96, 97]. Fuzzy logic [98, 99] has been known to be able to achieve a "human-like" experience of the navigation subsystem instructions. In [100] and [101], examples of fuzzy-powered applications are given with the aim of being human friendly.

Types of announced information In the relevant literature, an abundance of proposed instructions for a visually impaired user can be found. Representatively, [102] and [103] introduce 13 different types of guidance functions based on various arrangements of the path segments and junctions and situations the VI pedestrian may encounter. Inventing a custom instruction methodology would be an interesting but exhausting research topic on its own – for the implementation, a subset of the referenced studies is to be used.

4.1.2 Local path planning

While the high-level global path planner works discretely with a segment-granularity, another layer is needed to perform continuous navigation within each segment. The global planner also in principle does not need a precise pose information from the underlaying localization subsystem: immediate motion corrections are commanded by the local planner to keep the user on route.

The field of autonomous mobile robotics offers many inspirational approaches on this subject – local planning is a necessary part of robot control routines in nearly all cases (which, for example, does not hold true for segment-like planning). An overview of such methods can be obtained by studying any of the cited robotic "Bibles" [16, 17, 18]; let us consider only those conforming to requirements given by the pedestrian application.

To establish a reference of local planning functionality, let us consider the *potential fields* method – in essence gradient maps or vector fields that define a direction of motion for each point of a certain neighborhood around the current position [104]. The gradient is chosen in a form of potential function $U : \mathbb{R}^m \to \mathbb{R}$ by adding *attractive* (target) and *repulsive* (obstacle) potentials [18]. The direction is then generated by means of the *gradient descent* method, i.e. following $-\nabla U$. A significant benefit of this method is its flexibility – an arbitrary number of potentials can be used to express obstacles or preferred segment shape.

The human who interprets instructions of the local planner is obviously intelligent in the sense that the instructions do not need to be as exhaustive compared to the application in robotics[1]. The human user is also able to solve unexpected situations. The output of the local planner is thus expected to mainly *indicate the direction* to follow the segment shape.

[1] In robotics, the chassis typically does not possess any decisive authority, so that all control must come from the local planner – motion speed and turn radius as the very minimum.

A simpler arrangement can be then considered, expressing only the segment course, which is sufficient especially when obstacle avoidance is solved by the pedestrian themself (see below). In such case, a simple yet effective direction instruction engine can be obtained by pointing to a point lying on the segment slightly ahead (typically a few meters [14]).

Low-level navigation stimuli

Similarly to the global planner, also the local one needs a proper user interface channel able of conveying directions to the user. As pointed out at the beginning of the chapter, these low-level data are simple but frequent – while segment length can be announced once per tens or hundreds meters, the pedestrian's heading may need step-wise corrections.

This topic has been discussed in relevant literature. The authors of [105] compared haptic and auditory bearing-only indicators, working with a tolerance of $\pm 10\,^\circ$ for on-course or off-course signals; haptic and audio signals are rated on par with the remark that the haptic mechanism can be used even by deaf-blind users. The study [106] results in preference of the haptic interface and strongly argument against overuse of the audio channel.

Investigating the haptic interface further, single- or double-transducer arrangement is commonly proposed and evaluated; the cited papers differ in complexity of information expressed. Starting with [105] which only indicates "within $\pm 10\,^\circ$" or "outside $\pm 10\,^\circ$" binary information and ending with [107] conveying side and magnitude of heading deviation, both solutions use single vibratory channel. A simplified scheme is proposed by [108], inspired by a notional radar screen and also able to express both the side and magnitude components.

4.1.3 Obstacle avoidance

The task of obstacle avoidance is composed of two stages: first, the obstacle needs to be detected, which is partially responsibility of proper sensory equipment and suitable data processing. Second, detected obstacles need to be registered by the path planning subsystem; usually, only the local planner deals directly with obstacles – when avoidance is not possible, the responsibility is shifted to the high-level navigation layer which marks the segment as non-traversable and finds an alternative route.

While the second task can be, with some intentional simplification, considered just a matter of proper algorithmization, sensing an obstacle is not always straightforward. Let us begin by defining what actually is an obstacle for a VI pedestrian: intuitively, any object in the trajectory that the pedestrian can collide with can be considered an obstacle. However, the collision may not only occur as a frontal crash – equally problematic are smaller environment features like curbs, sewers, drains and even unexpected pavement unevenness; all these elements are likely to make the pedestrian stumble. What is worse, however, is that such features are often very hard to sense and distinguish from sensor noise.

It is not the objective of this thesis to enumerate all possible sensory principles that could detect some kind of an obstacle; obstacle detectors likely constitute vast majority of sensors used in robotics. Nevertheless, no robust wearable sensory technology was found which would reliably detect even small terrain irregularities, that may yet lead to a stumble and potential injury. For this reason, it was decided to suspend design and development efforts on this topic and assume further employment of the white cane for obstacle detection. Otherwise, an inherently unsafe subsystem for a user would be designed which is not worth the efforts.

4.2 Design

Building on the path planning analysis, this section explains particular design principles applied to solving the path planning problems; it is again divided along the global/local borderline.

4.2.1 Global path planning

The responsibility covered on the global level is twofold: first, an optimal segment path plan needs to be found, and second, the user needs to be instructed properly to execute this plan.

Criteria for pedestrian path planning

The analytical section has introduced three applicable algorithms commonly utilized in map search applications. The first design decision is the following choice: the A* algorithm has been selected, since it forms a superset of both Dijkstra and greedy BFS, and can easily descend to both these forms just by zeroing proper part of the cost function. Because the A* algorithm is well known, the description here will be limited to what is application-specific: the cost function.

Cost representation An important realization is that the path planning in scope of this thesis is focused on, to certain extent, soft and subjective matter – easiest path for the VI pedestrian – rather then well quantifiable problems of shortest or fastest route. Weight of an edge in map/graph has to, however, correspond to a numerical measure; quantification is thus inevitable.

The essential decision is to base the costs on distance rather than time of traversal (which is the common default path optimality metric of consumer "sat-nav" systems). The rationale is that the primary aim is to make the route as safe and easy as possible, which is intuitively related to total distance the pedestrian has to cover; time demands can be perceived as secondary [109].

Cost from start The exact part $g(n)$ of the cost function is evaluated deterministically and influences optimality of the resulting path plan.

$$g(n) = g(n-1) + w(E(n-1,n)) \tag{4.2}$$

$$w(e) = \left(\prod_{\Gamma} \gamma_k\right) w_0(e) + \sum_{\Delta} \delta_k \tag{4.3}$$

$$w_0(e) = \begin{cases} \lambda_{\mathrm{pt}}|e| & \text{when } e \text{ is public transport} \\ \lambda_{\mathrm{s}}|e| & \text{when } e \text{ is sidewalk} \\ \lambda_{\mathrm{st}}|e| & \text{when } e \text{ is staircase} \\ \lambda_{\mathrm{i}}k & \text{when } e \text{ is intersection with } k \text{ lanes} \end{cases} \tag{4.4}$$

$$\lambda_{\mathrm{pt}} < \lambda_{\mathrm{s}} < \lambda_{\mathrm{st}} << \lambda_{\mathrm{i}} \tag{4.5}$$

where $E(n-1,n)$ selects edge e leading from node $n-1$ to node n, $|e|$ stands for length of edge e and λ coefficients allow tuning of the rules in terms of their impact on the planned path. Additionally, there is the set Γ of multiplicative and set Δ of additive coefficients to express supplementary knowledge types that influence resulting cost of the edge, in case they are found tagged on the edge:

- dangerous spots are penalized additively for each such spot by contributions δ_{d}
- tactile pavement is rewarded multiplicatively by coefficient γ_{t}, in proportion to its relative length against current edge length
- acoustic signal at intersection is rewarded multiplicatively by coefficient γ_{a}

Multiplicative reward here means *decreasing* the cost, satisfied when $\gamma_i < 1$.

Heuristic The estimated cost of transition from the current node to the target cannot be determined based on such deterministic rules as introduced above; at the time of node n expansion, the details on the route continuation are naturally not yet known. The main criterion here is *admissibility* of the chosen heuristic, to not jeopardize solution optimality. Euclidean distance from the current node n to the target node t:

$$h(n) = \lambda_{\mathrm{h}} \left\| t - n \right\| \tag{4.6}$$

The tuning coefficient λ_{h} has been introduced to facilitate adjustment of mutual importance between $g(n)$ and $h(n)$ – or, how important the heuristically dictated direction towards target node is, compared to the exact cost of the current node. In the trivial case of $\lambda_{\mathrm{h}} = 0$, the A* algorithm degrades to Dijkstra.

High-level instructions generation

Based on the reasoning stated in the analysis above, fuzzy logic mechanisms have been selected to support the high-level message generation mechanism intended to inform and instruct the user about complex eventualities, mostly related to the segment map: shape of the actual segment, where to go on a junction, whether and which potential structural aids are currently usable (see Chapter 1 for overview) and so forth.

Fuzzy logic mechanisms have been applied to these aspects of the generation process:

- *Suitable instructions selection.* Instructions are selected from the database based on evaluation of the activation conditions in the form of fuzzy if-then rules [110].
- *Proper timing of the selected instructions.* Driven by a set of fuzzy conditions, particular instruction execution can be planned with respect to distance or time point within the current path segment.
- *Determination of instruction priority.* Again, based on if-then rules, the instruction can be prioritized according to any of the defined linguistic variables.
- *Evaluation of linguistic variables inside the instruction.* Variable placeholders inside the predefined instructions can be replaced by a linguistic or defuzzified value.

A similar approach of fuzzy activation conditions has been utilized in [111] for high-level behavior control of an autonomous mobile robot.

Linguistic variables Central to the idea of fuzzy instructions generation is utilization of linguistic variables – i.e. variables whose values are formed by words in natural language [112, 113] – to enable human-readable instruction rule database establishment. The basic linguistic variables defined for the instruction generator purposes are listed in Table 4.1.

Among the most important LVs is the current segment length (`segmentLength`) and user's position within the segment (distance walked from its start point, expressed by variables `distanceOnSegment` and `distanceOnSegmentRel`). The membership functions of these two selected linguistic variables are shown in Fig. 4.1. It can be seen that the segment length is introduced in a "logarithmic" manner that corresponds to usual binning of urban segment lengths (function "large" continues to infinity). On the other hand, the relative version of the user's distance on segment (`distanceOnSegmentRel`) partitioned uniformly allows one to plan execution of an instruction based on user's advance on the segment independently on segment length.

If-then rules and inference Instructions hold three different rule sets: activation conditions, timing and priority. The syntax of each condition is fixed to the following format:

```
if LV1 is <hedge> MF1 [and/or LV2 is <hedge> MF2 ...]  then LVN is <hedge> MFN
```

where `LVx` and `MFx` is linguistic variable with its respective membership function name, `<hedge>` is a placeholder for potential hedge modifier and the part in square brackets contains the second to N-th sub-expression. The below excerpt from a segment length announcement demonstrates the inherent clarity of fuzzy-based linguistic formulation:

```
if segmentLength is zero or segmentLength is small then activation is denied
if segmentLength is not zero and segmentLength is not small then activation is allowed
```

Mamdani's inference system is then used to evaluate the rules and after the inference mechanism execution, output variables can be defuzzified and processed. In case of the `activation` result quoted above, the defuzzyfied value is thresholded to prepare it for Boolean logic that follows.

Timing conditions of the instructions whose activation conditions have been met are periodically evaluated and when also met, the instruction is prepared for execution and sent to the audio module. The preparation includes inference of the priority value and evaluation of variable placeholders.

Example To illustrate the functionality, let us define the following instruction rules (obviously, in implementation, the message contents need to be significantly richer) – the conditions have been stripped from the `if-then` clause for easier orientation:

ID	Content	Activation condition	Instant
1	Continue straight.	`headingDiff` is `zero` `segmentType` is `sidewalk`	`distanceOnSegment` is `zero`
2	Turn to `{headingDiff}`.	`headingDiff` is not `zero` `segmentType` is `sidewalk`	`distanceOnSegment` is `zero`
3	Use the crosswalk.	`segmentType` is `crosswalk`	`distanceOnSegment` is `zero`
4	Go `{val(segmentLength)}` meters.	`segmentType` is not `crosswalk`	`distanceOnSegment` is `small`

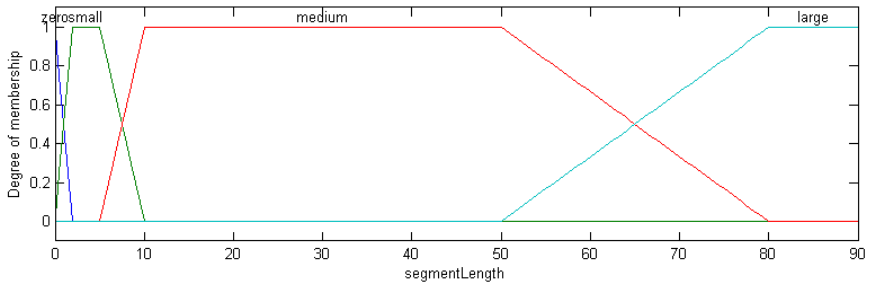
Additionally, consider the following path segments:

- segment, `segmentType` = `sidewalk`, `segmentLength` = 100 m (fuzzified as `large`), `heading` = 45°
- junction
- segment, `segmentType` = `sidewalk`, `segmentLength` = 60 m (fuzzified as `medium` = 0.67, `large` = 0.33), `heading` = 130°

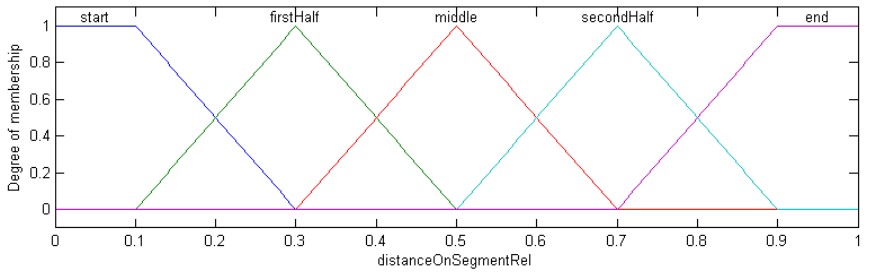
As soon as the user is at the beginning of the second segment, the system evaluates the instructions: ID 2 and 4 will be activated. ID 2 synthesis will be scheduled to the very start of the segment, and the placeholder `{direction}` will be replaced with the word "left" corresponding to the linguistic expression `right` (which will be chosen for the heading difference of 90°). ID 4 will be scheduled shortly after segment start, and the placeholder `{val(segmentLength)}` will receive an approximate segment length (it is not wise to flood the user with overly exact numbers).

Table 4.1: Overview of basic linguistic variables defined for the instructions generator.

Variable name	Meaning	Values
segmentType	Classification of the current segment	`sidewalk`, `crosswalk`, `road`, `stairs`, ...
segmentLength	Total length classification binning	`zero`, `small`, `medium`, `large`
distanceOnSegment	Current absolute distance from segment start	`zero`, `small`, `medium`, `large`
distanceOnSegmentRel	Relative and uniform position on segment	`start`, `firstHalf`, `middle`, `secondHalf`, `end`
headingDiff	Difference of current and last segment orientation, indicating direction of turn	`negativeLarge`, `negativeMedium`, `negativeSmall`, `zero`, `positiveSmall`, `positiveMedium`, `positiveLarge`



(a)



(b)

Figure 4.1: Visualization example of used linguistic variables: (a) `segmentLength` and (b) `distanceOnSegmentRel`. Trapezoidal membership functions used.

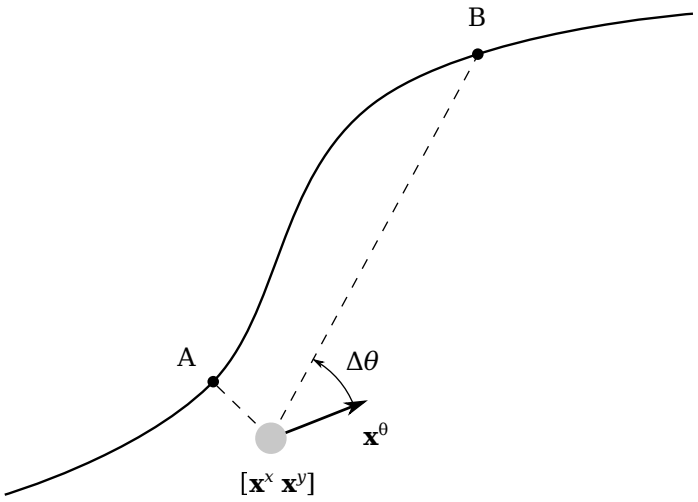


Figure 4.2: On the principle of local path planning based on heading correction.

4.2.2 Local path planning

Not considering the obstacle avoidance to be employed in the system, more complex local path planning methods are not justified. The simpler approach determining heading correction based on current segment's waypoint ahead has been selected. As Fig. 4.2 depicts, the functionality is comprised of the following steps:

1. find point A by perpendicular projection of position component of state $\mathbf{x}$ to the segment,
2. move along the segment N meters forward and find the point B,
3. compute heading update to aim to point B: $\Delta\theta = \arctan2(y^{\mathrm{B}} - \mathbf{x}^{y}, x^{\mathrm{B}} - \mathbf{x}^{x}) - \mathbf{x}^{\theta}$,
4. send $\Delta\theta$ to user interface for execution.

Low-level haptic instructions

Based on the analysis of applicable user interface mechanisms for conveying the real-time direction commands, haptic interface consisting of a pair of vibrating wristbands is proposed. The following instructions can be then handed over to the user:

- continuous vibration of one element – direction change command with vibration intensity dependent on heading error magnitude,
- continuous vibration of both elements – stop instruction,
- pulsing vibration – obstacle detected on the side or in front of the user (both elements pulsing).

Chapter 5

Implementation

For the research and development purposes, a two-phase experiment scheme was chosen: firstly, only raw sensory data are gathered without being processed on-line; the processing is then done off-line, outside the wearable platform. This approach is quite standard in such conditions since it brings a range of advantages: the processing pipeline is not restricted by real-time constraints (evaluated algorithms need not be strictly optimized), more importantly still, a multitude of approaches can be exercised and compared on exactly same inputs.

In this chapter, the architecture and important details of the data gathering and processing mechanisms are described. Starting with the wearable platform execution, characteristics of the employed sensory equipment and other specific instrumentation, the description continues with the data processing environment introduction. Finally, the relevant implementation details are given on the two main topics of localization and navigation.

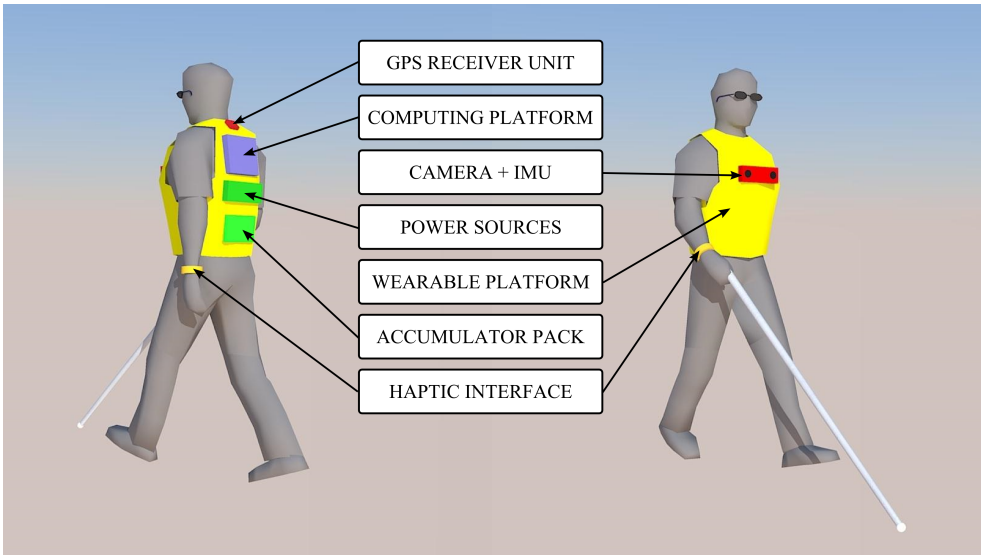


Figure 5.1: Sketch of the wearable platform main components; foot-mounted PDR IMU not visualized.

5.1 Wearable platform

5.1.1 First generation

In other to gather first data for the research on processing methods, the first-generation wearable platform (G1) was established. Based on a Load Bearing Equipment (LBE) system, it featured variable fixation points making the initial experiments easily reconfigurable. Figure 5.2 provides front and rear view on the arrangement (without cameras mounted).

Sensory subsystem was conceived quite generously in terms of custom development: starting with the critical component, inertial measurements unit (IMU). Depicted in Fig. 5.3, the device was boasting a 9DOF[1] measurement setup needed for implementation of a PDR or AHRS sensor. The GNSS localization source was represented by a module with the LS20031 receiver, capable of a 5 Hz SBAS/EGNOS-enhanced operation. Finally, four Microsoft LifeCam HD-3000 cameras with fixed-focus optics[2] and 1280x720 px maximal resolution at 30 fps. Among other devices of the "webcam" consumer range, it features quite a wide field of view (68.5° diagonally) and advantageous 16:9 ratio.

Computational platform was selected with a broad range of industrial communication ports in mind; PM-PV-D5251 in the PC/104+ format includes the dual-core Intel Atom D525 @1.8 GHz, 1 GB of DDR3 memory and a robust CompactFlash non-volatile storage module. The most powerful expansion interface is the on-board industrial PC/104+ socket – at early design time, the matching nDepth™ real-time stereo vision system [114] was foreseen as an auxiliary sensor for obstacle detection purposes.

Software platform was designed in a strictly modular way, using the great LCM[3] library [115] as the inter-module communication backbone running on the standard Debian system. The main paradigm (also shared with the current platform, see below) has been abstraction of particular data source technological details from the logical content – so that each component of the system can be replaced by a different type that, however, implements the same interface. Since all data were transferred over LCM channels, the library's generic tooling was heavily utilized for data logging and log playing. This inherent data processing mechanism enables great flexibility in system development and operation – modules above the driver layer basically do not know whether the processed data are live or played from a log file.

[1] Degrees of Freedom

[2] Fixed focus is preferable to automatic focus in outdoor employment: the usual sensing distances are well within the always-sharp range of approx. 0.5 m to infinity common for devices with a small sensor and high depth of field.

[3] LCM: Lightweight Communications and Marshalling was originally developed by the Talos team as the core of their DARPA Urban Challenge autonomous vehicle.

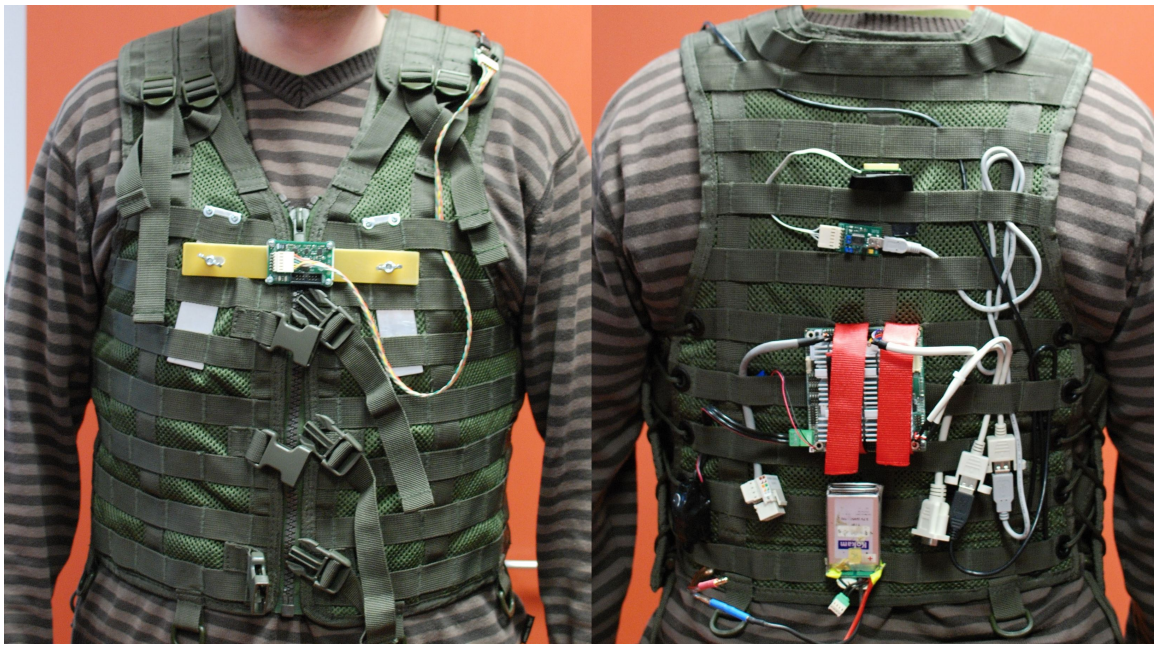


Figure 5.2: First-generation wearable platform, front and rear view, with the camera support frame detached.

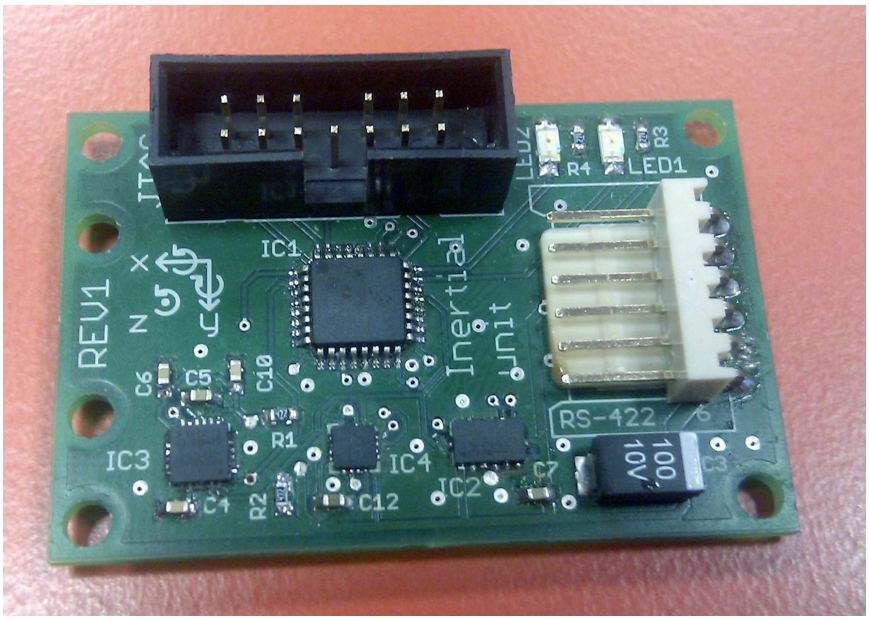


Figure 5.3: Custom Inertial Measurement Unit developed to support the first-generation wearable platform. Features the ADXL345 accelerometer, the ITG-3200 gyroscope and the HMC5883L magnetometer, together with a Freescale DSC for efficient on-board data processing; able of operation at sampling rate of 200 Hz.

5.1.2 Second generation

The second and current generation (G2) of the wearable platform was prepared to improve both mechanical and computational parameters of the original design.

Mechanically, the biggest difficulty of the G1 rig was its soft construction, with the camera platform holding to the LBE vest in a flexible manner; this complicated "wilder" manipulation with the system, such as is needed in order to perform magnetometer calibration (performing rotations about all axes) or extrinsic camera calibration. The second generation is then formed by a stiff modular plate that all equipment is fixed to – and which then connected to the LBE in an easily disassemblable manner as can be seen in Fig.5.4.

A progress has been made also in terms of the *host computer*; the original industrial single-board computer (SBC) was replaced by a more potent and recent NVIDIA Jetson TK1 platform featuring robust on-board eMMC non-volatile storage as well as modern, high-throughput buses as Gigabit Ethernet, USB3.0 or a miniPCIe slot. For a future real-time employment, the Jetson platform disposes of the NVIDIA Kepler GPU with 192 CUDA cores, enabling heavily parallelized processing.

Figure 5.4: Second-generation wearable platform, rear view.

Sensory equipment

Changes in the sensory subsystem of the G2 rig reflect not only the experience with G1 performance and (un)ease of use, but also professional maturing of the author[4]. A shift towards off-the-shelf sensory modules and plug-and-play connectivity of the USB bus can be observed in the architecture of G2 platform.

Osmium MIMU22BL Foot-mounted IMU sensor, a part of the OpenShoe project [116, 117], has been employed as the pedestrian dead reckoning data source. The device is remarkable not only for its on-board PDR processing pipe line, but also for the Multi-IMU (MIMU) nature. This approach builds on using multiple MEMS sensing components

[4] After graduating from the master of engineering programme, the author had been living under the euphoric impression that every problem can and should be engineered from scratch; the reality, however, only rarely allows this approach to succeed in finite time.

spatially arranged in way to compensate their offsets (mounting on opposite sides of PCB) and their signals averaged to improve noise characteristics [118].

Navilock NL-602U A consumer-grade GPS receiver has been chosen for several reasons – first and foremost, to get representative GNSS data that are likely to be qualitatively comparable to the data accessible on consumer-level smart devices that the system might be miniaturized into; i.e. no raw pseudo-range output, single-frequency only etc. Second, the price point and convenient USB connectivity made the choice only easier.

Microsoft LifeCam HD-3000 The same four pieces of USB "webcam" are used as in the first generation. Mounted on a stiff frame which guarantees constant relative transformation of their coordinate systems to the rig, they cover all four major directions: front, left, back and right. The camera mount has certainly its limits; while the position is defined precisely and permanently, the attitude of each camera cannot be defined neither precisely nor permanently without designing complicated fixtures. The problem is thus solved by accepting attitude deviations and compensating for it – see details in section 5.1.3 below.

There is a drawback in employment of multiple cameras: USB2 bandwidth limit[5]. Although there is a dedicated root hub for each USB port on the Jetson TK1 SBC, there are only 3 of them which is not enough for servicing 4 cameras simultaneously at full resolution and frame rate. Fortunately, the landmark detection functionality is not critically dependent on high frame rate; image resolution is then often lowered even intentionally to limit the number of SURF features detected and force the algorithm to work with larger features (in terms of solid angles they occupy in the image).

A working configuration has been found, concentrating all cameras on a single USB port through a hub – with image resolution of 416×240 px at framerate of 2 fps.

Xsens MTi-3 Fixed to the camera frame, the Xsens IMU [36] performs in the AHRS role, important mainly during the landmark mapping phase to provide global attitude reference (with reasonable dynamic response for compensation of the swinging motion, that the pedestrian's upper torso performs even when walking straight).

Robot Operating System (ROS)

The second-generation platform has also seen a software upgrade – although building on very similar, if not same, functional paradigms, the Robot Operating System framework[6] means a serious advance in usability and extensibility of the platform. Having gained a huge momentum through the recent years, the ROS environment is nowadays a de-facto

[5] One would assume that the adoption of USB3 improves the situation drastically; there are only two problems: (1) there are almost no native USB3 consumer-level cameras, (2) connecting multiple USB2 devices to a USB3 hub does not aggregate the USB2 traffic onto the USB3 channel...

[6] `http://www.ros.org/`

Table 5.1: Overview of the ROS nodes running on the wearable platform.

Package	Instances	Rate	Description
`usb_cam`	`cam_front`	2 Hz	Frame grabbing driver based upon Video4Linux that publishes still images from the four cameras.
	`cam_left`	2 Hz	
	`cam_back`	2 Hz	
	`cam_right`	2 Hz	
`xsens_driver`	`mtnode`	10 Hz	AHRS
`osmium_driver`	`osmium`	1–2 Hz	PDR
`nmea_navsat_driver`	`navsat`	1 Hz	GPS
`image_proc`	`image_proc_front`	2 Hz	Image rectification based on intrinsic calibration parameters.
	`image_proc_left`	2 Hz	
	`image_proc_back`	2 Hz	
	`image_proc_right`	2 Hz	
`apriltags_ros`	`apriltags_front`	2 Hz	Artificial landmark detectors for each of the rectified camera stream.
	`apriltags_left`	2 Hz	
	`apriltags_back`	2 Hz	
	`apriltags_right`	2 Hz	

standard of both stationary and mobile robotics instrumentation. This means that there are existing packages able of handling commonly used sensors; if none is existing for the particular sensor of choice, the environment is openly extensible.

The packages employed to perform the necessary sensory data capturing are listed in Table 5.1 together with their basic characteristics. The camera image chain starts with the generic `usb_cam` driver nodes and is followed by image undistortion and debayering step performed by `image_proc` nodes based on the intrinsic calibration (see section 5.1.3 below); the image is then ready for processing, either by a SURF natural landmark detector or by the AprilTag artificial landmark detector.

The only custom package that had to be programmed is the `osmium_driver`, due to relative obscurity of the Osmium device among ROS users[7]. The package follows the customary division to library and node part and was coded in Python which is the programming language of choice for such performance-modest tasks. It makes use of the standard `nav_msgs/Odometry` message type to convey step-wise difference of position and orientation and the corresponding covariance matrix. A unit quaternion is used to convey the difference attitude to prevent gimbal lock problems.

5.1.3 Calibration of cameras

There are two aspects to camera calibration – intrinsic and extrinsic – divided by the object of interest: the former estimates parameters of the optical system itself (sensor + lens), while the latter specifies the coordinate system transformation from `base_link` (i.e.

[7] As already mentioned at the beginning of this thesis, the pedestrian application does not attract that much attention compared to "traditional" wheeled/tracked/legged robotics.

pedestrian) to the camera. Overall, based on the calibration, a 3D world point can be projected using homogeneous coordinates to 2D image plane[8]:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \, [\mathbf{R}|\mathbf{t}] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{5.1}$$

Intrinsic calibration

Intrinsic calibration accounts for determination of the camera matrix $\mathbf{K}$, containing focal length in pixel units α_x, α_y, skew γ and principal image point $[u_0 \ v_0]$, and distortion coefficients needed to correct any non-linear lens imperfections.

$$\mathbf{K} = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.2}$$

Native ROS mechanisms were used to accomplish this task – the `camera_calibration` package was executed for each camera separately[9], using OpenCV calibration routines under the hood[10], by positioning a checkerboard of known size to various positions within the cameras' field of view. After gaining enough information and running the bundle adjustment algorithms, the resulting calibration was produced in the form of `camera_info` YAML files saved for each camera and subsequently automatically loaded by the `usb_cam` a published in each camera's `camera_info` topic. For example, the front camera matrix was determined as, while nominal parameters of the resolution are 416×240:

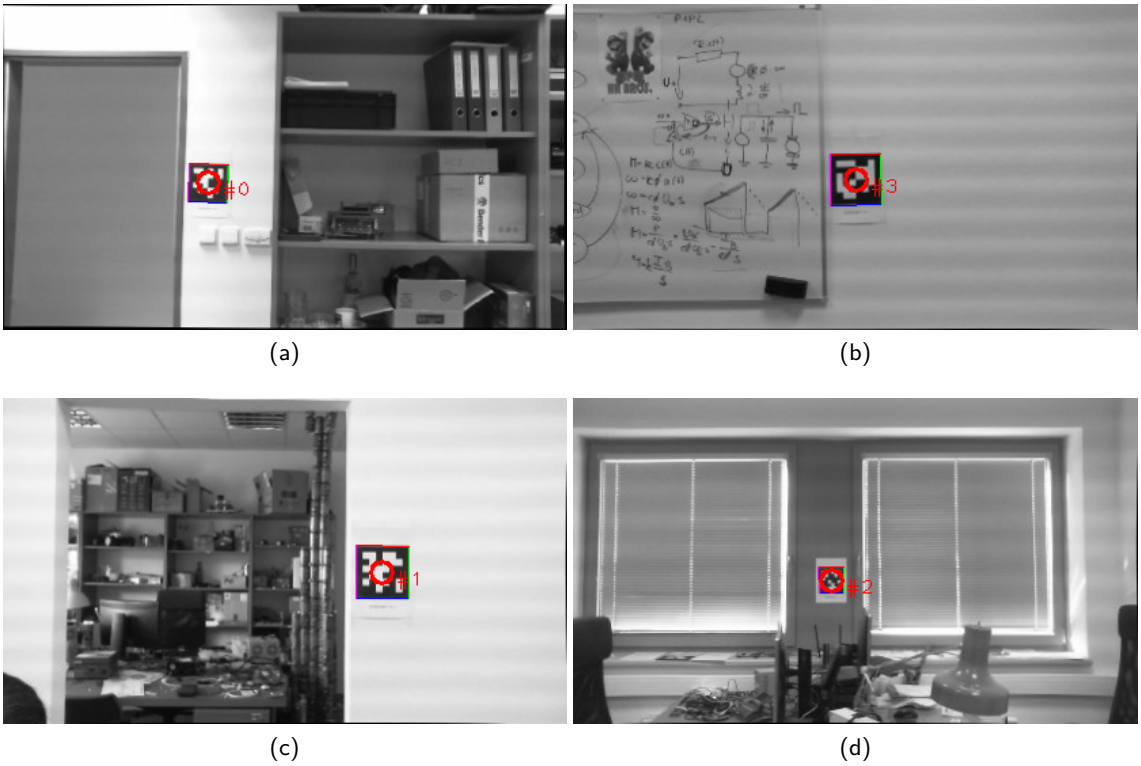
$$\mathbf{K}_f = \begin{bmatrix} 419.13 & 0 & 210.64 & 0 \\ 0 & 415.51 & 101.91 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.3}$$

Image undistortion and debayering performed by the `image_proc` package then uses the calibration profiles pubslihed to `camera_info` topics – based on the scenario, rectification can be done either in real-time during run-time or later during log playback when the performance is not real-time critical.

[8] `http://wiki.ros.org/image_pipeline/CameraInfo`

[9] `http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration`

[10] `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`



(a) (b)

(c) (d)

Figure 5.5: Extrinsic calibration images with AprilTag fiducial marks detected in the (a) front, (b) right, (c) left and (d) back camera stream.

Extrinsic calibration

Extrinsic calibration aims at estimation of attitude offsets that are primarily due to camera fixture imperfections; the result is then formalized as the rotation-translation matrix $[\mathbf{R}|\mathbf{t}]$ from Eq. 5.1.

As investigated by the analytical section of Chapter 3, artificial landmarks (or fiducial marks) offer a robust way of implementing such scenarios. Particularly, the AprilTag library [48, 69] packaged under ROS[11] has been employed based on their superior positioning accuracy.

The arrangement is captured by Fig. 5.5 and consists of fiducial marks suspended on walls in all four directions of the cameras; the room-referenced exact positions are known as well as the position of a pivoting axis on which the wearable platform is rotated. By extracting image coordinates of markers and knowing intrinsic camera parameters and positions of the fiducials relative to the camera frame, one is able to formulate a set of equations that can be solved using non-linear least squares optimization. The collection of scientific Python tools SciPy with its method `scipy.optimize.leastsq` implementing the Levenberg-Marquardt algorithm finds a good use here.

[11] `http://wiki.ros.org/apriltags_ros`

5.2 Localization

Based on data logged during experiments on the wearable platform into the ROS bag files, the localization algorithms were designed, developed and exercised in an off-line environment, allowing repetitive experiments with varying settings and easy comparison of each method performance on the very same data logs – see Chapter 6.

5.2.1 Off-line evaluation environment

MATLAB environment was used for its wide range of available functionality, either built-in or through add-on packages; namely, the Image Processing Toolbox was utilized for general image manipulation, while the Computer Vision System Toolbox provided tools for natural landmark processing (SURF).

An important part of the off-line environment was the Robotics System Toolbox with its extensive support for ROS: particularly, the ROS bag reading support was largely exploited to load data from experiments executed on the wearable platform and collected in the form of bags. While the toolbox comes with support for many standard ROS message types, there were custom messages logged as well. Fortunately, a Robotics System Toolbox add-on called "Robotics System Toolbox Interface for ROS Custom Messages"[12] is available and ready to import custom messages definitions.

5.2.2 General pose estimation framework

The whole recursive estimation framework has been custom implemented in MATLAB environment to have fine-grained control over the whole fusion process. The localization is based on local Cartesian map referenced to latitude-longitude global coordinates, so that integration of GNSS measurements is easily possible just by converting their lat-lon coordinates based on map the reference.

All parts of the data processing chain take ROS bags as argument which brings great flexibility in extraction of time-stamped measurements. The filters are accessed using a single functional entry point below referenced geenrically as `filterStep`, while in fact there is a separate function `ukfStep` for UKF and `particleStep` for PF. Employment of these functions is discussed together with the PDR interpolation implementation. Based on type of the filter, visualization methods have been established to help develop and illustrate the internal workings of the filters; Chapter 6 is largely built upon these.

Generic primitives allowing easy evaluation of the experiments were prepared as well – computation of fused trajectory errors compared to the reference route (perpendicular distance), error histograms plotting or error measures (MAE, RMSE, MaxAE) calculation.

[12] `https://www.mathworks.com/matlabcentral/fileexchange/49810-robotics-system-toolbox-interface-for-ros-custom-messages`

5.2.3 PDR data interpolation

Introduced in section 3.2.1, the multi-rate nature of the data fusion requires proper handling, in order to not introduce additional errors. The mechanism is presented in Algorithm 6; the behavior is driven by encountering a PDR data message (transferred as ROS message type `nav_msgs/Odometry` on topic `/stepwise_pdr`).

The measurements are processed retrospectively – they are buffered during the step phase between the PDR message arrivals, bounded by the interval $\left(t_i^{\mathrm{PDR}}; t_{i-1}^{\mathrm{PDR}}\right\rangle$. The buffer then for each GNSS or landmark measurement provides the time of arrival t_j, measurements vector $\mathbf{y}_j$ with its covariance matrix $\mathbf{W}_j$ and appropriate map of visible landmarks $\mathbf{map}_j^{\mathrm{visible}}$.

Algorithm 6 PDR retrospective interpolation mechanism (single PDR iteration)

1: $\Delta t^{\mathrm{PDR}} \leftarrow t_i^{\mathrm{PDR}} - t_{i-1}^{\mathrm{PDR}}$ ▷ determine last PDR step duration

2: **for** $j \leftarrow 1 : N$ **do**
3: $\quad \Delta t^{\mathrm{interp}} = (t_j - t_{j-1}) \ / \ \Delta t^{\mathrm{PDR}}$ ▷ compute the current time slice
4: $\quad \Delta\mathbf{u} \leftarrow \mathbf{u} \cdot \Delta t^{\mathrm{interp}}$ ▷ assemble the measurements vector and covariance
5: $\quad \Delta\mathbf{V} \leftarrow \mathbf{V} \cdot \Delta t^{\mathrm{interp}}$

6: $\quad$ **filterStep**$(\mathbf{x}_{i,j}, \mathbf{P}_{i,j}, \Delta\mathbf{u}, \mathbf{y}_j, \Delta\mathbf{V}, \mathbf{W}_j, \mathbf{map}_j^{\mathrm{visible}})$

7: $\quad \mathbf{u} \leftarrow R(-\Delta\mathbf{u}^{\theta})(\mathbf{u} - \Delta\mathbf{u})$ ▷ prepare u for next iteration - need to rotate it back
8: $\quad \mathbf{V} \leftarrow \mathbf{V} - \Delta\mathbf{V}$ ▷ estimate remaining process noise
9: **end for**

10: **filterStep**$(\mathbf{x}_i, \mathbf{P}_i, \mathbf{u}, null, \mathbf{V}, null, null)$ ▷ apply remaining u without measurement

Lines 3–5 form the first part of the interpolation, where the proportion of PDR advance is linearly driven by the particular measurement's time distance $\Delta t^{\mathrm{interp}}$ from the previous measurement; the delay is normalized by Δt^{PDR} so that all sub-steps sum up to 1. The interpolation is then finished on lines 7–8 which prepare the remaining measurements vector and covariance matrix for the next iteration. The only non-trivial operation is on line 7; as described in section 3.2.1, the rest of the PDR posse difference vector needs to be rotated about z-axis to counter the partial rotation done by the heading part of $\Delta\mathbf{u}$.

Function `filterStep` is the main entrypoint of the UKF or PF engine, accepting all inputs and producing an updated state vector with its covariance matrix. It is called from two places – first, on line 6 it processes the intermediate PDR step fragments with their respective sensor measurements. Finally, `filterStep` is called again on line 10 to gather the remaining part of the PDR pose difference – this time, there is no measurement coupled to this instant, so that the filter only performs the prediction step without correction as analyzed in section 3.1.7.

Figure 5.6: An example of the structure of Open Street Maps.

5.3 Global path planning

The global planner is a module whose functionality is mainly concentrated on path planning – the execution is then rather simple, since the segment-wise advance is easily traceable (unlike local planning which is intimately close to the localization subsystem). For this reason, only the path plan preparation was subjected to implementation and later exercised.

The Python environment was preferred to MATLAB because of a better availability of convenient tooling and libraries, starting with map source parsing and ending with graph search algorithms. Python is also a strategic choice in this case since it is one of the native languages supported by ROS.

Map sources Among possible map sources, the Open Street Maps project was chosen for a multitude of reasons – primarily the sensible level of details even providing sidewalk segments combined with the crowd-sourced nature enabling future custom additions. A representative example of the types of information typically captured in OSM can be seen in Fig. 5.6; not only the aforementioned sidewalks, but, importantly for this thesis, bus/tram stops are modeled as well.

Packages `osmgraph` & `networkx` While the OSM data contain an abundance of useful features, it was quickly discovered that its XML representation was not selected in a particularly planning-friendly way – the segments have no junctions defined, public transport lines are hard to extract etc. On that account, the `osmgraph` package[13] was utilized to help process the map files into a graph-like structure. The graph can be then searched using the popular `networkx` library[14].

[13] `https://github.com/Mapkin/osmgraph`
[14] `https://networkx.github.io/`

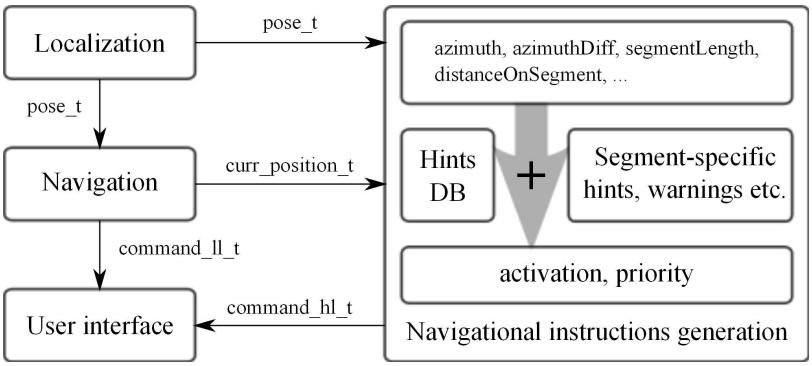


Figure 5.7: Simplified message flow of the fuzzy-logic driven subsystem of navigation instructions.

5.3.1 Instructions generation

The instructions subsystem was implemented in Java on the first-generation platform powered by LCM; in Fig. 5.7, there is a simplified diagram showing the message types relevant to the generation process. The design and experiments were done in simulated environment, replacing the localization, navigation and user interface modules by their deterministic counterparts with the same inter-process communication interface.

To implement the fuzzy logic execution core, the library Fuzzy Engine for Java[15] was utilized. It provides primitives for all necessary parts of the inference chain: class `LinguisticVariable` encapsulating the linguistic variable with its values and respective trapezoidal membership functions; class `FuzzyState` that holds the context of the inference; a `RuleBlock` containing parsed `FuzzyRule`s and the functional interface to execute them. The supported rule syntax was introduced in the design section; additionally, the following hedges are available: `not`, `very`, `somewhat`.

The operation of the module then runs according to Algorithm 7. The call-back function `segmentChanged` is bound to the messaging system and launches planning of the instructions every time a new segment is on; the rest starting from line 13 runs repetitively and gradually sends the instructions to UI layer once they are ready.

[15] `http://fuzzyengine.sourceforge.net/`

Algorithm 7 High-level fuzzy logic instruction generation pseudocode

```
1: function segmentChanged(newSegment)
2:     save segment properties, compute derived parameters
                                                    ▷ plan execution of instructions:
3:     create new context within FuzzyState
4:     set input values to all linguistic variables and fuzzify
5:     copy all instructions from database locally
6:     for each instruction on local list do
7:         execute fuzzy activation rules
8:         if activation denied then
9:             remove instruction from the list
10:        end if
11:    end for
12: end function
                                                    ▷ the main execution thread:
13: while true do
14:    create new context within FuzzyState
15:    set input values to all linguistic variables and fuzzify
16:    for each instruction on filtered list do
17:        execute fuzzy timing rules
18:        if activation denied then
19:            remove instruction from the list
20:        else
21:            execute fuzzy priority rules
22:        end if
23:    end for
24:    send prioritized instructions to user interface
25:    sleep for some time
26: end while
```

Chapter 6

Evaluation

For a work concerned with applied research and development, experimental evaluation of the designed principles is an integral part of the proposed solution.

6.1 Localization

Performance of the designed localization chain was verified by a number of experiments exercising the build-up of localization sources contribution. For all below presented localization experiments, the second-generation platform was used, utilizing gradually all of the on-board sensory equipment.

Workflow of the experiments was similar in all cases: sensory outputs logging during pedestrian execution of the test routes by means of the wearable platform; then, off-line processing and evaluation. The results were compared to the reference path and error of each estimated position was determined as perpendicular distance to the reference path. The resulting error set was processed by the usual performance measures – Mean Absolute Error (MAE), Root-Mean-Square Error (RMSE) and Maximum Absolute Error (MaxAE).

6.1.1 Test routes

Three different test routes were established in order to emulate both specialized, rather worst-case scenarios and representative segments close to real-world environment.

Test polygon Intentionally condensed space of this rectangular route leads to amplification of the known GNSS positioning mis-behavior. This path was conceived as benchmark case for the GNSS-PDR fusion step – its sharp corners are a good exercise for fused trajectory faithfulness evaluation.

Loop route Route starting and ending in one spot, partially open, partially covered by tree canopies. There are several representative situations contained within this route: both

short and long range to nearest obstacles create varying conditions for landmark tracking; tree canopies then locally deviate GNSS estimates.

Urban route The heaviest deterioration in GNSS performance is most often brought by the urban environment with its efficient reflective surfaces, introducing multi-path signal travel effects. The "urban" test route was conceived to exercise the researched methods from this very point of view; formed by a pathway between two buildings, there are ample reflections, signal shields and the trajectory even features a loop around a tree to magnify GNSS problems. The GNSS data rendered in section 6.1.4 show that this intention has indeed been fulfilled.

6.1.2 Unprocessed sensory data

To illustrate imperfections and limitations of major sensory data sources, that were first outlined theoretically in Section 3.1, Fig. 6.1 provides typical outputs of the Osmium MIMU22BL PDR unit (steps accumulated in Fig. 6.1a), Navilock NL-602U GPS receiver and Xsens MTi-3 AHRS unit (Fig. 6.1b) on the loop route.

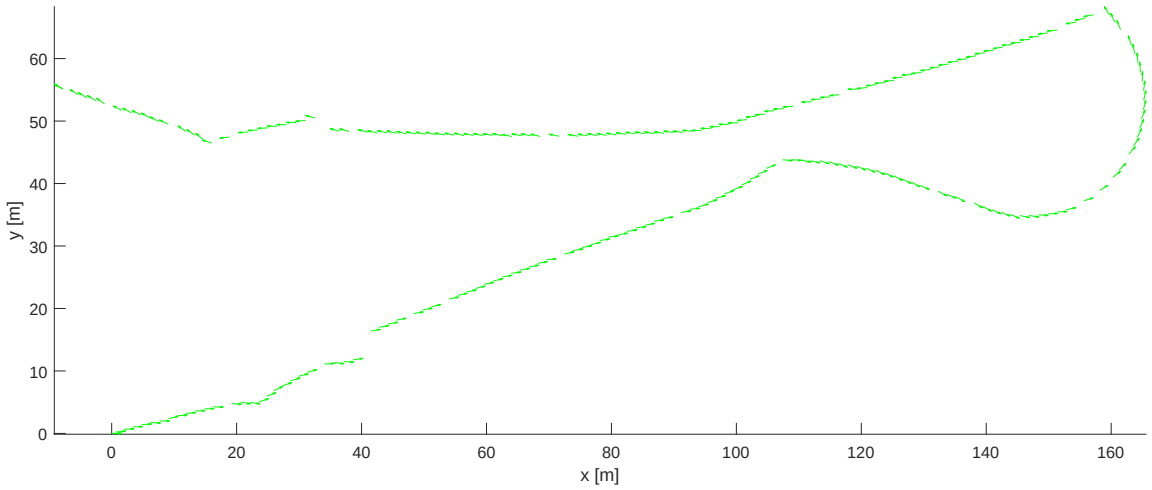
PDR performance has been generally found fantabulous, especially on straight or sharp-cornered trajectories; behavior captured in Fig. 6.1 has been observed on larger-radius curves. In any case, it is clear that the residual yaw drift needs to be compensated. *GNSS* usually performed within 2-3 meters in less covered terrain, while vicinity of even few-story buildings threw it out of reference path almost order of magnitude more. *AHRS* yaw estimates rendered as blue arrows show considerable drift – despite being hard- and soft-iron compensated for the particular magnetic conditions of the wearable platform, absolute north-referenced yaw output was not stable. A two-point (offset & trend) compensation was adopted as a countermeasure for mapping purposes.

6.1.3 GNSS-PDR fusion evaluation

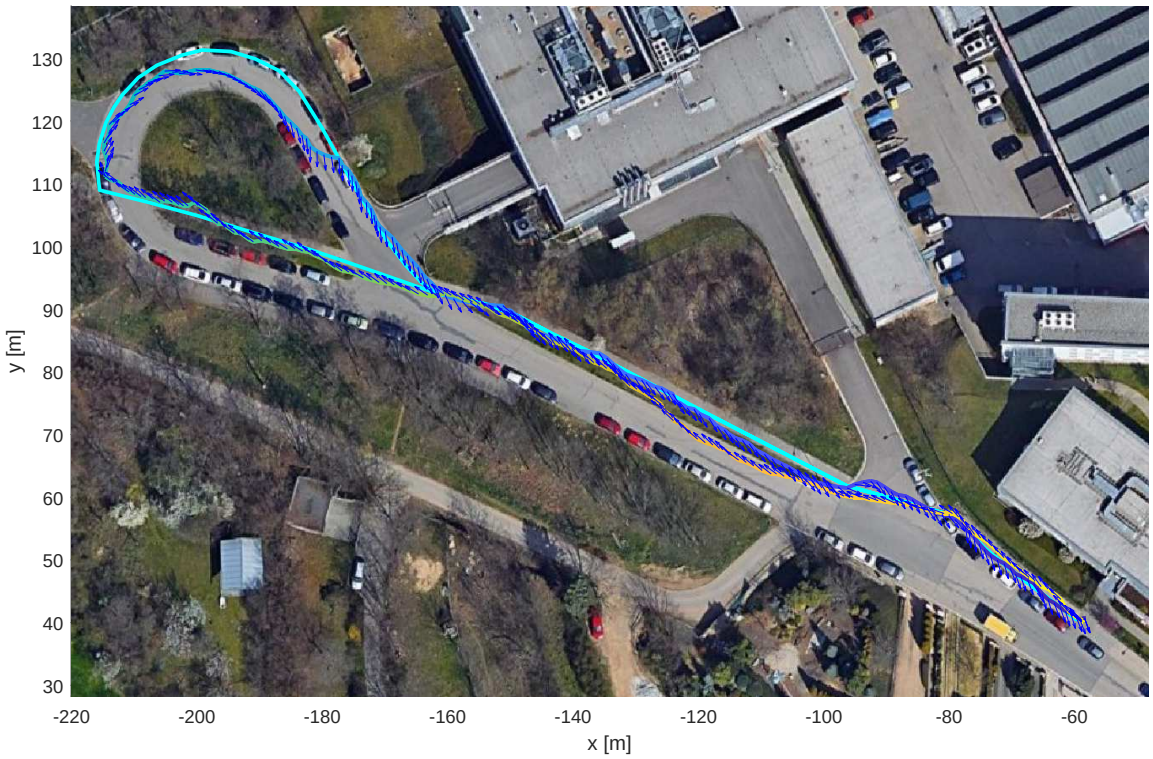
The first stage of the localization mechanism is an integration of the precise step-wise position differences produced by PDR. The motivation and expected outcome are improvements in estimated trajectory – reduction of local deviations and more faithful behavior during sharp turns. The fusion process is driven by the PDR messages; this introduces aperiodic behavior, which is fairly unusual in the realm of data fusion. It does not, however, bring any numerical deterioration of results. Tables 6.1 and 6.2 summarize the values for the GPS-only, UKF and PF cases.

Test polygon experiment

The primary experiment on GNSS-PDR fusion stage is the rectangular test polygon route. This benchmark was executed 15 times during several days at random times to sample from varying GNSS satellite orbital configurations causing varying DOP.



(a)



(b)

Figure 6.1: Visualization of raw sensory data on the loop route. (a) PDR data, step-wise increments simply accumulated – a significant divergence in the heading angle can be observed; (b) GPS data begin blue and end yellow, blue arrows denote the non-processed AHRS data.

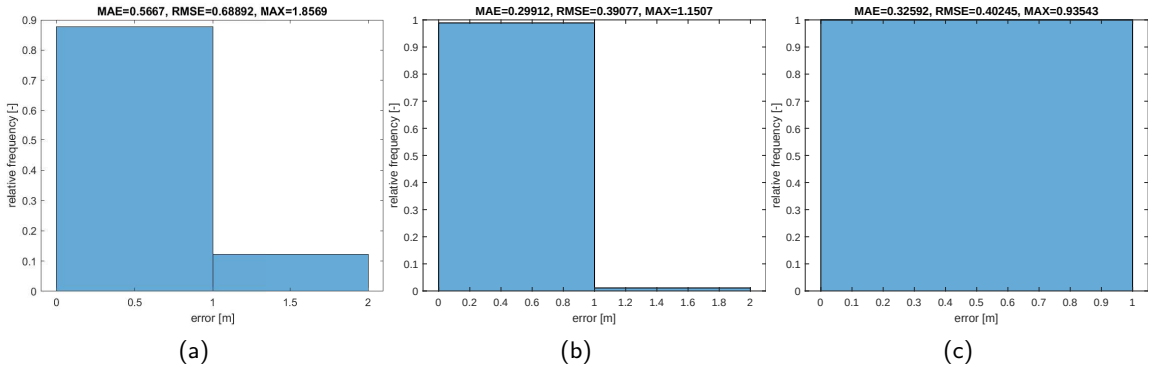


Figure 6.2: Histograms of position error on test polygon for (a) pure GPS data, (b) UKF fusion and (c) PF fusion.

Table 6.1: GNSS-PDR position estimation errors on test polygon related to Fig. 6.3.

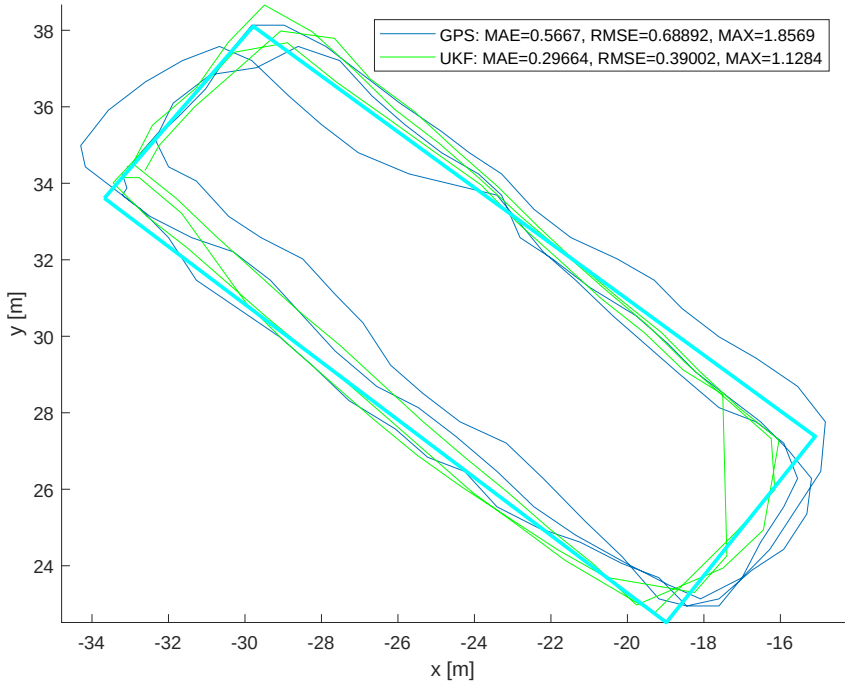
	MAE [m]	RMSE [m]	MaxAE [m]
GPS only	0.5667	0.6882	1.8569
UKF fusion	0.2966	0.3900	1.1284
PF fusion	0.3593	0.4378	1.0961

From this set of experiments, one representative instance has been selected for a detailed showcase. The whole set was then analyzed to elaborate the measures on a significant number of executions.

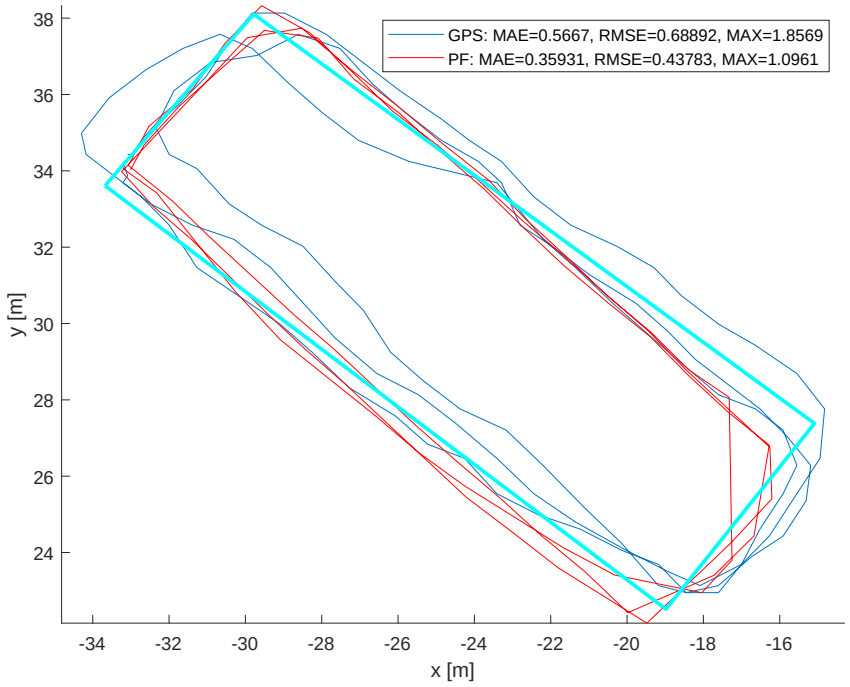
Single execution details Three full circles were logged around this path. It can be seen from both Fig. 6.3 and Table 6.1, that fusing PDR data with GNSS estimates leads to considerably tighter, repeatable results. UKF and PF perform here roughly on par, bringing down both the MAE and RMSE metrics approximately by 40 %. Figure 6.4 then adds further information to illustrate the data fusion:

- Uncertainty in the position component of the system state is visualized by red ellipses that denote boundaries of single-sigma covariance.
- Tracking of GNSS location estimate to the fused position is given by the red arrows.
- Estimated heading is indicated with the step-wise resolution.

It can be seen that the initial position uncertainty has been chosen greater in case of the particle filter, because the initially generated particle set has to safely encompass the GNSS uncertainty into a grater extent – otherwise, the measurement update is not effective and the system is prone to particle deprivation. After that, the expressed uncertainty stabilizes on a comparable level as can be concluded from the size of covariance ellipses near end of the motion.

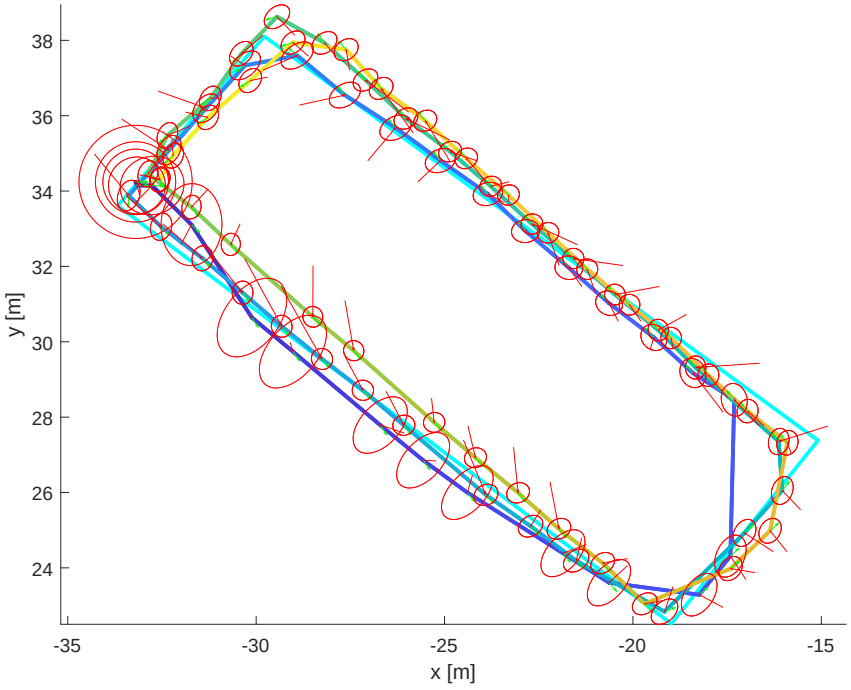


(a)

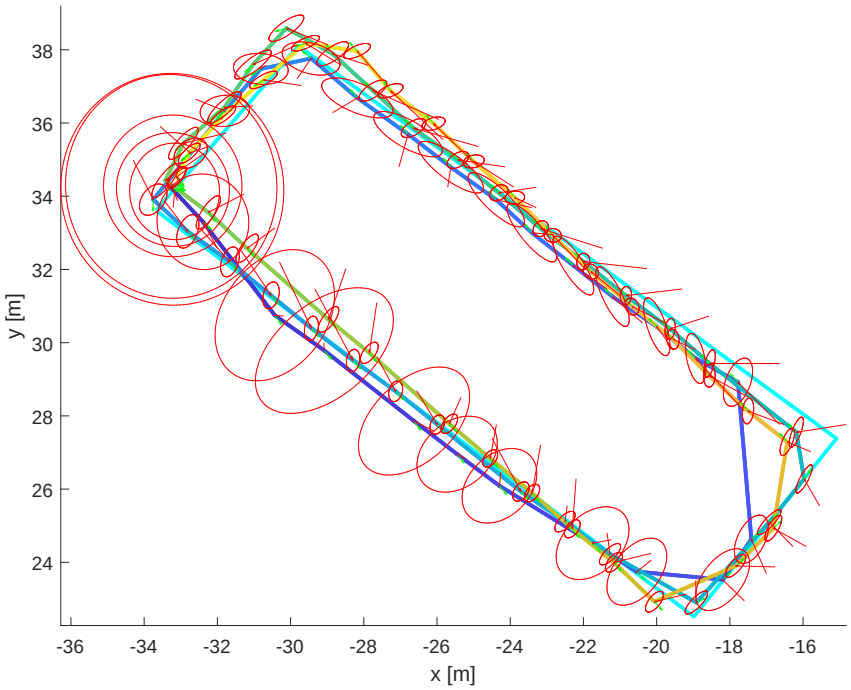


(b)

Figure 6.3: Visualization of estimated paths of three rounds around the test polygon for GNSS-PDR fusion. The reference path is shown as thick cyan line; pure GPS data are blue. (a) UKF fusion result is plotted in green; (b) PF fusion result is plotted in red.



(a)



(b)

Figure 6.4: A different view on the same results as shown in Fig. 6.3. Fused path begins blue and end yellow. Green arrows denote right foot direction, red arrows point from GPS measurement to fused position. Red ellipses represent $\mu \pm \sigma$ equivalent covariance of position part. (a) UKF fusion result, (b) PF fusion result.

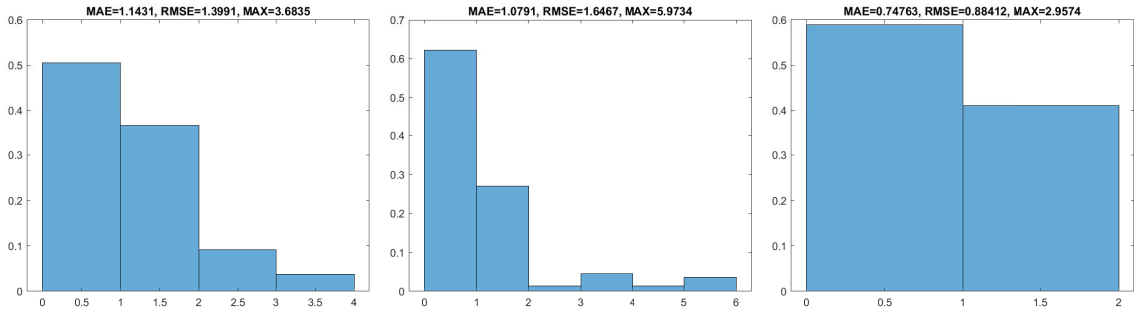


Figure 6.5: Histograms of position error on loop route for (a) pure GPS data, (b) UKF fusion and (c) PF fusion. Deterioration in UKF is pronounced; however, as explained, only due to the initial convergence.

Multiple passes, average behavior Processing the aforementioned 15 polygon execution data sets, it has been found that the GNSS-PDR sensor fusion improves the worst-case error figures in 100 % cases, in each case to a slightly difference extent in the range of 25–45 %. At the same time, MAE and RMSE metrics show improvements to a similar degree. The reference path in this experiment is understood as shape standard rather an absolute positioning reference; since there is no other geo-referenced data source, the actual GNSS positioning accuracy determines *offset* type of error which, however, cannot be compensated by PDR addition alone.

Loop test route experiment

Bringing more real-world environment to the test, the loop route guides the pedestrian on a sidewalk and, similarly to the other test routes, begins and ends in out spot to clearly show accumulated errors. Fusion of PDR data into GNSS here brings improvements both in numerical error measures in Table 6.2 and in smoothness of the fused trajectories, while sharp corners are improved as well (see Fig. 6.6a).

Particle filtering wins according to error measures compared to UKF, based on reasons elaborated below. All in all, the GNSS-PDR fusion brings considerable improvements in MAE and RMSE metrics, again with up to 40% improvements.

Initial convergence Since the initial state estimate $\hat{\mathbf{x}}_0$ is gained from the first GNSS fix only, the attitude information is missing and is initialized to zero, i.e. along x axis. The UKF case then suffers from slower initial convergence of the attitude as can be seen in Fig. 6.6a, while the multimodal character of PF allows to estimate the attitude almost instantaneously (Fig. 6.6b). When this initial phase is removed from the trajectory error evaluation, UKF results are more or less comparable to PF performance.

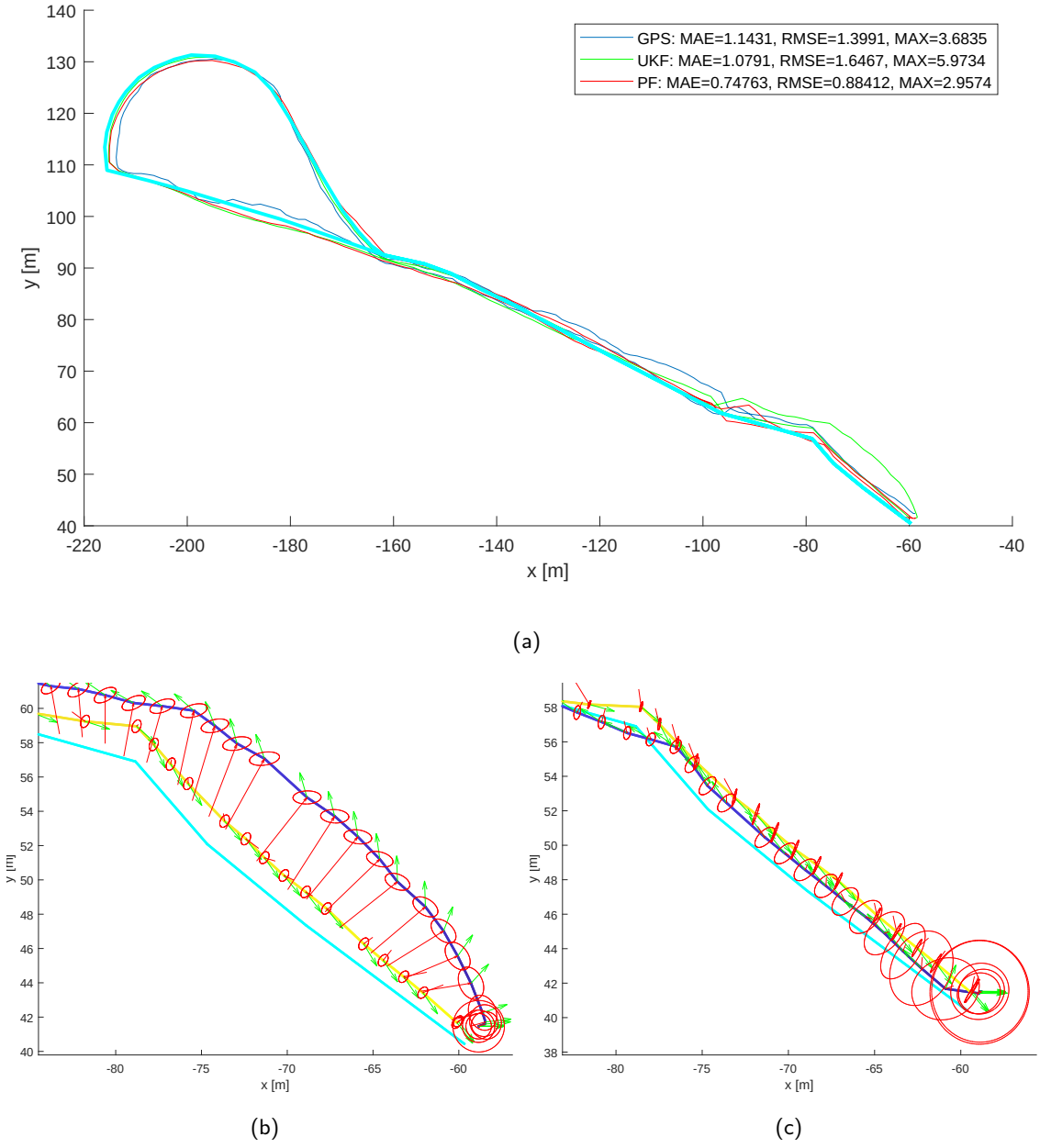


Figure 6.6: Estimated paths on loop test route. The reference path is shown as thick cyan line, fused path begins blue and end yellow. Green arrows denote right foot direction, red arrows point from GPS measurement to fused position. Red ellipses represent $\mu \pm \sigma$ equivalent covariance of position part. (a) Overall situation; (b) UKF fusion result, magnified start and end; (c) PF fusion result, magnified start and end.

Table 6.2: GNSS-PDR position estimation errors on loop test route related to Fig. 6.6.

	MAE [m]	RMSE [m]	MaxAE [m]
GPS only	1.1431	1.3991	3.6835
UKF fusion	1.0791	1.6467	5.9734
UKF fusion without start	0.8115	0.8765	3.8574
PF fusion	0.7476	0.8841	2.9574

6.1.4 Landmarks utilization evaluation

During all experiments, the complete sensory system was logged for further processing – so that the above presented experiments could be processed with the landmark-augmented localization mechanisms again and compared. Since the PF framework was not utilized for the landmark estimation, only the UKF results will be directly comparable.

Mapping phase

The data to be examined first are results of the mapping phase. Starting with individual identified landmark sequences, 6.8 showcases a typical urban landmark correctly matched in 10 subsequent camera images; Fig. 6.9, on the other hand, depicts a sequence with first two mismatched features. Fig. 6.10 then shows the landmarks localized in map using the generalized triangulation mechanisms described in section 3.2.2 – blue color represents the former one and red-colored markers belong to the latter one, with two mis-identified features easily distinguishable.

Figure 6.11 demonstrates features extracted and processed by the landmark mapping mechanism that can be categorized into the following bins:

Urban Exercised in Fig. 6.11a, urban landmarks are constituted by distinct architectural patterns, which makes them very stable in terms of repeatable detection.

Natural One of the best contrasting part of the surrounding during majority of day is the skyline; coincidently, the skyline contours also tend to be relatively stable landmark sources (Fig. 6.11b).

Mixed Composed typically by a dark tree trunk on bright artificial background, the combination produces strong features that are fairly often encountered around footpaths, as Fig. 6.11c shows. Together with the natural landmarks, the triangulated position is often "virtual" – the landmark is formed by projection of spatially separated objects and the mapped position may not be coincident with them.

Non-stationary Inevitably, a part of the detected features belong to objects that move within or even disappear from the scene (mostly cars, given the target environment). Disappearance is the easier case – the feature will simply not be active and "just"

consumes resources. Moving the feature bearing object is then comparable to a feature mismatch and needs to be mitigated by employing proper localization mechanisms.

"Shine&shadow" SURF detector is prone to give priority to temporary patches of light, especially pronounced during sunny days, when the limited dynamic range of cameras creates distinct hot spots and deep shadows, that are very appealing for the SURF detector.

Reproducible observability of such features is too often influenced by a particular position of the Sun in the sky, extent of cloudiness and such nuances.

Discussion It has been found that landmarks closer than a few meters often do not pass the sequence detection conditions – the mapping pedestrian simply passes them too quickly to be registered on a significant number of subsequent images. This fact was assessed a minor limitation only; a change of camera frame rate may help solve it (which was not possible on G2 platform). Front and back cameras also often produced image sequences hard to process into any meaningful map; the small span of angles towards the landmarks made the subsequent landmark localization diverge in many cases and limit the map to only a few hits. Variations in sunshine intensity had different impact on each landmark type, as discussed above; urban landmarks belong to the most stable ones, which is encouraging since urban is the target environment.

In the mapping process as it was designed, AHRS-based yaw correction plays a major role – even though the mapping was performed walking deliberately calmly and trying to minimize body swing, yaw oscillates more than ±5 degrees which is enough to deteriorate the triangulation significantly. If needed, AHRS could be eliminated from the system and replaced by heading change estimation from front/back cameras, as shown in Fig. 6.7; from the data throughput and computational resources point of view, AHRS is much more economical though.

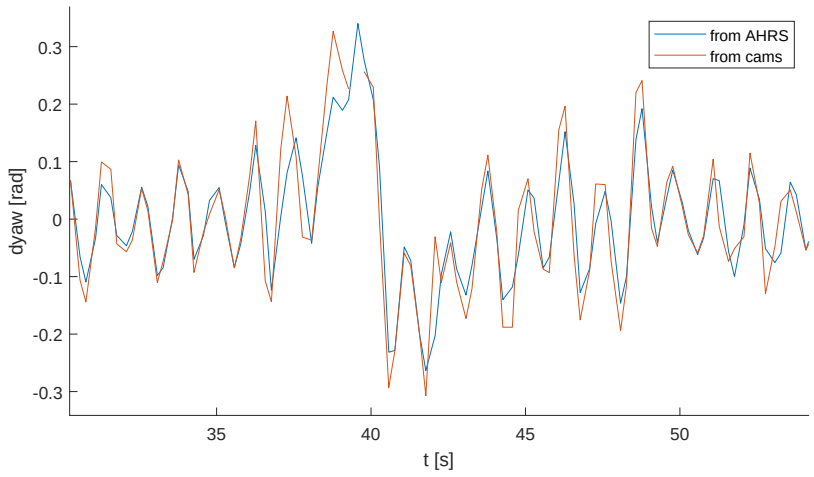


Figure 6.7: Comparison of heading change extracted from AHRS and front/back cameras.

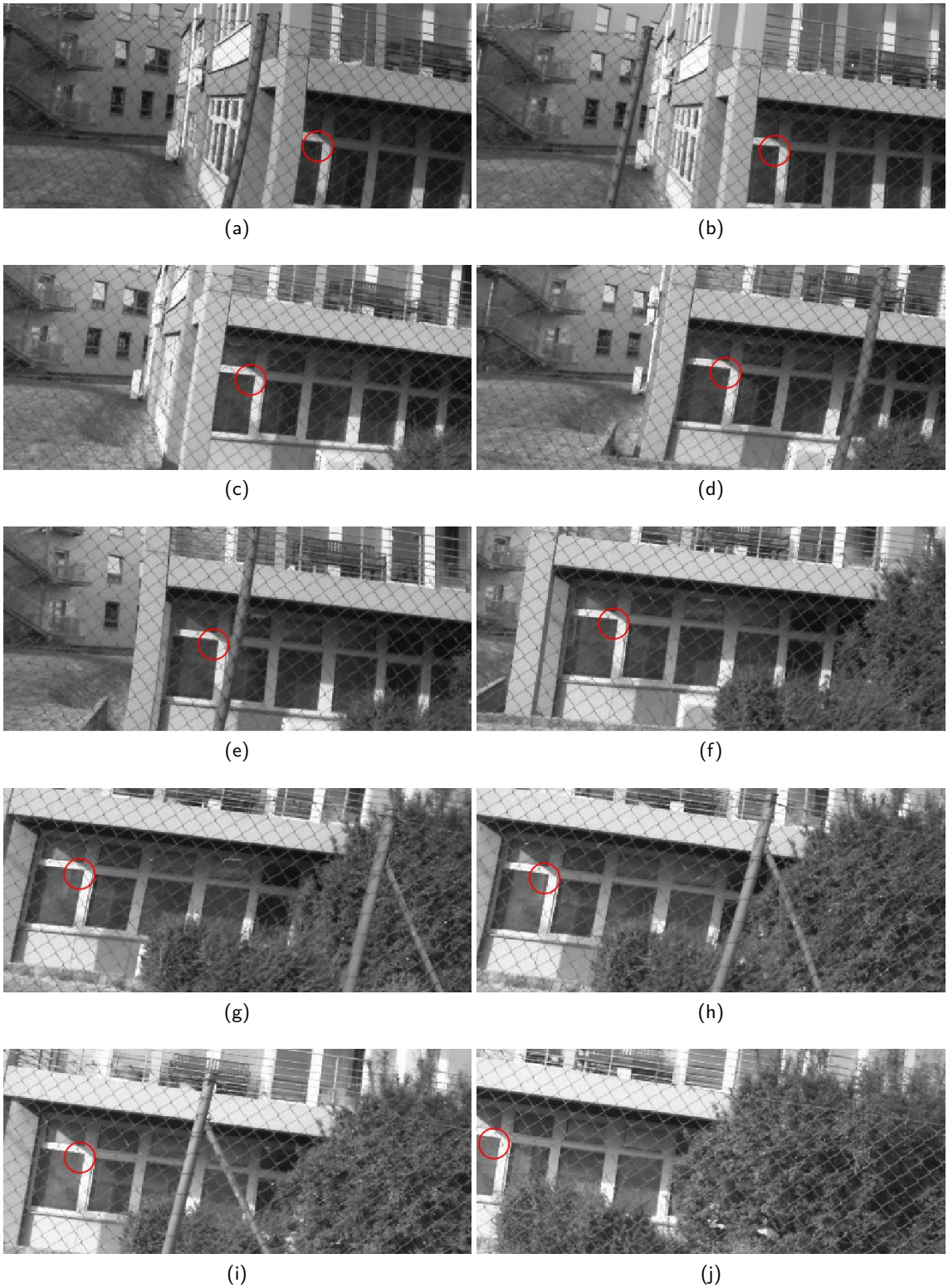


Figure 6.8: Sequence of images from left camera with the same urban landmark position detected in all of them, marked by the red circles.

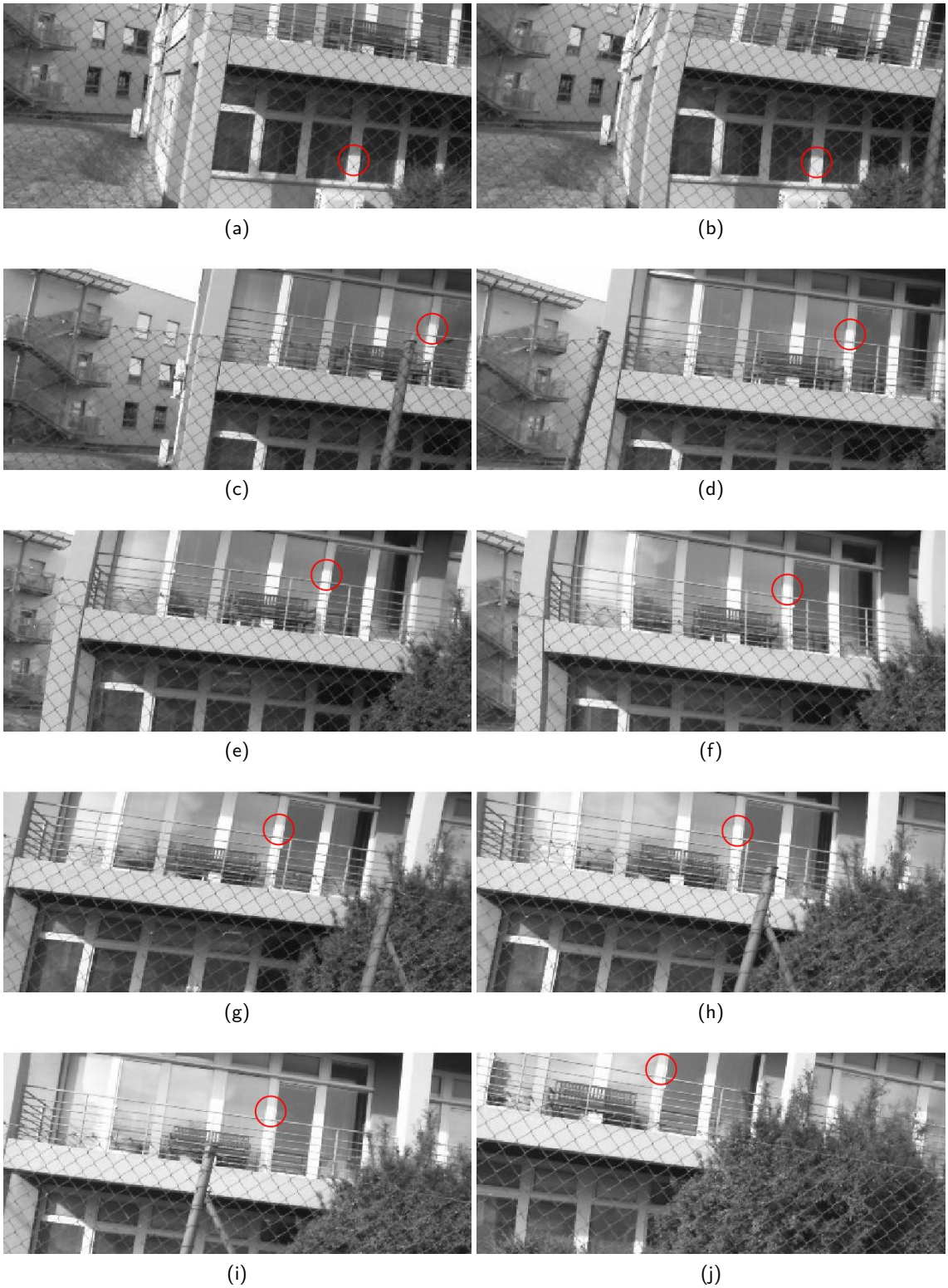


Figure 6.9: Sequence of images from left camera with first two landmarks mismatched to the rest – a clearly similar feature of the building led to confusable SURF descriptors.

Figure 6.10: Urban landmark localized from the sequence of images in Figure 6.8 – blue arrows point from the place of image acquisition in the extracted direction of the landmark. Blue cross represents the triangulated location of the landmark in the map. Similarly, red-colored markers belong to landmarks shown in 6.9 with the two mis-identified landmarks aiming more to the left. Other colors denote yet another landmarks.

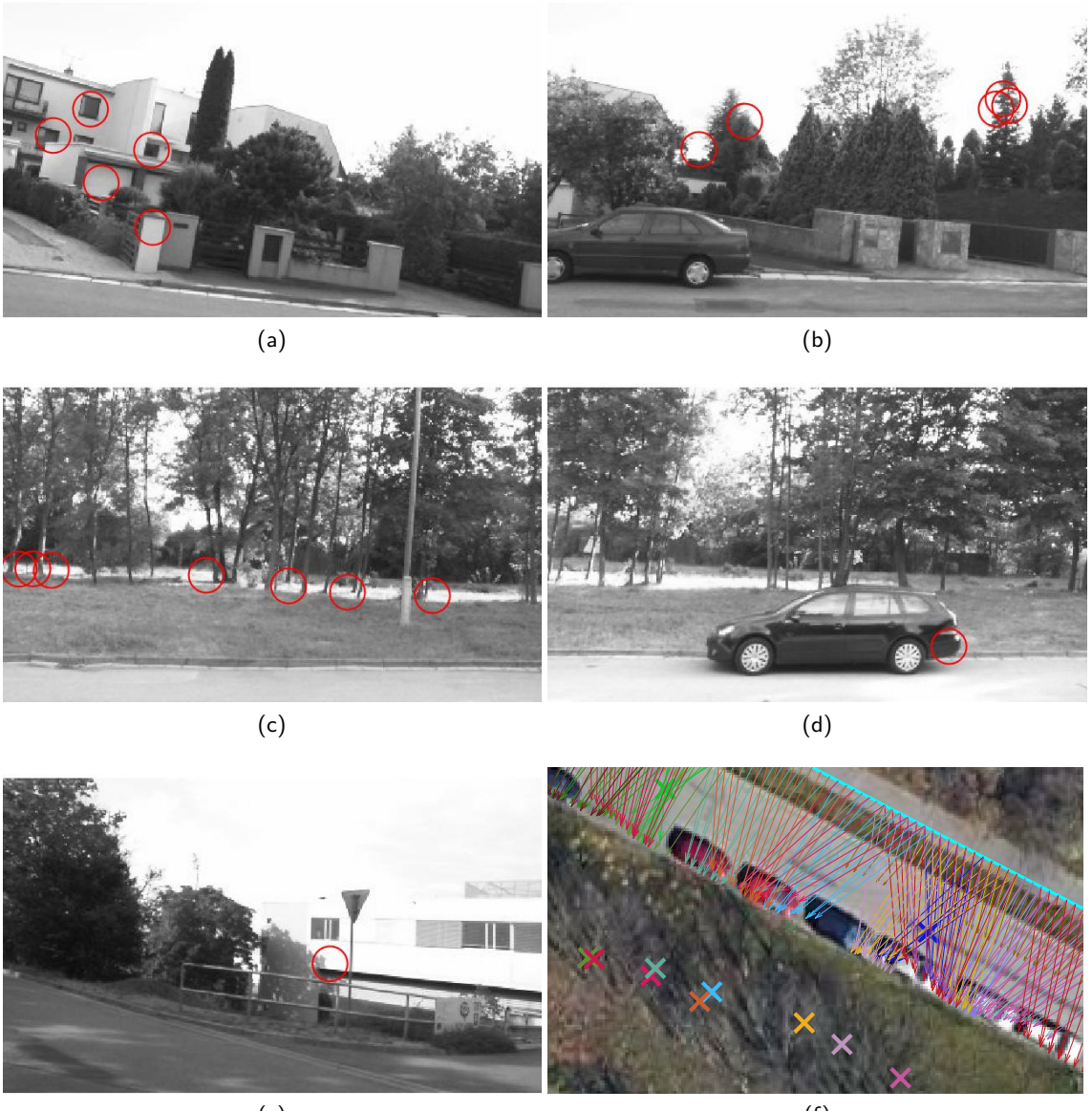


Figure 6.11: Samples of different type landmarks: (a) urban, (b) natural, (c) mixed, (d) non-stationary and (e) "shine&shadow". Figure (f) shows a map of the localized mixed landmarks from image (c) – a tree line next to the road.

Localization phase

As a rule during the landmark-augmented localization experiments, landmark map was processed on one of the data sets and localization pass was executed on other data sets using this map, so that the excessively optimistic resubstitution case was ruled out. Otherwise, the execution and methodology of this batch of experiments was exactly the same as during the above introduced cases.

Loop test route experiment Further exploiting the loop route experiment outlined in section 6.1.3, the existing data sets which already contained the whole sensory outputs logged were utilized. The immediately observable improvement of the additional source of geo-referenced information is instantaneous initial convergence of heading – the difference is striking, comparing Fig. 6.6b and Fig. 6.12b. This gratifying result was then one of the reasons why the landmark augmentation mechanism was only adopted to the UKF framework; with the major PF benefit – initial convergence – equalized by landmark employment, there was not that much motivation left for its employment[1].

Urban test route experiment Employment of GPS-PDR-landmark fusion has reduced maximum deviation from 8.45 m (GPS-only) to 2.76 m as can be seen in Fig. 6.13. Although pure GPS performs well at the trajectory beginning, it worsens considerably during the loop in the middle, practically washing out the trajectory shape; the return path then evinces a complete divergence from the route. The fused trajectory preserves shape details rather well; its MAE of 1.52 m and RMSE of 1.63 m then improve respective GPS-only figures by ca. 22 % and 37 %.

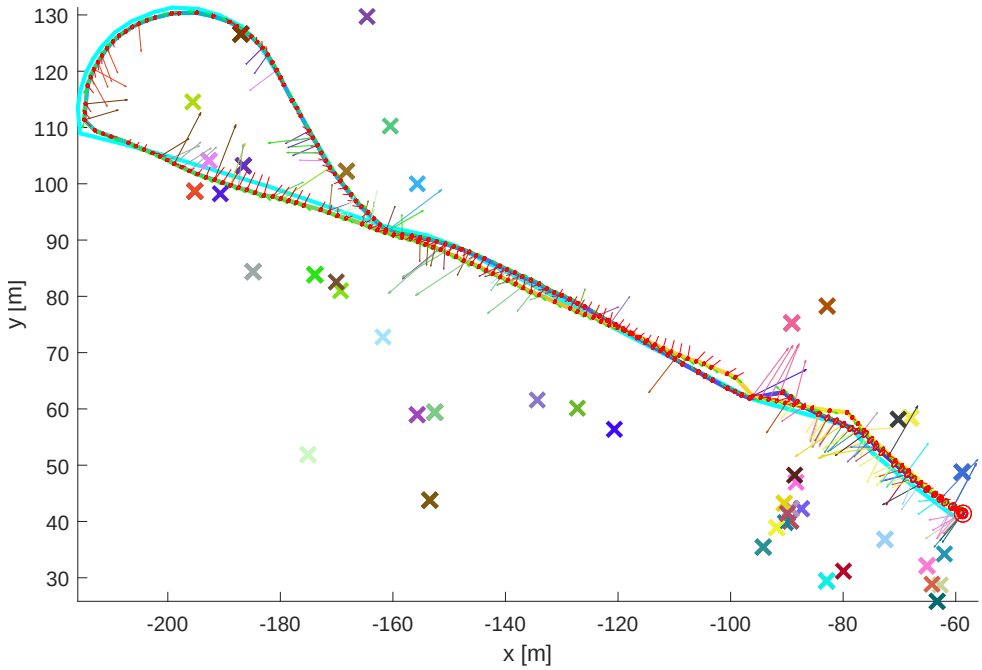
[1]The random nature of PF made experiment evaluation uneasy in some cases, when just repeating the computation on exactly same data set led to slightly (and sometimes largely) different results.

Table 6.3: GNSS-PDR-landmark position errors on loop test route related to Fig. 6.12.

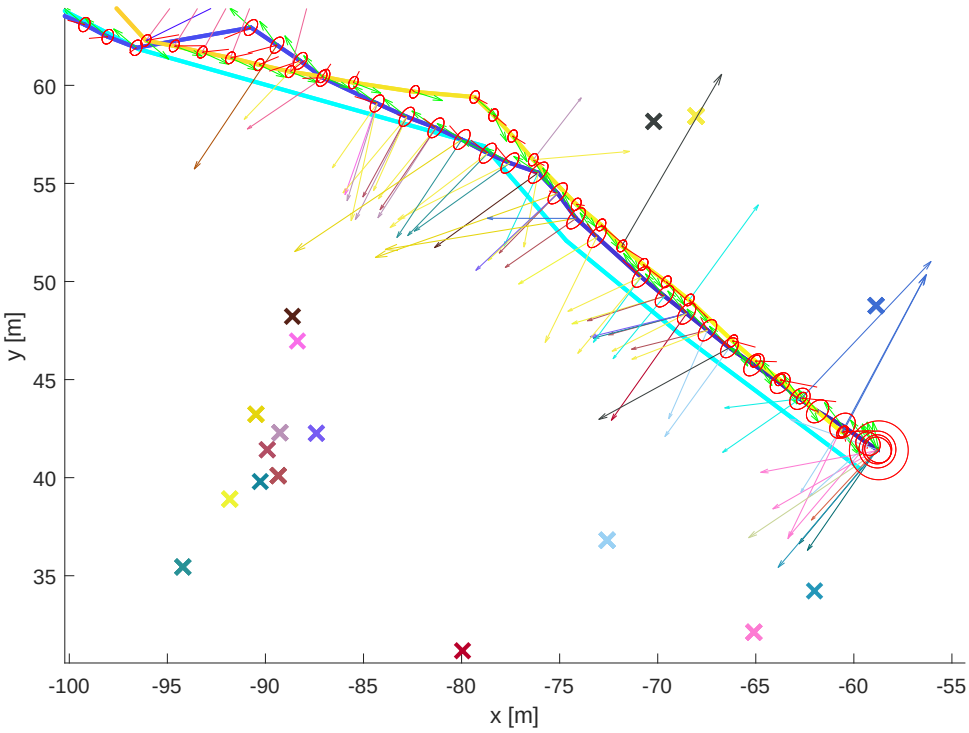
	MAE [m]	RMSE [m]	MaxAE [m]
GPS only	1.1431	1.3991	3.6835
GNSS-PDR, UKF	1.0791	1.6467	5.9734
GNSS-PDR-landmark, UKF	0.8699	1.0043	2.6077

Table 6.4: GPS-only versus full GPS-PDR-landmark fusion position estimation errors on urban test route related to Fig. 6.13.

	MAE [m]	RMSE [m]	MaxAE [m]
GPS only	1.9418	2.6104	8.4491
GPS-PDR-landmark fusion	1.5194	1.6328	2.7581

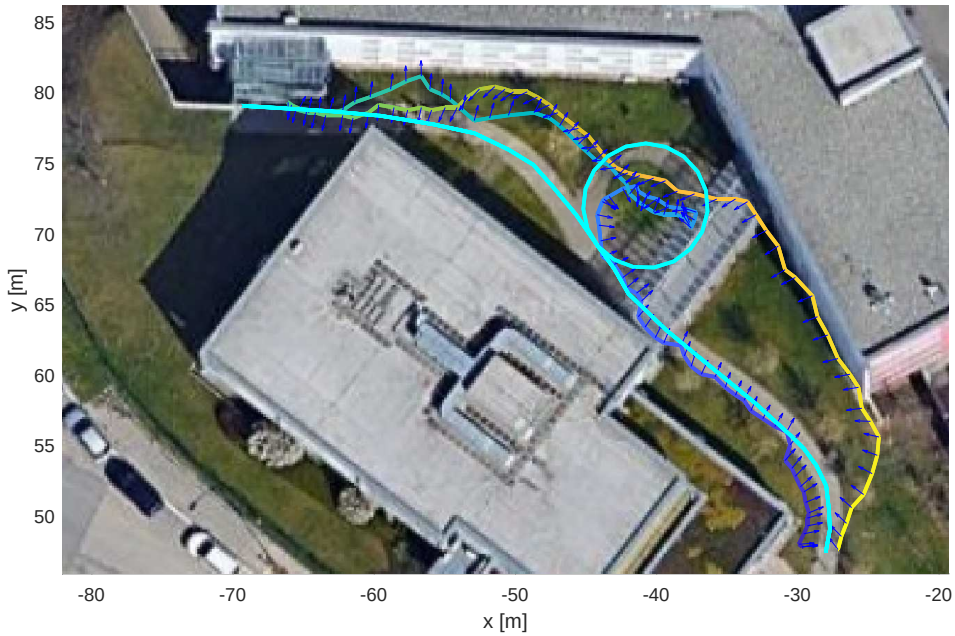


(a)

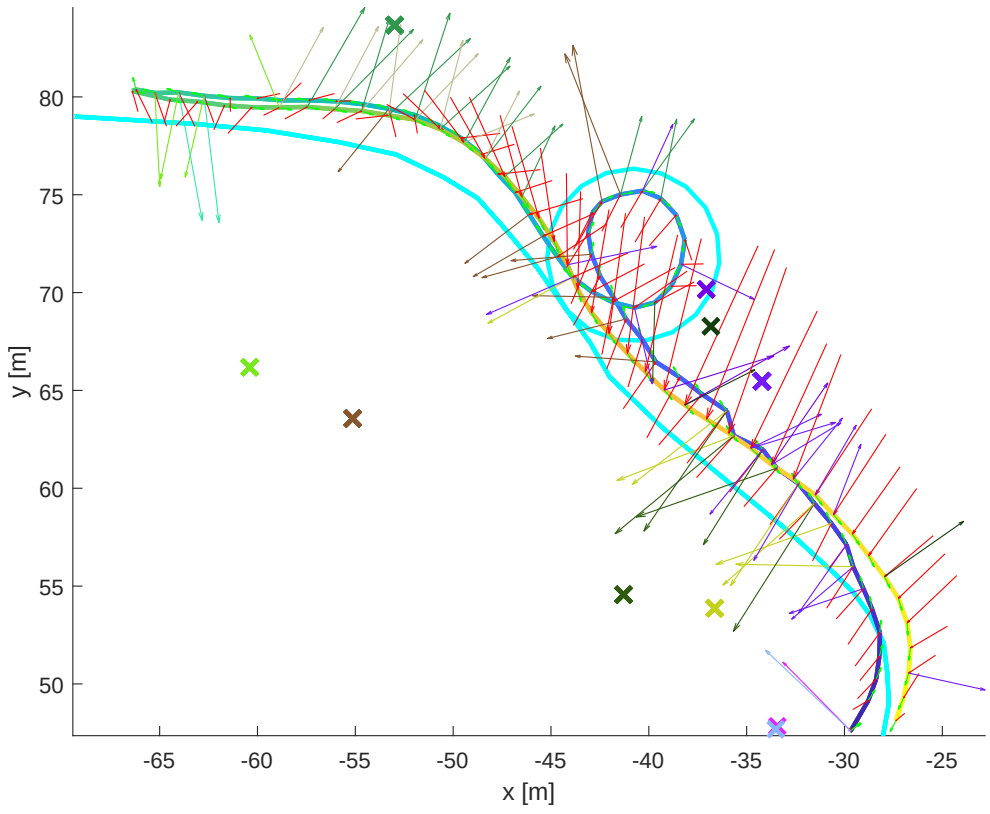


(b)

Figure 6.12: Result of bearing-only landmarks shown in Fig. 6.10 and Fig. 6.11 fused together with GPS and PDR data within the UKF framework; left camera visualized only for understandability. (a) Overall situation; (b) Magnified start and end – dramatic improvement of initial heading convergence is apparent (compare to initial convergence of UKF fusion without landmark aid in Fig. 6.6b).



(a)



(b)

Figure 6.13: Landmark-augmented fusion of PDR and GNSS on the urban route. (a) Reference trace is shown in cyan; GPS-only path beginning blue and ending yellow evinces heavy multi-path effects; (b) Fusion result – red arrows pointing from GPS measurement to fused position indicate how considerable the fused trajectory improvement really is.

Overall experimental measures By assessing 8 different data sets evaluated according to the mapping/localization data set separation rule, it has been found that the employment of natural landmarks, fused together with GNSS and PDR sensory data, brings worst-case error down in 100 % cases by 42 % on average, compared to GNSS only case (generally lower improvement percentages were observed on the loop route where the GNSS is not obstructed to such extent). At the same time, MAE and RMSE measures evinced decrease in the range of 20–40 %. Compared to the GNSS-PDR fusion, the landmark augmentation mechanism has brought further limitation of MaxAE metrics, especially on the urban route.

Discussion When added to the GNSS-PDR constellation, the landmark detection and matching mechanism significantly improves heading reference, which is distinctively visible on the initial convergence of heading. It also helps limit larger GNSS deviations – while PDR disciplines the short- to mid-sized trajectory features, landmarks bring another geo-referenced data source which cover the long term.

Strongly depending on the feature map, regarding which the discussion above has commented on lower yield of the front and back cameras, side cameras proved more useful again. This aspect is perceived as the major limitation of the described mechanism – intuitively, the front/back camera pair seems to be more important for keeping the pedestrian's position estimate accurate laterally and thus managing to navigate them within the breadth of a sidewalk. On the other hand, potential users are not expected to discard their white cane, which helps in this context substantially.

Feature matching performance, when constrained to a reasonably near set of landmarks to match to, did not evince any striking number of false matches. Correct match success rate depended on a multitude of factors, including landmark type, sunshine conditions, camera motion blur and others; selecting the map reference and localization pass data sets from different days and day times has nevertheless demonstrated a certain level of robustness. In this context, the upright SURF flavor may be more efficient and less prone to false matches – at the expense of source image rotation prior to feature extraction, for which conventional SURF was preferred.

6.2 Global navigation

6.2.1 Global path planning experiments

Chapter 5 introduced the Open Street Maps as the source of the high-level segment map data used for experimental evaluation. As already noted there, the structure of the map data representation makes it difficult for planning utilization with respect to the public transport integration. Nevertheless, even the data available allowed several sensible experiments to be executed; two of them are described below.

During all mentioned experiments, the following cost coefficients were chosen in agreement with the foundations of section 4.2: $\lambda_{\mathrm{pt}} = 0.1$, $\lambda_{\mathrm{s}} = 0.5$, $\lambda_{\mathrm{st}} = 3$. The intersection

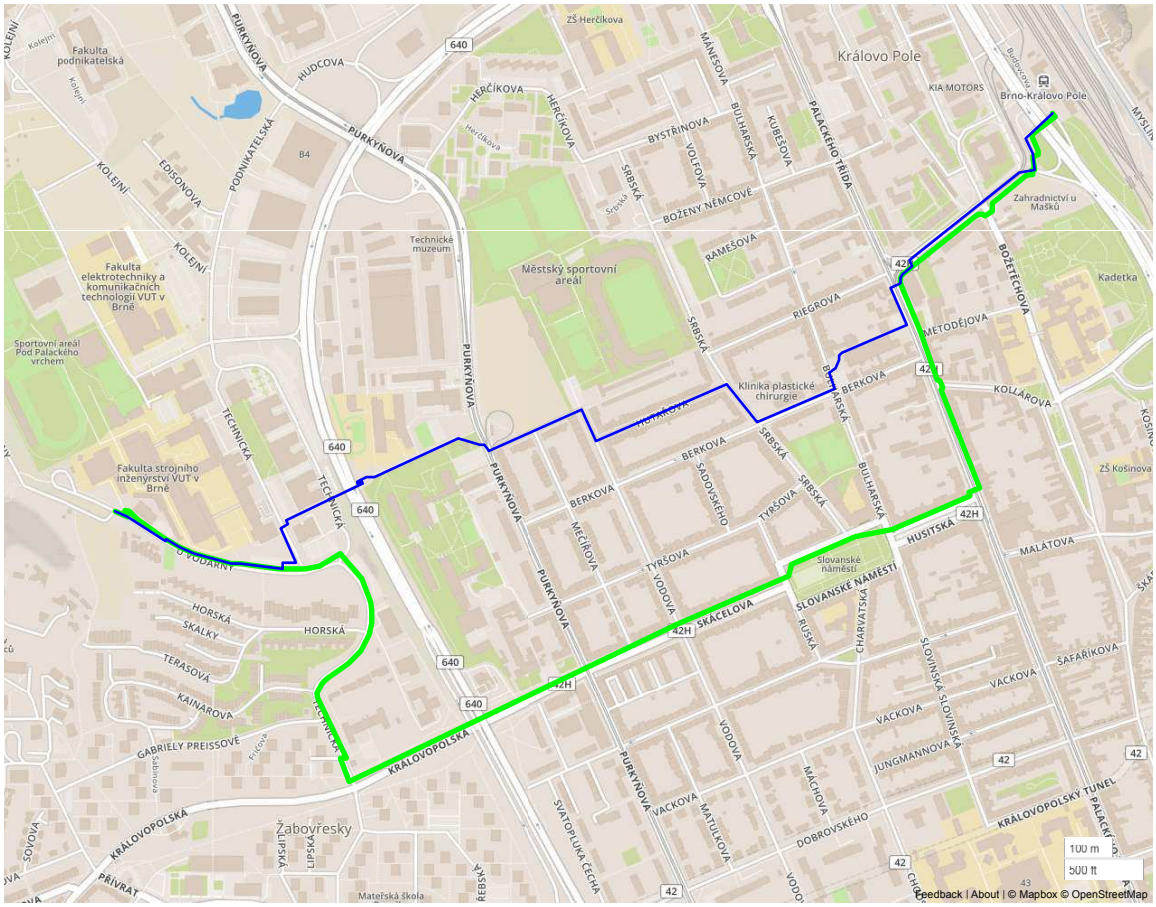


Figure 6.14: Exercise of the public transport preference (green) over sidewalk-only route.

costs λ_{i} were not employed in the proposed form, since there was no reasonable way of determining the lanes count. The heuristic coefficient λ_{h} was experimented with as well; for the below described exercises it was set to $\lambda_{\mathrm{h}} = 0$ though, effectively disabling the heuristic part, to be able to assess solely the cost function optimality.

First, the chosen costs were exercised no a mixed route consisting both of sidewalk and public transport segments. As Fig. 6.14 captures, two variant path plans were compared; the blue one was planned with only pedestrian motion allowed and thus minimizing the distance and risk costs. The green trace was allowed to utilize any blind-friendly motion type available, i.e. primarily public transport with remaining parts of the route covered by walking. It can be observed that the lower cost of the public transport enables selection of the route even though it clearly means a greater distance traversed.

The second scenario in Fig. 6.15 examines this mechanism even more extremely; only bus connections were allowed on the green route to explore the functionality of the public transport preference in case of a greater difference between the viable walk-only and public transport-aided routes. The resulting green trace is indeed more cumbersome that in the previous case, which was expected though due to the special experiment assignment.

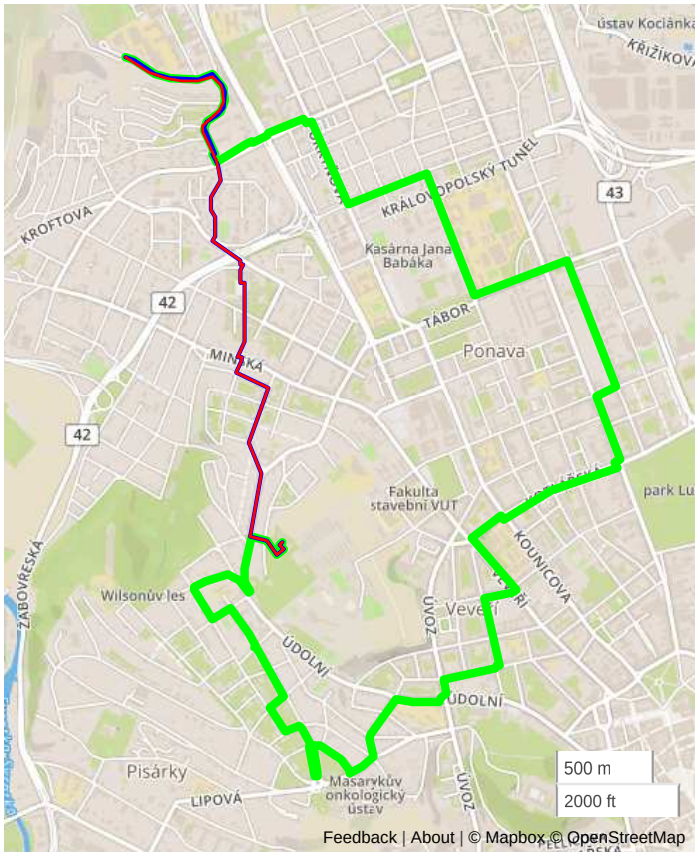


Figure 6.15: Pronounced difference between a bus-only transport (green) and shortest path.

6.2.2 High-level instructions generation evaluation

As presented in the Implementation chapter, the fuzzy instruction generator was implemented within the G1 platform, based on simulation of system events to guarantee repeatability of experiments. Since high-level information understanding is a subjective matter and suitable exact quantification methods were not found, able to evaluate the performance numerically, only human-centered experiments were performed. Moreover, since finding a proper technical solution to instructions generation was solved by this thesis, rather than finding the best set of actual instructions which is a huge topic on its own (refer e.g. to [102, 103]), the evaluation was focused on viability of the design and not on the contents.

Fuzzy logic has been found convenient for notation of text-based conditions and variables – even though some of the messages did not actually benefit from fuzzy activation (simple conditions that could have been implemented by Boolean logic as well), the instruction engine proved to be a very flexible “smart database”. Rather than the fuzzy activation, the deliberately inexact linguistic expressions were found to be the benefit of the solution.

Chapter 7

Conclusions

The thesis gives a thorough description of motivation, approaches and results of research carried out on methods forming the core of a pedestrian information system with focus on visually impaired users. The work was divided into the two fundamental topics of *localization* and *path planning*; this responsibility division is common in the field of autonomous mobile robotics which was introduced as the reference frame for the research.

The main attention was focused on the localization mechanism with respect to the pedestrian application; as discussed, global navigation satellite systems (GNSS) alone do not provide sufficient level of performance to achieve reliable pedestrian localization, especially in urban environments. Two stages of GNSS augmentation were proposed, using data fusion frameworks of the unscented Kalman filter and the particle filter – firstly, pedestrian dead reckoning (PDR) pose change estimates were fused to GNSS measurements to improve local trajectory shape faithfulness. Secondly, since addition of PDR is only effective in short- to mid-term, a natural landmark detection engine was designed as another geo-referenced positioning mechanism complementing and competing with GNSS.

Path planning on relatively limited-sized maps (which a pedestrian can expectedly traverse) is a task routinely solved using the formalism of graph theory and state space search algorithms. The methods, however, ultimately depend on proper definition of optimality criteria – analyzing the specifics of pedestrian path selection with focus on visually impaired users and formulation of appropriate cost functions was then the main objective of the path planning chapter. In an attempt of human-oriented design, a fuzzy logic driven navigation instructions generator was introduced, conveying infrequent high-level information through speech synthesis. Based on experience in mobile robotics, a suitable solution of local path planning was proposed as well; a haptic feedback mechanism was chosen to give real-time direction commands.

Obstacle detection and avoidance as the optional topic within path planning was researched and potential sensing and processing instrumentation was evaluated in terms of estimated usability. The analysis concluded that a full-fledged sensory replacement of the white cane is not yet available, which would be reliably capable of detection of even small ter-

rain irregularities, that may, however, make the visually impaired person stumble. Within the scope of the thesis, the topic was thus suspended.

7.1 Performance

Proposed localization mechanisms were experimentally verified on real-world data in representative environments containing both natural and urban features. The GNSS-PDR fusion alone was shown to be capable of reducing maximal position deviation by 20–40 % and RMSE by 35–45 %, compared to GNSS-only performance on reference routes. Furthermore, addition of the landmark-based geo-referencing improved heading convergence and stability, being able to correct GNSS outliers caused by multi-path effects to large extent in the urban-like environment. During the described urban test route experiment, the employment of the GNSS-PDR-landmark fusion reduced the maximum position deviation from 8.45 m to 2.76 m while evincing MAE of 1.52 m and RMSE of 1.63 m.

Path planning performance was assessed on unmodified map data gained from the crowd-sourced Open Street Map project. Based on the formulated optimality criteria, several cases of path planning were examined focusing on public transport utilization. Since the optimality cannot be, in case of the visually impaired user, evaluated strictly in terms of the shortest or fastest route, the experimental evaluation was driven by soft, human-centric standards. Within possibilities of the available segment map data, the planning was assessed to perform similarly to how a human would choose the path plan.

7.2 Contribution

Theoretical contribution Pedestrian localization belongs to topics less frequently investigated in literature, compared to other localization topics – this thesis helps to better the situation by assembling a range of relevant sources, analyzing potential contributions of various sensory means and formulating applicable processing methods. Particularly, two stages of sensor data fusion were elaborated; the natural landmark detection subsystem brings a novel approach of map building based on spatially located visual landmarks.

An important mechanism enabling seamless fusion of all the multi-rate sensory data sources is the introduced notion of retrospective odometry interpolation. While still bound to the step-wise PDR updates, asynchronously incoming observations from other sensors can be processed by the prediction-correction recursive state estimation iteration without losing valuable precision of the PDR data.

Theoretical contribution of the path planning chapter is twofold: first, optimality criteria and cost function formulation for urban path selection were established, focusing on a visually impaired pedestrian with utilization of public transport. Second, the employment of fuzzy logic was conceived to give the navigation instructions a human-friendly envelope.

Practical contribution As the most widely applicable practical outcome, the thesis provides a thorough benchmark of the GNSS-PDR and GNSS-PDR-landmark fusion mechanisms in realistic outdoor environments; standard error metrics were used to allow comparison of the achieved performance to other approaches.

The wearable platform has been established in two generations, exploring ways to accomplish modularity and extendability by design. In the second generation, custom ROS drivers were implemented for the Osmium MIMU22BL inertial measurement unit serving in the role of the PDR sensor.

Last but not least, the localization framework was implemented based on both parametric and non-parametric filters with advanced visualization mechanisms and positioning error measures computation. This has allowed running comparison studies evaluating behavior of various data fusion schemes and quickly experimenting with the control parameters.

7.3 Future work

Albeit having performed well in evaluation exercises, each subsystem offers certain room for improvements. For one, landmark detection in localization phase could be further evolved by adopting a model-driven matching engine – then, even the lower-quality landmarks could be mapped without leading to excessive false detection rates. Data mining from the Open Street Maps may be an apt candidate for further investigation as well, mainly focusing on the public transport routes.

The presented improvements of worst-case positioning error are encouraging enough to think about next phase of the project – an evaluation of engineering efforts that would be necessary to transform the wearable platform into a user-friendly solution. Realistically, non-trivial engineering challenges are to be expected though, due to the extensive sensory equipment and essential ease of use requirements.

Bibliography

1. WORLD HEALTH ORGANIZATION. *Vision impairment and blindness (fact sheet)* [online]. 2017 [visited on 2018-02-10]. Available from: http://www.who.int/mediacentre/factsheets/fs282/en/.
2. WONG, S. Traveling with blindness: A qualitative space-time approach to understanding visual impairment and urban mobility. *Health & Place*. 2018-01, vol. 49, pp. 85–92. Available from DOI: 10.1016/j.healthplace.2017.11.009.
3. MARSTON, J. R.; GOLLEDGE, R. G.; COSTANZO, C. M. Investigating Travel Behavior of Nondriving Blind and Vision Impaired People: The Role of Public Transit. *The Professional Geographer*. 1997-05, vol. 49, no. 2, pp. 235–245. Available from DOI: 10.1111/0033-0124.00073.
4. WORTH, N. Visual Impairment in the City: Young People's Social Strategies for Independent Mobility. *Urban Studies*. 2013-01, vol. 50, no. 3, pp. 574–586. Available from DOI: 10.1177/0042098012468898.
5. JENKINS, S. *Shared space is the future for London's roads* [online]. 2012-01-31 [visited on 2018-03-04]. Available from: https://www.standard.co.uk/comment/comment/shared-space-is-the-future-for-londons-roads-7313484.html.
6. IMRIE, R. Shared Space and the Post-politics of Environmental Change. *Urban Studies*. 2013-04, vol. 50, no. 16, pp. 3446–3462. Available from DOI: 10.1177/0042098013482501.
7. THE GUIDE DOGS FOR THE BLIND ASSOCIATION. *Shared Surfaces* [online]. 2018 [visited on 2018-03-04]. Available from: https://www.guidedogs.org.uk/how-you-can-help/campaigning/shared-surfaces/.
8. SVÁROVSKÝ, S.R.O. *Bílé hole* [online]. 2018 [visited on 2018-02-10]. Available from: http://www.svarovsky.cz/o-bilych-holich/.
9. DEPARTMENT FOR TRANSPORT, HER MAJESTY'S GOVERNMENT. *Using tactile paving surfaces* [online]. 2007-06-05 [visited on 2018-03-10]. Available from: https://www.gov.uk/government/publications/guidance-on-the-use-of-tactile-paving-surfaces.
10. INGOLFSON. *Sylvia Park New Train Station II* [online]. 2007 [visited on 2018-03-10]. Available from: https://commons.wikimedia.org/wiki/File:Sylvia_Park_New_Train_Station_II.jpg.
11. THE GUIDE DOGS FOR THE BLIND ASSOCIATION. *What a guide dog does* [online]. 2018 [visited on 2018-03-10]. Available from: https://www.guidedogs.org.uk/services-we-provide/guide-dogs/what-a-guide-dog-does/.

12. RETSCHER, G.; FU, Q. Integration of RFID, GNSS and DR for Ubiquitous Positioning in Pedestrian Navigation. *Journal of Global Positioning Systems*. 2007-06, vol. 6, no. 1, pp. 56–64. Available from DOI: 10.5081/jgps.6.1.56.

13. BARANSKI, P.; STRUMILLO, P. Enhancing Positioning Accuracy in Urban Terrain by Fusing Data from a GPS Receiver, Inertial Sensors, Stereo-Camera and Digital Maps for Pedestrian Navigation. *Sensors*. 2012-05, vol. 12, no. 6, pp. 6764–6801. Available from DOI: 10.3390/s120606764.

14. TREUILLET, S.; ROYER, E. Outdoor/Indoor Vision Based Localization for Blind Pedestrian Navigation Assistance. *International Journal of Image and Graphics*. 2010, vol. 10, no. 04, pp. 481–496. Available from DOI: 10.1142/s0219467810003937.

15. ALAOUI, F. T.; BETAILLE, D.; RENAUDIN, V. Pedestrian Dead Reckoning Navigation with the Help of A∗-Based Routing Graphs in Large Unconstrained Spaces. *Wireless Communications and Mobile Computing*. 2017, vol. 2017, pp. 1–10. Available from DOI: 10.1155/2017/7951346.

16. THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, 2005. ISBN 978-0262201629.

17. SIEGWART, R.; NOURBAKHSH, I. R. *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents series)*. The MIT Press, 2004. ISBN 978-0262195027.

18. CHOSET, H.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, G. A.; BURGARD, W.; KAVRAKI, L. E.; THRUN, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations (Intelligent Robotics and Autonomous Agents series)*. A Bradford Book, 2005. ISBN 0-262-03327-5.

19. DIEBEL, J. *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*. 2006.

20. MOAKHER, M. Means and Averaging in the Group of Rotations. *SIAM Journal on Matrix Analysis and Applications*. 2002-01, vol. 24, no. 1, pp. 1–16. Available from DOI: 10.1137/s0895479801383877.

21. MAO, J. Optimal orthonormalization of the strapdown matrix by using singular value decomposition. *Computers & Mathematics with Applications*. 1986-03, vol. 12, no. 3, pp. 353–362. Available from DOI: 10.1016/0898-1221(86)90194-x.

22. MARKLEY, F. L.; CHENG, Y.; CRASSIDIS, J. L.; OSHMAN, Y. Averaging Quaternions. *Journal of Guidance, Control, and Dynamics*. 2007-07, vol. 30, no. 4, pp. 1193–1197. Available from DOI: 10.2514/1.28949.

23. MADGWICK, S. O. *Quaternions* [online]. 2011-09-24 [visited on 2018-03-25]. Available from: http://x-io.co.uk/res/doc/quaternions.pdf.

24. EUROPEAN SPACE AGENCY. *Navipedia: The reference for Global Navigation Satellite Systems* [online]. 2018 [visited on 2018-04-08]. Available from: http://www.navipedia.net/index.php/Main_Page.

25. MILLIKEN, R. J.; ZOLLER, C. J. Principle of Operation of NAVSTAR and System Characteristics. *Navigation*. 1978-06, vol. 25, no. 2, pp. 95–106. Available from DOI: 10.1002/j.2161-4296.1978.tb01322.x.

26. JANUSZEWSKI, J. Sources of Error in Satellite Navigation Positioning. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*. 2017, vol. 11, no. 3, pp. 419–423. Available from DOI: 10.12716/1001.11.03.04.

27. KLOBUCHAR, J. Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users. *IEEE Transactions on Aerospace and Electronic Systems*. 1987-05, vol. AES-23, no. 3, pp. 325–331. Available from DOI: 10.1109/taes.1987.310829.

28. COMMISSION, E. *Ionospheric correction Algorithm for Galileo Single Frequency Users* [Issue 1.2] [online]. 2016 [visited on 2018-04-08]. Available from: https://www.gsc-europa.eu/system/files/galileo_documents/Galileo_Ionospheric_Model.pdf.

29. SUBIRANA, J. S.; ZORNOZA, J. J.; HERNÁNDEZ-PAJARES, M. *Tropospheric Delay: Navipedia* [online]. 2011 [visited on 2018-03-31]. Available from: http://www.navipedia.net/index.php/Tropospheric_Delay.

30. LANGLEY, R. B. Dilution of Precision. *GPS World*. 1999, vol. 10, no. 5, pp. 52–59. ISSN 1048-5104.

31. GOMEZ-GIL, J.; RUIZ-GONZALEZ, R.; ALONSO-GARCIA, S.; GOMEZ-GIL, F. A Kalman Filter Implementation for Precision Improvement in Low-Cost GPS Positioning of Tractors. *Sensors*. 2013-11, vol. 13, no. 11, pp. 15307–15323. Available from DOI: 10.3390/s131115307.

32. ALI, K.; CHEN, X.; DOVIS, F.; CASTRO, D. D.; FERNANDEZ, A. J. GNSS signal multipath error characterization in urban environments using LiDAR data aiding. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. IEEE, 2012-10. Available from DOI: 10.1109/estel.2012.6400123.

33. TAKEUCHI, E.; YAMAZAKI, M.; OHNO, K.; TADOKORO, S. GPS measurement model with satellite visibility using 3D map for particle filter. In: *2011 IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011. Available from DOI: 10.1109/robio.2011.6181350.

34. ILAND, D.; IRISH, A.; MADHOW, U.; SANDLER, B. *Rethinking GPS: Engineering Next-Gen Location at Uber* [online]. 2018-04-19 [visited on 2018-07-15]. Available from: https://eng.uber.com/rethinking-gps/.

35. BEZICK, S. M.; PUE, A. J.; PATZELT, C. M. Inertial Navigation for Guided Missile Systems. *Johns Hopkins APL Technical Digest*. 2010, vol. 28, no. 4, pp. 331–342. ISSN 0270-5214.

36. XSENS NORTH AMERICA INC. *MTi 1-series* [online]. 2018 [visited on 2018-07-20]. Available from: https://www.xsens.com/products/mti-1-series/.

37. KONVALIN, C. *Compensating for Tilt, Hard-Iron, and Soft-Iron Effects* [online]. 2009-12-01 [visited on 2018-06-18]. Available from: https://www.sensorsmag.com/components/compensating-for-tilt-hard-iron-and-soft-iron-effects.

38. OLLIW. *IMU Data Fusing: Complementary, Kalman, and Mahony Filter* [online]. 2015-01-16 [visited on 2018-03-25]. Available from: http://www.olliw.eu/2013/imu-data-fusing/.

39. YUN, X.; BACHMANN, E. R.; MOORE, H.; CALUSDIAN, J. Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007. Available from DOI: 10.1109/robot.2007.363845.

40. JIMENEZ, A.; SECO, F.; PRIETO, C.; GUEVARA, J. A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In: *2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 2009-08. Available from DOI: 10.1109/wisp.2009.5286542.

41. NILSSON, J.-O.; GUPTA, A. K.; HANDEL, P. Foot-mounted inertial navigation made easy. In: *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014-10. Available from DOI: 10.1109/ipin.2014.7275464.

42. GOYAL, P.; RIBEIRO, V. J.; SARAN, H.; KUMAR, A. Strap-down Pedestrian Dead-Reckoning system. In: *2011 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2011. Available from DOI: 10.1109/ipin.2011.6071935.

43. ALVAREZ, J. C.; ALVAREZ, D.; LÓPEZ, A.; GONZÁLEZ, R. C. Pedestrian Navigation Based on a Waist-Worn Inertial Sensor. *Sensors*. 2012, vol. 12, no. 8, pp. 10536–10549. Available from DOI: 10.3390/s120810536.

44. DO, T.-N.; LIU, R.; YUEN, C.; ZHANG, M.; TAN, U.-X. Personal Dead Reckoning Using IMU Mounted on Upper Torso and Inverted Pendulum Model. *IEEE Sensors Journal*. 2016-11, vol. 16, no. 21, pp. 7600–7608. Available from DOI: 10.1109/jsen.2016.2601937.

45. NILSSON, J.-O.; ZACHARIAH, D.; SKOG, I.; HÄNDEL, P. Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging. *EURASIP Journal on Advances in Signal Processing*. 2013, vol. 2013, no. 1. Available from DOI: 10.1186/1687-6180-2013-164.

46. ILYAS, M.; CHO, K.; BAEG, S.-H.; PARK, S. Drift Reduction in Pedestrian Navigation System by Exploiting Motion Constraints and Magnetic Field. *Sensors*. 2016, vol. 16, no. 9, pp. 1455. Available from DOI: 10.3390/s16091455.

47. ROMANOVAS, M.; GORIDKO, V.; AL-JAWAD, A.; SCHWAAB, M.; TRAECHTLER, M.; KLINGBEIL, L.; MANOLI, Y. A study on indoor pedestrian localization algorithms with foot-mounted sensors. In: *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2012. Available from DOI: 10.1109/ipin.2012.6418886.

48. OLSON, E. AprilTag: A robust and flexible visual fiducial system. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011-05. Available from DOI: 10.1109/icra.2011.5979561.

49. VALGREN, C.; LILIENTHAL, A. J. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*. 2010, vol. 58, no. 2, pp. 149–156. Available from DOI: 10.1016/j.robot.2009.09.010.

50. KALIA, R.; LEE, K.-D.; SAMIR, B.; JE, S.-K.; OH, W.-G. An analysis of the effect of different image preprocessing techniques on the performance of SURF: Speeded Up Robust Features. In: *2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*. IEEE, 2011. Available from DOI: 10.1109/fcv.2011.5739756.

51. ROSTEN, E.; DRUMMOND, T. Machine Learning for High-Speed Corner Detection. In: *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, 2006, pp. 430–443. Available from DOI: 10.1007/11744023_34.

52. LOWE, D. G. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. IEEE, 1999. Available from DOI: 10.1109/iccv.1999.790410.

53. LOWE, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 2004, vol. 60, no. 2, pp. 91–110. Available from DOI: 10.1023/b:visi.0000029664.99615.94.

54. BAY, H.; TUYTELAARS, T.; GOOL, L. V. SURF: Speeded Up Robust Features. In: *Computer Vision – ECCV 2006*. Springer Berlin Heidelberg, 2006, pp. 404–417. Available from DOI: 10.1007/11744023_32.

55. MURILLO, A. C.; GUERRERO, J. J.; SAGUES, C. SURF features for efficient robot localization with omnidirectional images. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007. Available from DOI: 10.1109/robot.2007.364077.

56. HERTZBERG, C.; WAGNER, R.; BIRBACH, O.; HAMMER, T.; FRESE, U. Experiences in building a visual SLAM system from open source components. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011. Available from DOI: 10.1109/icra.2011.5980140.

57. LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. BRISK: Binary Robust invariant scalable keypoints. In: *2011 International Conference on Computer Vision*. IEEE, 2011. Available from DOI: 10.1109/iccv.2011.6126542.

58. MIKOLAJCZYK, K.; SCHMID, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005, vol. 27, no. 10, pp. 1615–1630. Available from DOI: 10.1109/tpami.2005.188.

59. SE, S.; LOWE, D. G.; LITTLE, J. J. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*. 2005, vol. 21, no. 3, pp. 364–375. Available from DOI: 10.1109/tro.2004.839228.

60. HE, X.; ZEMEL, R. S.; MNIH, V. Topological map learning from outdoor image sequences. *Journal of Field Robotics*. 2006, vol. 23, no. 11-12, pp. 1091–1104. Available from DOI: 10.1002/rob.20170.

61. HILE, H.; GRZESZCZUK, R.; LIU, A.; VEDANTHAM, R.; KOŠECKA, J.; BORRIELLO, G. Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2009, pp. 59–76. Available from DOI: 10.1007/978-3-642-01516-8_6.

62. BETKE, M.; GURVITS, L. Mobile robot localization using landmarks. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*. IEEE, 1994. Available from DOI: 10.1109/iros.1994.407399.

63. FONT, J. M.; BATLLE, J. A. Mobile Robot Localization. Revisiting the Triangulation Methods. *IFAC Proceedings Volumes*. 2006, vol. 39, no. 15, pp. 340–345. Available from DOI: 10.3182/20060906-3-it-2910.00058.

64. PIERLOT, V.; DROOGENBROECK, M. V. A New Three Object Triangulation Algorithm for Mobile Robot Positioning. *IEEE Transactions on Robotics*. 2014, vol. 30, no. 3, pp. 566–577. Available from DOI: 10.1109/tro.2013.2294061.

65. AFTAB, K.; HARTLEY, R.; TRUMPF, J. Lq-Closest-Point to Affine Subspaces Using the Generalized Weiszfeld Algorithm. *International Journal of Computer Vision*. 2015, vol. 114, no. 1, pp. 1–15. Available from DOI: 10.1007/s11263-014-0791-8.

66. KELLY, A. Precision dilution in triangulation based mobile robot position estimation. *Intelligent Autonomous Systems*. 2003, vol. 8, pp. 1046–1053.

67. KATO, H.; BILLINGHURST, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE Comput. Soc, 1999. Available from DOI: 10.1109/iwar.1999.803809.

68. FIALA, M. ARTag, a Fiducial Marker System Using Digital Techniques. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005. Available from DOI: 10.1109/cvpr.2005.74.

69. WANG, J.; OLSON, E. AprilTag 2: Efficient and robust fiducial detection. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016. Available from DOI: 10.1109/iros.2016.7759617.

70. ELMENREICH, W. *An Introduction to Sensor Fusion*. Treitlstr. 1-3/182-1, 1040 Vienna, Austria, 2001. Research Report. Technische Universität Wien, Institut für Technische Informatik.

71. POLYA, G. Über den zentralen Grenzwertsatz der Wahrscheinlichkeitsrechnung und das Momentenproblem. *Mathematische Zeitschrift*. 1920, vol. 8, no. 3-4, pp. 171–181. Available from DOI: 10.1007/bf01206525.

72. SLOTINE, J.-J. E.; LI, W. *Applied Nonlinear Control*. Prentice-Hall, Inc., 1991. ISBN 0-13-040890-5.

73. KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*. 1960, no. 82, pp. 35–45.

74. KREJSA, J.; VECHET, S. Infrared Beacons based Localization of Mobile Robot. *Electronics and Electrical Engineering*. 2012, vol. 117, no. 1. Available from DOI: 10.5755/j01.eee.117.1.1046.

75. SADEGHI, B.; MOSHIRI, B. Second-order EKF and Unscented Kalman Filter Fusion for Tracking Maneuvering Targets. In: *2007 IEEE International Conference on Information Reuse and Integration*. IEEE, 2007. Available from DOI: 10.1109/iri.2007.4296672.

76. PEREA, L.; HOW, J.; BREGER, L.; ELOSEGUI, P. Nonlinearity in Sensor Fusion: Divergence Issues in EKF, Modified Truncated GSF, and UKF. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2007. Available from DOI: 10.2514/6.2007-6514.

77. JULIER, S. J.; UHLMANN, J. K.; DURRANT-WHYTE, H. F. A new approach for filtering nonlinear systems. In: *Proceedings of 1995 American Control Conference - ACC'95*. American Autom Control Council, 1995. Available from DOI: 10.1109/acc.1995.529783.

78. WAN, E. A.; MERWE, R. van der. The unscented Kalman filter for nonlinear estimation. In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. IEEE, 2000. Available from DOI: 10.1109/asspcc.2000.882463.

79. JULIER, S. J.; UHLMANN, J. K. Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*. 2004, vol. 92, no. 3, pp. 401–422. Available from DOI: 10.1109/jproc.2003.823141.

80. WU, Y.; HU, D.; WU, M.; HU, X. Unscented Kalman filtering for additive noise case: augmented versus nonaugmented. *IEEE Signal Processing Letters*. 2005-05, vol. 12, no. 5, pp. 357–360. Available from DOI: 10.1109/lsp.2005.845592.

81. LIU, J. S.; CHEN, R. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*. 1998-09, vol. 93, no. 443, pp. 1032–1044. Available from DOI: 10.1080/01621459.1998.10473765.

82. ZHOU, J.; KNEDLIK, S.; LOFFELD, O. INS/GPS Tightly-coupled Integration using Adaptive Unscented Particle Filter. *Journal of Navigation*. 2010, vol. 63, no. 03, pp. 491–511. Available from DOI: 10.1017/s0373463310000068.

83. ZHOU, J.; EDWAN, E.; KNEDLIK, S.; LOFFELD, O. Low-cost INS/GPS with nonlinear filtering methods. In: *2010 13th International Conference on Information Fusion*. IEEE, 2010. Available from DOI: 10.1109/icif.2010.5712023.

84. TAWK, Y.; TOMÉ, P.; BOTTERON, C.; STEBLER, Y.; FARINE, P.-A. Implementation and Performance of a GPS/INS Tightly Coupled Assisted PLL Architecture Using MEMS Inertial Sensors. *Sensors*. 2014, vol. 14, no. 2, pp. 3768–3796. Available from DOI: 10.3390/s140203768.

85. DROLET, L.; MICHAUD, F.; COTE, J. Adaptable sensor fusion using multiple Kalman filters. In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*. IEEE, 2000. Available from DOI: 10.1109/iros.2000.893222.

86. SON, Y. S.; KIM, W.; LEE, S.-H.; CHUNG, C. C. Asynchronous Sensor Fusion using Multi-rate Kalman Filter. *The Transactions of The Korean Institute of Electrical Engineers*. 2014, vol. 63, no. 11, pp. 1551–1558. ISSN 2287-4364. Available from DOI: 10.5370/KIEE.2014.63.11.1551.

87. ZHOU, Y. A Kalman filter based registration approach for asynchronous sensors in multiple sensor fusion applications. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004. Available from DOI: 10.1109/icassp.2004.1326252.

88. WANG, Y.; KOSTIC, D.; JANSEN, S. T. H.; NIJMEIJER, H. Filling the gap between low frequency measurements with their estimates. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014. Available from DOI: 10.1109/icra.2014.6906606.

89. ZHANG, P.; GU, J.; MILIOS, E. E.; HUYNH, P. Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter. In: *IEEE International Conference Mechatronics and Automation, 2005*. IEEE, 2005. Available from DOI: 10.1109/icma.2005.1626777.

90. BÖRLIN, N. *Nonlinear Optimization: Least Squares Problems – The Gauss-Newton method* [online]. 2007-11-22 [visited on 2018-07-15]. Available from: `https://www8.cs.umu.se/kurser/5DA001/HT07/lectures/lsq-handouts.pdf`.

91. HART, P.; NILSSON, N.; RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*. 1968, vol. 4, no. 2, pp. 100–107. Available from DOI: `10.1109/tssc.1968.300136`.

92. DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959, vol. 1, no. 1, pp. 269–271. Available from DOI: `10.1007/bf01386390`.

93. ZADEH, L. A. Soft computing and fuzzy logic. *IEEE Software*. 1994-11, vol. 11, no. 6, pp. 48–56. Available from DOI: `10.1109/52.329401`.

94. ZADEH, L. A. Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*. 1994, vol. 37, no. 3, pp. 77–84. Available from DOI: `10.1145/175247.175255`.

95. NILSSON, N. J. Probabilistic logic. *Artificial Intelligence*. 1986, vol. 28, no. 1, pp. 71–87. Available from DOI: `10.1016/0004-3702(86)90031-7`.

96. DEMPSTER, A. P. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Mathematical Statistics*. 1967, vol. 38, no. 2, pp. 325–339. Available from DOI: `10.1214/aoms/1177698950`.

97. SHAFER, G. *A Mathematical Theory of Evidence*. Princeton University Press, 1976-04-11. ISBN 069110042X.

98. ZADEH, L. A. Fuzzy sets. *Information and Control*. 1965-06, vol. 8, no. 3, pp. 338–353. Available from DOI: `10.1016/s0019-9958(65)90241-x`.

99. KLIR, G. J.; YUAN, B. *Fuzzy Sets & Fuzzy Logic*. Addison Wesley Pub Co Inc, 1995-05-11. ISBN 0131011715.

100. TERANO, T.; SUGENO, M.; MUKAIDO, M.; SHIGEMASU, K. *Fuzzy Engineering toward Human Friendly Systems*. 1992. ISBN 978-90-5199-082-9.

101. PEDRYCZ, W.; GOMIDE, F. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. John Wiley & Sons Inc., 2007-08-11. ISBN 978-0-471-78857-7.

102. GAUNET, F.; BRIFFAULT, X. Exploring the Functional Specifications of a Localized Wayfinding Verbal Aid for Blind Pedestrians: Simple and Structured Urban Areas. *Human-Computer Interaction*. 2005, vol. 20, no. 3, pp. 267–314. Available from DOI: `10.1207/s15327051hci2003_2`.

103. GAUNET, F. Verbal guidance rules for a localized wayfinding aid intended for blind-pedestrians in urban areas. *Universal Access in the Information Society*. 2006, vol. 4, no. 4, pp. 338–353. Available from DOI: `10.1007/s10209-003-0086-2`.

104. VĚCHET, S.; KREJSA, J. Concurrent mapping and localization based on potential fields. In: FUIS, V. (ed.). *Engineering Mechanics 2011*. Prague: Institute of Thermomechanics AS CR, 2011, pp. 647–650. ISBN 978-80-87012-33-8.

105. MARSTON, J.; LOOMIS, J. M.; KLATZKY, R.; GOLLEDGE, R. G. Nonvisual Route Following with Guidance from a Simple Haptic or Auditory Display. *Journal of Visual Impairment & Blindness*. 2007-04, vol. 101, no. 4, pp. 203–211.

106. GUSTAFSON-PEARCE, O.; BILLETT, E.; CECELJA, F. Comparison between audio and tactile systems for delivering simple navigational information to visually impaired pedestrians. *British Journal of Visual Impairment*. 2007, vol. 25, no. 3, pp. 255–265. Available from DOI: 10.1177/0264619607079807.

107. PIELOT, M.; POPPINGA, B.; BOLL, S. PocketNavigator. In: *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services - MobileHCI '10*. ACM Press, 2010. Available from DOI: 10.1145/1851600.1851696.

108. RÜMELIN, S.; RUKZIO, E.; HARDY, R. NaviRadar: A Novel Tactile Information Display for Pedestrian Navigation. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST'11*. ACM Press, 2011. Available from DOI: 10.1145/2047196.2047234.

109. IRO, A.; KONSTANTINOS, C. Pedestrian navigation and shortest path: Preference versus distance. *Ambient Intelligence and Smart Environments*. 2013, vol. 17, no. Workshop Proceedings of the 9th International Conference on Intelligent Environments, pp. 647–652. ISSN 1875-4163. Available from DOI: 10.3233/978-1-61499-286-8-647.

110. NOVÁK, V. *Základy fuzzy modelování*. Praha: BEN - technická literatura, 2003. ISBN 80-7300-009-1.

111. BONARINI, A.; INVERNIZZI, G.; LABELLA, T. H.; MATTEUCCI, M. An architecture to coordinate fuzzy behaviors to control an autonomous robot. *Fuzzy Sets and Systems*. 2003, vol. 134, no. 1, pp. 101–115. Available from DOI: 10.1016/s0165-0114(02)00232-4.

112. ZADEH, L. A. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*. 1975-01, vol. 8, no. 3, pp. 199–249. Available from DOI: 10.1016/0020-0255(75)90036-5.

113. BANKS, W. *Linguistic Variables: Clear Thinking with Fuzzy Logic* [online]. 2014-04-24 [visited on 2018-08-11]. Available from: https://www.phaedsys.com/principals/bytecraft/bytecraftdata/LinguisticVariables.pdf.

114. ROBOTICS, F. *PC/104+ nDepth Vision System* [online]. 2018 [visited on 2018-07-18]. Available from: http://www.focusrobotics.com/products/systems.html.

115. HUANG, A. S.; OLSON, E.; MOORE, D. C. LCM: Lightweight Communications and Marshalling. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010. Available from DOI: 10.1109/iros.2010.5649358.

116. NILSSON, J.-O.; SKOG, I. *OpenShoe: Foot-mounted INS for Every Foot* [online]. 2016 [visited on 2018-07-20]. Available from: http://www.openshoe.org/.

117. GT SILICON PVT LTD. *Osmium MIMU22BL / 22BLP / 22BLPX – A Shoe-based PDR Sensor (IMU)* [online]. 2016 [visited on 2018-07-20]. Available from: https://www.inertialelements.com/osmium-mimu22bl.html.

118. SKOG, I.; NILSSON, J.-O.; HANDEL, P. An open-source multi inertial measurement unit (MIMU) platform. In: *2014 International Symposium on Inertial Sensors and Systems (ISISS)*. IEEE, 2014. Available from DOI: 10.1109/isiss.2014.6782523.

List of author's publications

KREJSA, J.; VECHET, S.; HRBACEK, J.; SCHREIBER, P. High Level Software Architecture for Autonomous Mobile Robot. In: *Recent Advances in Mechatronics*. Springer Berlin Heidelberg, 2010, pp. 185–190. ISBN 978-3-642-05021-3. Available from DOI: 10.1007/978-3-642-05022-0_32.

HRBACEK, J.; RIPEL, T.; KREJSA, J. Ackermann mobile robot chassis with independent rear wheel drives. In: *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*. Skopje: IEEE, 2010-09, T5–46–T5–51. ISBN 978-1-4244-7856- 9. Available from DOI: 10.1109/epepemc.2010.5606853.

RIPEL, T.; HRBACEK, J.; KREJSA, J. Ackerman Steering chassis with independently driven back wheels. In: *Proceedings of the 1st international conference Robotics in Education, RiE 2010*. Bratislava: Slovak University of Technology, 2010, pp. 127–132. ISBN 978-80-227-3353- 3.

RIPEL, T.; HRBACEK, J. Complex Mechanical Design of Autonomous Robot Advee. In: *Mechatronics 2011*. Berlin: Springer Berlin Heidelberg, 2011, pp. 121–129. ISBN 978-3-642-23243-5. Available from DOI: 10.1007/978-3-642-23244-2_15.

HRBACEK, J.; HRBACEK, R.; VECHET, S. Modular Control System Architecture for a Mobile Robot. In: *Proceedings of the 17th international conference Engineering Mechanics 2011*. Prague: Institute of Thermomechanics, Academy of Sciences of the Czech Republic, 2011, pp. 211–214. ISBN 978-80-87012-33-8.

RIPEL, T.; HRBACEK, J.; KREJSA, J. Design of the Frame for Autonomous Mobile Robot with Ackerman Platform. In: *Proceedings of the 17th international conference Engineering Mechanics 2011*. Prague: Institute of Thermomechanics, Academy of Sciences of the Czech Republic, 2011, pp. 525–518. ISBN 978-80-87012-33-8.

KREJSA, J.; VECHET, S.; HRBACEK, J.; RIPEL, T.; ONDROUSEK, V.; HRBACEK, R.; SCHREIBER, P. Presentation robot Advee. *Engineering Mechanics*. 2012, vol. 18, no. 5/6, pp. 307–322. ISSN 1802-1484.

VECHET, S.; KREJSA, J.; HRBACEK, J. Artificial ant colony method for state-space exploration. In: *Engineering Mechanics 2012, Book of extended abstracts*. Prague: Institute of Theoretical and Applied Mechanics, 2012, pp. 366–367. ISBN 978-80-86246-40-6.

TOMAN, J.; HRBACEK, J.; SINGULE, V. Design of universal control unit for brushless DC motors. In: *Engineering Mechanics 2012, Book of extended abstracts*. Prague: Institute of Theoretical and Applied Mechanics, 2012, pp. 342–343. ISBN 978-80-86246-40-6.

RIPEL, T.; KREJSA, J.; HRBACEK, J. Patient Activity Measurement in Active Elbow Orthosis. In: *Mechatronics 2013*. Berlin: Springer International Publishing, 2014, pp. 817–824. Available from DOI: 10.1007/978-3-319-02294-9_103.

SINGULE, V.; HRBACEK, J. Vývoj testovacího standu pro BLDC motory. In: *Sborník příspěvků Mezinárodní konference učitelů elektrotechniky SEKEL 2014*. Prague: ZČU Praha, 2014, pp. 24–30. ISBN 978-80-213-2480-0.

RIPEL, T.; KREJSA, J.; HRBACEK, J.; CIZMAR, I. Active Elbow Orthosis. *International Journal of Advanced Robotic Systems*. 2014-01, vol. 11, no. 9, pp. 1–10. ISSN 1729-8806. Available from DOI: 10.5772/58874.

HRBACEK, J.; SINGULE, V.; HOUSKA, P. Design of PMSM-based electric motor test stand. In: *Proceedings of the 16th International Conference on Mechatronics - Mechatronika 2014*. Brno: IEEE, 2014-12, pp. 630–634. ISBN 978-80-214-4817-9. Available from DOI: 10.1109/mechatronika.2014.7018327.

HRBACEK, J.; SINGULE, V. Generating Navigational Audio Instructions Using Fuzzy Logic. In: *Solid State Phenomena Vols. 220-221 (2015): Mechatronic Systems and Materials VI*. Solid State Phenomena. Switzerland: Trans Tech Publications, 2014, pp. 435–438. ISBN 9783038352273.

HRBACEK, J.; SINGULE, V.; HOUSKA, P. Mechatronic design of four-quadrant electric drive test stand. In: *2015 International Conference on Electrical Drives and Power Electronics (EDPE)*. Kosice: IEEE, 2015-09, pp. 353–358. ISBN 978-1-4673-9661-5. Available from DOI: 10.1109/edpe.2015.7325319.

VECHET, S.; HRBACEK, J.; KREJSA, J. Environmental Data Analysis for Learning Behavioral Patterns in Smart Homes. In: *Proceedings of the, 2016 17th International Conference on Mechatronics – Mechatronika (ME) 2016*. Prague: Czech Technical University in Prague, 2016, pp. 386–391. ISBN 978-80-01-05882-4.

VECHET, S.; HRBACEK, J.; RUZICKA, M.; MASEK, P.; KREJSA, J. Behavioral Patterns Identification in Smart-Home Application. In: *Mendel 2016 - 22nd International Conference on Soft Computing*. Brno: Brno University of Technology, 2016, pp. 149–152. ISBN 978-80-214-5365-4.

VECHET, S.; KREJSA, J.; HRBACEK, J.; CHEN, K.-S. Probabilistic Reasoning in Diagnostic Expert System for Smart Homes. In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2017-08, pp. 708–715. Available from DOI: 10.1007/978-3-319-65960-2_87.

HRBACEK, J.; VECHET, S.; SINGULE, V. Improving Local Pose Estimates by Fusion of Pedestrian Dead Reckoning to Global Positioning Measurements. *IEEE Sensors Letters*. 2018. ISSN 2475-1472. Manuscript submitted for publication.

Appendix A

List of Abbreviations

AHRS	Attitude and Heading Reference System
BRISK	Binary Robust Invariant Scalable Keypoints
DOF	Degree of Freedom, or Depth of Field
DoH	Determinant of Hessian
DoG	Difference of Gaussian
EKF	Extended Kalman Filter
ESA	European Space Agency
FOSS	Free and Open-Source Software
FREAK	Fast Retina Keypoints
FoV	Field of View
GLOH	Gradient Location and Orientation Histogram
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HOG	Histogram of Oriented Gradients
IMU	Inertial Measurement Unit
KF	Kalman Filter
LoG	Laplacian of Gaussian
MEMS	Micro Electro-Mechanical System
MIMU	Multi Inertial Measurement Unit
PCB	Printed Circuit Board
PCBA	Printed Circuit Board Assembly
PDF	Probability Density Function
PDR	Pedestrian Dead Reckoning
PF	Particle Filter
RANSAC	Random Sample Consensus
ROS	Robot Operating System
SBAS	Satellite-based Augmentation System
SBC	Single-Board Computer
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SURF	Speeded Up Robust Features
VI	Visual Impairment
UKF	Unscented Kalman Filter
ZUPT	Zero-Velocity Update

List of Figures

List of Tables

List of Algorithms