Visual Odometry for Pedestrians Based on Orientation Attributes of SURF

Chadly Marouane[1(✉)], Robert Gutschale[2], and Claudia Linnhoff-Popien[2]

[1] Ludwig Maximilian University, VIRALITY GmbH,
Rauchstraße 7, 81679 Munich, Germany
marouane@virality.de
[2] Ludwig Maximilian University, Oettingenstraße 67, 80538 Munich, Germany
gutschale@cip.ifi.lmu.de, linnhoff@ifi.lmu.de

Abstract. With the decreasing size and prize of cameras, visual positioning systems for persons are becoming more attractive and feasible. A main advantage of visual methods is that they can be independent of any infrastructure and are therefore applicable in indoor as well as outdoor scenarios. As such, they are an attractive alternative to infrastructure based methods. This paper presents a method that uses visual data to create a two-dimensional trajectory of the pedestrians movement. A camera that is mounted on the persons upper body is used to obtain the image data. With the SURF algorithm, feature points that posses specific attributes are extracted from the image frames. Based on these attributes, the method determines the pedestrians steps and estimates the heading at each step. As each determination of a new position is based on previous estimations, the method accumulates errors with increasing distances. An extensive evaluation with different test persons for various scenarios demonstrates that the method achieves a reasonably good overall accuracy for shorter distances. For distances of up to 25 m, a mean error of 5.52 m for indoor scenarios and of 7.56 m for outdoor scenarios has been determined. Furthermore, the method is also reliably functional at increasing walking and running speeds. An additional evaluation shows the usability of one of the SURF-attributes for the implementation of an activity detector for different movement speeds. The method robustly detects steps with a high accuracy at an error rate of approximately one percent. However, just like other Pedestrian Dead Reckoning methods, the heading estimation proves to be challenging and to be a source of errors.

Keywords: Visual odometry · Visual pedometer · Position measurement

1 Introduction

Due to the increasing popularity and high demand of location based services, there is a very active body of research and development of various methods,

Y. Bi et al. (eds.), *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*,
Lecture Notes in Networks and Systems 16, DOI 10.1007/978-3-319-56991-8_13

using many different sensors and signal data. In the last years, methods that use visual data from a camera are gaining in popularity. Nowadays, cameras can be built at a small scale while still being affordable. This makes them perfectly suited to be used for tasks such as localization and navigation of smaller robots, micro aerial vehicles and persons. A huge advantage of visual methods is, that they can be completely independent from infrastructure. Many indoor and outdoor positioning systems use external signals, such as GPS, WIFI, or signals from customized beacons. While there are systems that achieve a satisfactory accuracy, their dependence on external signals makes them susceptible to failure when that signal is not available. The most prominent example of this is GPS, which is usually completely unusable inside of buildings. Many visual methods exist that determine the position and orientation of the camera and its movement over time. Often, a trajectory of the movement is created. This process is called Visual Odometry. The early research in this field was motivated by Moon and Mars rovers, but the decreasing size and prize of cameras draws a lot of attention on methods that work on cars, small robots and micro aerial vehicles. But also Visual Odometry systems for persons are gaining more and more attention, as virtually all mobile phones nowadays are equipped with a camera and even additional cameras are easily and comfortable wearable, for example in the form of Google Glass or GoPro cameras. Another well established approach to track the movement of a person is found in the field Pedestrian Dead Reckoning. Methods of this field consist of a pedometer, that detects when a person took a step and a heading provider, that estimates the heading of the person. Combined with the step length, a trajectory of the pedestrians movement is created. This field is popular, as microelectromechanical systems - which are used to produce the signal data - are nowadays available at small sizes and reasonable prizes, which makes them easily wearable.

The goal of this work is to develop a localization system for pedestrians, based on visual data. To process the visual data, the algorithm SURF - Speeded Up Robust Features - is used. Contrary to typical Visual Odometry methods, the recreation of the cameras movement is based on a step detection method. As the method therefore contains elements from Pedestrian Dead Reckoning as well as Visual Odometry systems, it is classified as a crossover of both methods, labeled Pedestrian Visual Odometry. The main focus is to examine the behaviour of the SURF-attributes *orientation* and evaluate this with respect to their usability for the localization system. Based on the results, an algorithm is developed and implemented that creates step-precise trajectories of the camera-wearing pedestrian. Finally, the developed method is evaluated on visual data of indoor and outdoor paths, created by multiple persons.

The work is structured as follows. First Sect. 2 describes the various foundations, concepts and methods this work is built upon, followed by a presentation of related work in Sect. 3. Section 4 details the basic concept, which is based on the SURF-attribute *orientation*. The evaluation is presented in Sect. 5. Its main part consists of an indoor and two outdoor scenarios, with videos recorded by five different persons. Finally, a conclusion of this work is given in Sect. 6.

2 Foundations

This work builds on various foundations, concepts and methods from the fields of image processing, Visual Odometry and Pedestrian Dead Reckoning.

2.1 Image Processing - Interest Points

Although there exists no universal, explicit definition of an interest point in the literature, the underlying concept can be clearly described. The goal is to find distinctive and reproducible points, that reduce the image to a set of certain features. These locations can then be used to describe the characteristics of the image. Usually, distinctive features are can be found at edges, corners and blob-like structures. A blob is a region of pixels in an image that are distinct from its surrounding considering certain properties such as the brightness [29]. Methods in this field typically consist of a detector and a descriptor. The detector defines how a feature point is found, the descriptor describes the characteristics at the location of the feature point. Typically, an image patch of a certain size and shape around the interest point is taken and described in a specified way, based for example on the histogram or the spatial frequency. If an interest point should be found in another image, for example for object recognition, a matching step is executed. In this step, corresponding feature points are calculated for two images, based on specific similarity metrics. A selection of influential and popular algorithms for interest point detectors and descriptors are the *Canny Edge Detector* [12,13], several *Corner Detectors* [18,34,36,38,46,47], *SIFT* [25, 31,32,37], *SURF* [1,4–6] and the *Binary Descriptors* [2,9,11,19,28,45,48].

2.2 Visual Odometry

The term Visual Odometry (VO), coined by the landmark paper of Nister et al. [40] is inspired by the concept Odometry. Where classic Odometry uses motion sensors such as wheel encoders to estimate the movement, VO relies on visual data obtained from a single or multiple cameras. The goal is to use image sequences to sequentially estimate the camera-movement between image frames. Visual Odometry is used and closely related to the tasks of SLAM (simultaneous localization and mapping). An inherent problem of the classic Odometry is its dependency on favorable ground conditions. For example when using wheel encoders on a slippery or uneven ground, wheel slippage might occur which reduces the accuracy. Early work in the field Visual Odometry was inspired by this problem and motivated by the development of Moon and Mars rovers, where harsh ground conditions are to be expected. More recently, Visual Odometry is used in autonomous cars, unmanned aerial vehicles and micro aerial vehicles.

The typical steps of a VO method are illustrated in [49]. An image sequence is obtained, using a single or multiple cameras. On every new image of the sequence, a feature detection and a feature matching or feature tracking needs to be performed. In the next step, the camera motion between two frames is estimated. Based on the matched or tracked features of two images that represent

the same 3D feature, the relative movement of the camera is calculated and the trajectory is updated with the current camera position. To improve the accuracy of the created trajectory, an iterative refinement over a specified number of last images can be done.

2.3 Pedestrian Dead Reckoning

As the name implies, Pedestrian Dead Reckoning (PDR) is the process of determining the position of a pedestrian based on previous estimations. PDR methods typically consist of three elements: step detection, step length estimation and heading estimation [17]. The data from these elements is then combined to continually update the position of the pedestrian. As all Dead Reckoning methods, PDR relies on previous position estimations and therefore accumulates errors. Step detection, the first element, is on its own a very active research and development field with various commercially available solutions, called pedometers. A pedometer is typically an external portable device or integrated in personal devices such as MP3-players, mobile phones or fitness wearable devices. They are mostly either hand-held or bodymounted, only a few approaches achieve satisfactory results for unrestricted use [10]. Bodymounted positions are for example on a shoe, the hip, or a helmet. The position of the device results in certain advantages, challenges and restrictions.

Pedometers use data from one or multiple sensors from which the steps are extracted. Nowadays, the usage of inertial measurement units (IMU), such as an accelerometer is very popular. Due to the continuing developments of microelectromechanical systems (MEMS), those sensors can be produced at small scale and relatively low costs. Often, to optimize the sensor-data, it is filtered, for example with a lowpass filter. To identify steps, usually peaks or zero-crossings of the accelerometer signal are detected. In the second element of PDR, step length estimation, the covered distance of each step is estimated. It can be sufficient to assume a constant step length, which can either be measured directly for different persons or be found during a calibration phase. If more precision at the step-level is necessary, the dynamic step length is estimated using for example the step frequency, vertical velocity, acceleration magnitude, or a combination of such signals. Another approach is to model the movement of the pelvis as an inverted pendulum and estimate the step length from its vertical movement. The third element, heading estimation, provides the orientation the pedestrian is headed. Such a compass module is often achieved using magnetometer or gyroscope data. The main challenge is the reliability of the data signal, as it is influenced for example by magnetic variances due to the surrounding [16,23], the persons movement or by the tilt of the measuring device [44].

3 Related Work

In this section, a selection of related work and methods from the fields Visual Odometry, Pedestrian Dead Reckoning and Visual Positioning Systems is presented.

3.1 Visual Odometry

Autonomous robots and especially Moon and Mars rovers have been a source of motivation since the early Visual Odometry systems, such as [38]. Corke et al. [14] use an omnidirectional camera to estimate the motion of the rover. Maimone et al. [33] describe the Visual Odometry system that is used in the Mars rovers *Spirit* and *Opportunity*. More recently, the usage of Visual Odometry in cars, and micro aerial vehicles (MAV) gained popularity. Kitt et al. [26] estimate the egomotion of a car in all 6 degrees of freedom, using a stereo-camera approach. Scaramuzza et al. [50], as well as Nourani-Vatani et al. [41] reduce the complexity, by restricting the motion model, using the Ackermann principle.

Visual Odometry systems where the camera is carried by a person, are also developed. Oskiper et al. [42] use two stereo cameras, one facing forwards, the other backwards, to estimate the persons egomotion.

3.2 Pedestrian Dead Reckoning

A central question for body-worn pedometers is the placement on the body, as it can yield certain characteristics and advantages. Foot-, or shoe-mounted pedometers are popular, as zero-velocity phases can clearly be identified. They occur when the pedestrian is standing still or when the foot hits the ground during walking.

Jimenez et al. [20] compared foot-mounted MEMS-based PDR algorithms and concluded that the main error-source for positioning errors occur during the heading estimation. In [21], they propose a PDR system with a drift correction based on the detection of ramps, which are often found in buildings. Bebek et al. [8] use a pressure sensor in addition to the inertial measurement unit, to more reliably detect zero-velocity phases. Beauregard [7] uses a helmet-mounted IMU sensor. The PDR system is calibrated and validated using GPS, while the step length is predicted using a neural network. The usage of mobile devices such as smartphones or MP3-players motivates the development of PDR-systems without special hardware such as boots or helmets. Methods, where the mobile device is held in the hand, such as Pratama et al. [43] proposed, as well as where the device is placed at the hip and waist area. Jin et al. [22] for example placed it inside a trouser pocket of the pedestrian. UPTIME, developed by Alzantot and Youssef [3], uses a support vector machine to differentiate between walking, jogging and running. It can cope with an arbitrary phone orientation and uses a gait-based dynamic step length estimation. A different approach to acquire sensor data for the pedometer has been investigated by Liu et al. [30]. Based on similar work from DiVerdi and Hollerer [15], they use the image sequence of a body-mounted camera. To produce the signal data, a matching of SIFT features is applied and the spatial movement of the matches is extracted. Based on this periodic data, the steps are counted.

3.3 Visual Positioning Systems

Visual positioning systems aim to achieve localization based on visual data. They either use a camera infrastructure to locate people, such as used in the EasyLiving project [27]. The other approach is to determine the position based on image data obtained from a camera-carrying person. Signpost, developed by Mullonie et al. [39] requires the user to point the camera of their smartphone at special markers, that encode the positioning information. Such systems can only position a person at discrete locations, where a marker is deployed. Kawaji et al. [24] use omnidirectional panoramic images and determine the current position from the closest match in an image database. MoVIPS, presented in [51] and improved upon in [35] uses SURF and a distance estimation algorithm on images obtained from mobile phones. This method determines the position and the orientation of the person based on the closest match of a database which is populated with geo-referenced images.

4 Concept

This section describes the developed method for *Pedestrian Visual Odometry*. The fundamental goal is to extract a step-precise trajectory of the movement of a pedestrian from a video-input. In the following subsections, the basic workflow and concept are first presented and then described in more detail.

4.1 Basic Workflow

The method follows a pipeline, adapted from [49]. First, an `Image Sequence` needs to be obtained. In the second step, a `Feature Detection` is performed on each frame of the sequence, followed by a `Feature Matching`, or a `Feature Tracking`. The last step covers the actual `Motion Estimation`, which constitutes the major portion of this work. The implementation of the last three steps follows the basic concept outlined in Fig. 1. `Feature Detection` and `Feature Matching/Tracking` of the pipeline are combined into one process, named `SURF Extraction and Matching`. Before the `Motion Estimation`, a `Preprocessing`-step is applied on the SURF-attributes. The `Motion Estimation` consists of two processes, `Step Detection` and `Heading Estimation`. The former detects the steps of the pedestrian, using the SURF-attribute *orientation*. The latter estimates the direction in which the pedestrian is heading, using the x-attribute of the interest points. As the final step, `Trajectory Building` builds the trajectory, based on the step- and heading-estimation-data of the previous processes.

4.2 Image Sequence and Feature Detection

The video-data is obtained using a camera with a body mount. The camera is statically mounted to the chest of the pedestrian, which means it always points in the walking-direction. The described method can only detect forward-steps.

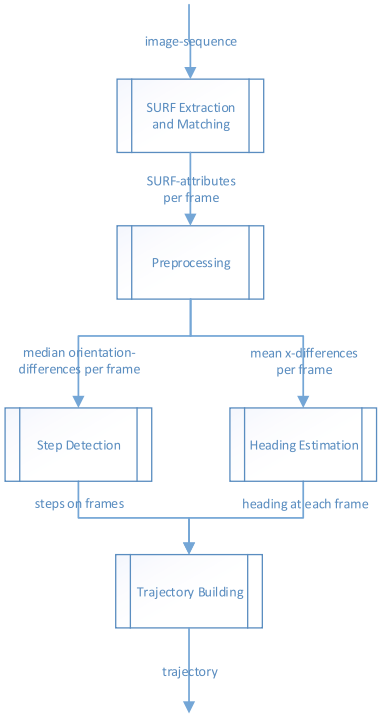


Fig. 1. Overview of the basic concept.

Other kind of movement, such as lateral steps, where the camera-view doesn't point in the walking direction, or backwards steps can not be detected. For the image processing, the SURF-algorithm is used. The image-sequence is iteratively processed. At each frame, the interest points are detected, then described and finally matched with the interest points of the previous frame. Resulting is a sequence of interest point matches for all frame pairs, each containing the SURF-attributes *orientation* and x, which are relevant for the next steps. Additionally a `tracking window` with a size n can be applied at each frame i. If n is greater than 1, each interest point is compared to its tracked match from frame $i - n$.

4.3 Preprocessing

A number of preprocessing steps are necessary on the SURF-attributes, before `Step Detection` and `Heading Estimation` can be performed.

The preprocessing step sums up the two-dimensional lists of the SURF-attributes *orientation* and x of all interest points in one frame into a one-dimensional measure. As a measure, the mean, or the median of all values in one frame is applicable. The medians of the SURF-*orientations*, which are used by the `Step Detection`, are subsequently filtered using a Butterworth Lowpass Filter.

4.4 Motion Estimation - Step Detection

As introduced in Sect. 4.1 and illustrated in Fig. 1, the `Step Detection` uses the SURF-attribute *orientation* to detect the pedestrians steps. Due to the `Preprocessing` phase, its input is a lowpass filtered sequence of medians of the SURF-*orientations* for each frame. The output is a sequence of the framenumbers in which a step was detected and their corresponding *orientation* value. Figure 2 shows a detailed illustration of the process.

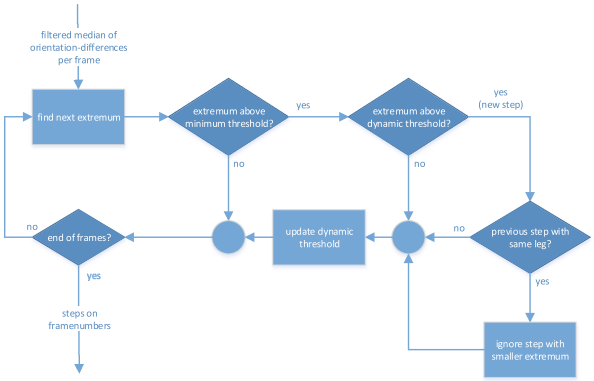
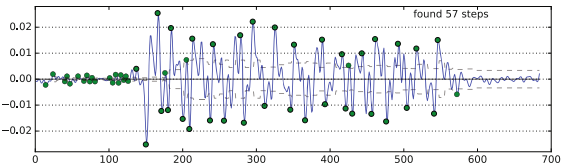


Fig. 2. Overview of the `Step Detection` process.

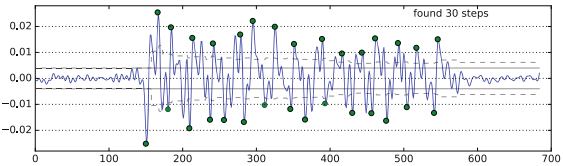
Essentially, the input sequence is searched for extrema, as they correspond to the steps of a pedestrian. A peak-extrema with a positive value is a step with the left leg, a valley-extrema with a negative value is a step with the right leg. All valley-extrema with a positive value are ignored, as well as all peak-extrema with a negative value. When a new extremum is found, a few tests are performed to determine if it counts as a new step. First, it needs to be decided, whether the value of the extremum is in a range that should count as a step, or whether it can be disregarded as noise. To achieve this, two thresholds are applied. The first threshold, δ_{min}, is static, with a predefined value. If the extremum has a value that is below δ_{min}, it is considered as noise. As the camera is mounted directly on the chest of the pedestrian, even when the person is standing still there is some movement due to breathing, or other small movements of the upper body. With δ_{min}, this kind of false positives can be significantly reduced. Although the static threshold alone can yield good results, it is not sufficient to reliably detect all steps of the pedestrian. As different people have different walking styles, the extrema also show variations in their values and characteristic step patterns. Due to this effect, a second, dynamic threshold, δ_{dyn}, is applied. If the extremum is greater than δ_{dyn}, it is considered as a new step. When two consecutive steps are found that were made with the same leg, i.e. two consecutive positive or two consecutive negative extrema, one of them is disregarded. It is assumed, that steps are always taken alternately and

therefore the extrema should also occur as such. In a case where this assumption is violated, the extremum with the bigger value is considered as the step, the other one is ignored. The dynamic threshold is updated, when an extremum is found that lays outside δ_{min}. For the update, a specified number of previous steps are considered. For each previous step-pair (i.e. one step with the left and one step with the right leg), the absolute value of the difference of its *orientation*-values is calculated. The new value for δ_{dyn} is then taken as a ratio of the median of those differences: $ratio * median(absolute\_differences)$.

Figure 3 shows two example plots of a video, processed with different δ_{min} settings. The pedestrian first was standing still for a few seconds, then took 30 steps - beginning with the right leg - and again stood still for a few seconds. The figure shows one plot without the minimum threshold (3a) and one plot where δ_{min} is set to a value, so that the noise in the beginning is ignored (3b). It can be observed, that due to the minimum threshold, not only the noise while standing still is ignored, but also the dynamic threshold increases much quicker as soon as the first step-pair is detected and is not updated as frequently. When no δ_{min} is used, a lot of false-positive steps with very low values are discovered in the beginning and δ_{dyn} only reaches a reasonable value after a certain number of correct steps are detected. In the meantime, more false steps might be found, as it is the case in plot 3a at approximately framenumber 180. Additionally, as δ_{dyn} is updated on every step that lays over δ_{min}, another problem appears when no δ_{min} is used. Due to the too frequent updates, δ_{dyn} might decrease to a value that leads to more falsely detected steps. This occurs in plot 3a at framenumber 200 and again at framenumber 430, where each an additional step-pair is detected.



(a) No minimum threshold δ_{min}



(b) Minimum threshold δ_{min} set, so that noise gets ignored

Fig. 3. Two `Step Detection` plots of the same video (30 steps). The *SURF-orientation* values are plotted for each frame, with the framenumber on the x-axis and the value on the y-axis. Plot 3a shows the `Step Detection` without a minimum threshold δ_{min}, plot 3b with a minimum threshold. The orientation values are plotted with a blue line, the detected steps are marked with green dots. The dynamic threshold δ_{dyn} is plotted with a dashed gray line, δ_{min} with a solid gray line.

Those effects lead to a drastically higher number of detected steps, while the usage of δ_{min} in plot 3b results in the correct number.

4.5 Motion Estimation - Heading Estimation

To estimate the direction the pedestrian is heading to and its changes over time, two methods are used. The first is a naive algorithm, the second uses the regions, as described in Sect. 4.3. As the input, both methods use the sequence of medians of the x-attributes over all frames, as shown in Fig. 1 and the detailed illustration of the `Heading Estimation` in Fig. 4. The heading estimations are then processed for each frame.

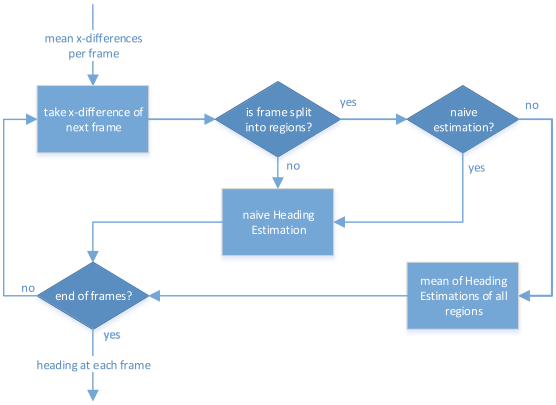


Fig. 4. Overview of the `Heading Estimation` process.

The usage of the x-attribute follows a basic assumption. As the camera is body-mounted to the chest of the pedestrian, its coordinate system is coupled to the upper-body movement of the person. When walking, the upper-body also swings to a certain degree to the sides. When walking a curve, the camera changes its direction at the same rate as the person. Both are movements that are mainly parallel to the ground. This means, that the principal motion of the interest points between two successive image-frames is occurring in the direction of the x-axis. When walking a straight line, the thusly occurring small changes in x-direction ideally cancel each other out every two steps. A change of direction from the pedestrian – either while standing or while walking – produces mainly a prolonged increase or decrease in the x-attributes of the interest points. This motivates the assumption, that it is sufficient to only use the x-attribute for a reliable heading estimation. The naive `Heading Estimation` uses the median of x-differences. It is then divided by the `heading calibration`, which is dependent on the camera and the resolution of the image-frame. This parameter specifies how many pixels of movement in the x-direction correspond to a turn of

one degree. If the horizontal field of view of the camera is known, the value of the `heading calibration` can be directly calculated by dividing the horizontal resolution for the frame by that value. Section 5.3 provides more details on this.

4.6 Motion Estimation - Trajectory Building

The last step is to combine the results of the `Step Detection` and `Heading Estimation` steps into a trajectory. It receives two input data sequences, one containing the framenumbers where a step was detected and the other containing the heading estimation at each framenumber. Additionally, the step-length of the pedestrian needs to be specified. This is a fixed parameter, which should be adjusted for each person. Section 5.3 details how this parameter was determined for the test-persons of the evaluation. The sequence of steps is now iterated. At each step, the corresponding heading estimation is taken from the heading estimations sequence. The coordinates for the detected step are calculated using the sinus- and cosinus-functions:

$$x_{step_i} = x_{step_{i-1}} + step\_length * sin(heading\_estimation_{step_i})$$
$$y_{step_i} = y_{step_{i-1}} + step\_length * cos(heading\_estimation_{step_i})$$

5 Evaluation

In this section, the developed method for Pedestrian Visual Odometry is evaluated. On the basis of three scenarios, where the paths were recorded multiple times with different people, the accuracy is determined in terms of spatial- and heading-differences.

5.1 General Set-Up

To record the videos, a GoPro Hero3+ camera was used. Its video capture mode was set to a format of 16×9, with a resolution of 1920×1080 and a narrow field of view. This results in minimal fisheye distortion effects. With this setting, the camera captures videos at 25 frames per second. A GoPro camera was chosen, as it can easily be carried by a person with the use of a body mount. The resolution has been changed to 640×360 during the `SURF Extraction and Matching` step. A resolution below that value often resulted in significantly decreased accuracy. The three scenarios of the evaluation are one indoor and two outdoor paths. These paths are described and illustrated in Sect. 5.4. The video-data for those scenarios was created with five persons - one female and four males - differing in body height. The test data-set consists of 29 videos for the indoor and outdoor path. The ground truth data for the steps was counted with a manual counter.

5.2 Evaluation Method

To evaluate the developed method for Pedestrian Visual Odometry, the accuracy of the step detector and the accuracy of the trajectory needs to be compared with the ground truth data. The real steps were counted with the help of a mechanical tally counter. As the person who is recording the video should be influenced as less as possible, the steps were counted by another person. The ground truth data for the trajectories was created with the help of map data of the building, for the indoor path and map data provided by OpenStreetMap for the outdoor paths. Each map is stored as a size-fixed image with a corresponding calibration value that encodes how many pixels represent one meter. Each ground truth trajectory consists of a series of (x, y)-coordinates, based on its map image. For the accuracy evaluation of the step detector, the resulting value for each video can simply be compared to the hand counted ground truth data. The aforementioned videos where no ground truth data for the steps exists are ignored during this step. To evaluate the accuracy of the resulting trajectory, a more complex method has been employed. For each video, 25 measurements are calculated, that each analyses a segment with a specified length of the trajectory. Each measurement determines the euclidean difference and the difference in heading at a number of points, compared with the ground truth trajectory. For the indoor path, a segment length of 50 m was chosen, with measurement points at a distance of 5, 10, 25 and 50 m from the start of the trajectory segment. As the outdoor paths are longer, a segment length of 100 m was chosen, with measurement points at a distance of 5, 10, 25, 50, 75 and 100 m from the start of the trajectory segment. To ignore accumulated errors, the resulting trajectory is reset to the ground truth at the start of the trajectory segment. This reset point for a measurement is chosen randomly, with the only restriction being that its distance to the end of the trajectory must be at least 50 m for the indoor path and 100 m for the outdoor paths. As the developed method for Pedestrian Visual Odometry determines the current position based on previous position estimates, it accumulates an error, called drift. The evaluation method was chosen, as it independently evaluates trajectory segments of a fixed length and at fixed distances. On the one hand, this makes it possible to evaluate various segments of one trajectory without previously accumulated drift and on the other hand, it makes it easy to compare the accuracy for different paths and scenarios.

5.3 Parameter Settings

(1) *SURF-Parameters:* The SURF algorithm provides a few parameters that may be configured. With the exception of the `threshold`, all other parameters were left at their default value of the OpenSURF implementation. With the SURF-`threshold`, the number of extracted interest points can directly be manipulated. While a lower `threshold`-value yields to the extraction of more interest points, they are also more noisy and therefore may be less descriptive. Although the accuracy of the indoor trajectories were improved up to a certain point by reducing the threshold, a more thorough evaluation

is necessary. The used indoor test data is a very specific environment with a narrow corridor, a lot of white walls and only a small part inside a larger room. It should be evaluated whether the conclusion that a reduced SURF-`threshold` produces a higher accuracy holds true for a variety of indoor environments. For the outdoor test data the reduction of threshold has no influence.

(2) *Lowpass Filter Cutoff Frequency:* The cutoff frequency was evaluated beginning at a value of 0.5 Hz, with increments of 0.5 Hz. A small cutoff frequency of around 1.5 Hz yields extremely smooth plots. Additionally, at that frequency the step plot doesn't show any characteristic local extrema at the peaks and valleys and is a sinus-like curve with varying amplitudes. While this would simplify the step detection, it also cuts away too much information, leading to falsely suppressed steps. Therefore a higher cutoff frequency of 4 Hz was chosen, as it is a good compromise of filtering while preserving the characteristic step data.

(3) *Thresholds for the Step Detection:* The minimum threshold is set to a low value, to reduce falsely detected steps where δ_{dyn} is less than δ_{min}. This usually only occurs at the beginning of a video, before the person took any steps. To eliminate the maximum of false positives, while also not suppressing real steps, an individual value for δ_{min} should be set for each person. A good value for δ_{min} has been found at a quarter of the *orientation*-value of an average step. At each newly detected step, δ_{dyn} is then updated using the following formula: $\delta_{dyn}^{new} = 0.25 * median(\Delta_5)$, with Δ_5 being the pairwise differences in the *orientation*-attribute of the last 5 steps. This method produced good results for all five persons.

(4) *Step Length:* As the step length is not dynamically estimated, it was statically measured. Each of the five persons was asked to take a number of steps at their normal walking speed. The distance of 10 steps was measured, from which the average step length was calculated. This process was repeated so that at least three measurements were obtained for each person. From these, the final estimation of the average step length for each person was calculated by a proposed formula from Pratama et al. [43].

(5) *Tracking-Window:* Early on, a tracking window of size 3 or 4 yielded visible accuracy improvements of the trajectories. A more thorough evaluation confirmed those results. The videos were processed with increasing tracking windows of size 2 to 5, a window size of 1 equals a regular matching. Table 1 shows the average improvements of different tracking window sizes, evaluated on trajectory segments with a length of 50 and 100 m. It can be seen, that higher tracking windows greatly increase the euclidean error and its standard deviation at shorter distances. Improvements of up to 12.3% for the error and 23.1% can be observed. Although the mean error is only slightly improved at a distance of 100 m, still its standard deviation is greatly improved. The heading estimation generally is also improved by up to 3°, while its standard deviation gets worse for shorter, but then increases for larger distances.

Table 1. Improvements for various tracking windows at distances of 50 m and 100 m. E-P lists the improvement in percentage of the euclidean error, E-STD lists the improvements of the euclidean errors standard deviation. H-D lists the difference in the heading estimation, H-STD lists the difference of the heading estimations standard deviation, negative values indicate an improvement. Values for a tracking window of size 1 are not listed, as this corresponds to a regular matching.

Tracking window	50 m				100 m			
	E-P	E-STD	H-D	H-STD	E-P	E-STD	H-D	H-STD
2	−2,5	5,4	6,7	1,4	3,2	1.1	9.2	−2,6
3	10,9	10,6	−0,1	0,8	2.6	13.3	−2,3	−3,3
4	9,8	19,8	−2,25	2,75	1.6	21,8	−3,0	−4,9
5	12,3	23,1	−2,1	2,9	−3.8	43,4	−2,0	−6,2

For the developed method, a tracking window of size 3 was chosen. This results in great improvements, while also maintaining about 40% of the original interest point matches. A tracking window of size 4 only contains about 28% of the original matches. As a higher number of matches principally results in more robustness of the developed method, a tracking window of size 3 was preferred over higher ones.

(6) *Heading Estimation:* The horizontal field of view (FOV) of the camera is used. The specified mode of the GoPro Hero3+ records videos with a horizontal FOV of 64.4°. The heading calibration is calculated with the following formula: $heading\_calibration = horizontal\_resolution/64.4$.

5.4 Scenarios

To determine the accuracy of the method, the three scenarios have been evaluated. The videos for all three scenarios were inside or outside an university building. The following sections detail the results for the indoor and the two outdoor scenarios. Each section describes the evaluated path and presents, as well as discusses the result. The same parameter settings were used for all scenarios, the results were obtained using a tracking window of size 3.

(1) Path1- *an Indoor Path:* The indoor path is shown in Fig. 5. It is located at the basement level of the building and is mostly a narrow path with white walls. Its length is 148.50 m. No closed doors or other such obstacles are passed, although the person recording the video sometimes has to evade other people or obstacles on the ground. The middle of the path - curves 2 to 6 in Fig. 5 - lays inside the cafeteria. This is a contrasting environment, as it is a large room, with different objects such as tables, chairs and vending machines. On some videos *path1* has a small variation, illustrated by the dotted line in Fig. 5, as some parts of the cafeteria were closed during the filming of those videos.

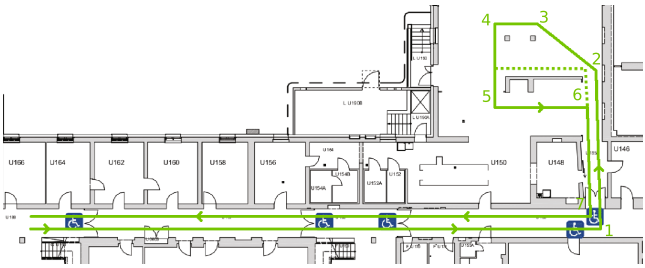


Fig. 5. Overview of *path1*. In some videos *path1* has a small variation, as some parts of the cafeteria were closed, illustrated with the dotted line between curves 5 and 6. The walking direction is indicated by the arrows, the curves are numbered.

For the first part up to curve 1, it can be observed that all test videos accumulate a drift error to the left at approximately the same rate. The huge difference of the total accuracy is because of a higher drift between curves 5 and 6 and because of heading errors at curves 6 and 7. Curve 7 produces a huge error on all test videos, even at the good trajectory a turn of only about 45° was recognized instead of 90°. For the step detection, a mean absolute error over all videos of 8.86 with a standard deviation of 21.75 has been calculated. This means, that on average nearly 9 steps over the ground truth data are detected, which corresponds to 5 percent of the total steps. Most of the falsely detected steps occur during prolonged sections where just a few interest point matches are found. During such sections, the *orientation*-signal isn't as periodically and often doesn't show the usual characteristics, leading to a significant number of falsely detected steps. This is shown in Fig. 6, where after a prolonged sequence of an irregular *orientation*-signal and resulting falsely detected steps - especially around frame 1700 -, the characteristic step pattern resumes at frame 1800.

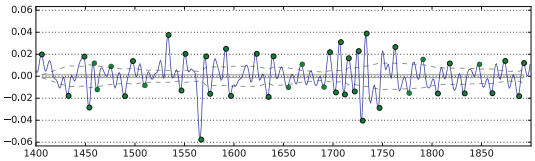


Fig. 6. Exemplary *orientation*-plot of falsely detected steps of *path1*.

The statistical results for *path1* are shown in Fig. 7. It can be observed, that the euclidean errors are relatively small for short distances. At a distance of 5 m it is 0.77 m with a standard deviation of 0.17. Those errors grow to 5.52 m with a standard deviation of 1.19 at a distance of 25 m and to 11.31 m with a standard deviation of 3.72 at 50 m. The mean heading error is overall relatively small, although a significant standard deviation can be observed even for short distances. The mean error steadily increases, which means that mainly drift to

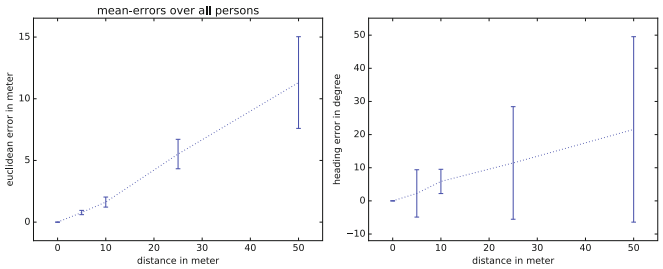


Fig. 7. Results for *path1*. The left plot shows the mean euclidean errors in meter and the standard deviation at the specified distances for all measurements, the right plot shows the mean heading error in degree and the standard deviation at the specified distances for all measurements.

the left occurred. At 5 m the heading error is 2.25°, at a distance of 50 m it is 21.56°. For larger distances, a huge increase in the standard deviation can be observed from 7.14° at 5 m and 3.66° at 10 m, to 27,97° at 50 m.

For this indoor scenario, it can be concluded that the method can produce reasonably good results under certain circumstances. The major challenge seems to be the white walls of the corridor. When walking a straight line, a drift to the left occurs, that accumulates to a large and steadily increasing error. On curves, the effect can be drastically more significant. If the pedestrian is heading towards a white wall at the curve, this often leads to only few or even no interest point matches for individual framepairs. When this happens for prolonged sequences, a low accuracy of the curve results, as can be observed at curve 7 of *path1*. On areas, where the indoor environment has more structure, more interest point matches are found for each framepair. Therefore, the resulting trajectory has a higher accuracy. This can be observed at the cafeteria part of *path1* - curves 2 to 6.

(2) Path2- *an Outdoor Path: Path2*, as shown in Fig. 8, is a symmetrical round-trip path outside the building, with a length of 387.5 m. The first part - between curves 1 and 2 in Fig. 8 - resembles a street with parked cars. The rest resembles a park-like environment, although the building is always on the right, or respectively left side. Only few other people were present during the filming of the videos, usually before curve 1 - a main entrance of the building-, or at curve 3 - the outside area of the cafeteria.

For the step detection, a mean absolute error over all videos of 3.8 with a standard deviation of 13.89 has been calculated. This means, that on average nearly 4 steps over the ground truth data are detected, which corresponds to 0.84 percent of the total steps. Figure 9 shows the statistical results for *path2*. Similar to *path1*, it can be observed that for short distances of up to 10 m only relatively small euclidean errors (0.89 and 2.26 m) with small standard deviations occur (0.22 and 0.47 m). At 25 m the mean error grows to 7.95 m, with a standard deviation of 2.41 and steadily increases to 32.43 m with a standard deviation

Fig. 8. Overview of *path2*. The walking direction is indicated by the arrows, the curves are numbered.

of 4.26 at 100 m. The mean heading errors are all positive, which means that mainly drift to the left occurred. It can be observed, that the heading error is relatively small for all distances and range from 5.06° to 9.67°. The standard deviation ranges from 7.06° to 10.28° for distances of up to 75 m and increases to 20.36° at a distance of 100 m.

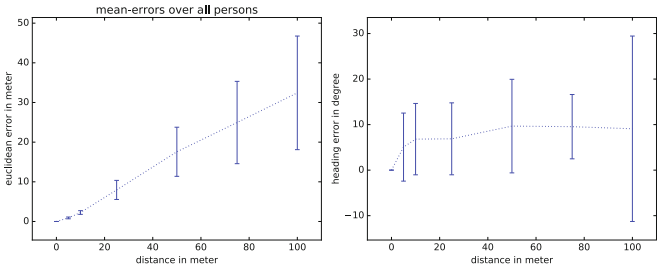


Fig. 9. Results for *path2*. The mean euclidean and heading errors, as well as the standard deviations for all measurements are shown.

For this outdoor scenario, the developed method for Pedestrian Visual Odometry produces good results, that can be extremely close to the ground truth trajectory. The main error source are inaccurately detected curves. Usually the number of interest point matches is very high for each frame-pair. Additionally, there was very little noise in the form of other moving persons, bicycles or cars. These two aspects seem to have a direct effect on the drift of the heading estimation which is constantly low with also a small standard deviation for distances of up to 75 m. Compared with the results of *path1*, the mean euclidean error is significantly higher at a distance of 50 m (17.57 m, compared to 11.31).

(3) Path3- *an Outdoor Path With Noise:* In some cases the trajectories accumulates a slight drift to the right between curves 2 and 3. Additionally, curve

5 isn't mostly recognized to its full degree. The low accuracy in some bad trajectories is explained due to multiple curves that aren't recognized with their real degree and more drift to the left. Curve 2 is recognized with a too large degree. Between curves 3 and 4, an increasing drift to the left can be observed. Curves 4 and 5 on the other hand aren't recognized to their full degree. For the step detection, a mean absolute error over all videos of 3.6 with a standard deviation of 13.02 has been calculated. This means, that on average nearly 4 steps over the ground truth data are detected, which corresponds to 0.9 percent of the total steps. The statistical results for *path3* are shown in Fig. 10. For short distances, the euclidean errors are very similar to those of *path2*. Up to a distance of 50 m, the mean euclidean errors and standard deviations are practically the same. For distances of 75 and 100 m, the errors increase at a higher rate up to a maximum of 44.24 m with a standard deviation of 9.93. The heading error is relatively small for distances of up to 25 m (−0.36°, 5.57° and 3.31°), with small standard deviations at 5 m (4.55°) and 10 m (5.41°) and a larger one at 25 m (12.89°). For increasing distances, the heading errors also increase up to a value of 44.09° with a standard deviation of 32.28°.

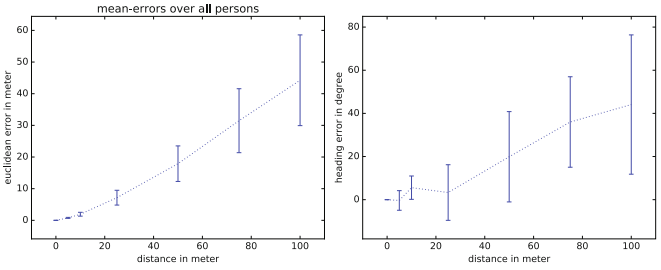


Fig. 10. Results for *path3*. The mean euclidean and heading errors, as well as the standard deviations for all measurements are shown.

Compared with the results of *path2*, the mean euclidean errors and the mean heading errors are significantly higher for increasing distances. This should not be surprising. *Path3* is more challenging, as it is an environment where a large degree of noise in the form of other moving persons, bicycles and cars occurs, that sometimes pass close to the camera or move in front of it. It also contains two challenging curves. The first, curve 4, is a 180° curve during which the pedestrian passes a narrow, but open gate of a white wall and immediately steps onto the sidewalk of a street. It can be observed that on most of the test videos, this curve is detected with a lower degree. The second, curve 5, has a larger radius than other curves and lies directly at the intersection of a street. This results in a lower detected degree for many test videos, probably due to the movement of other persons or cars on the street. Additionally, as the camera view is not obstructed for example by a wall, more interest points are detected that are far away and

don't move much between succeeding frame-pairs, which also contributes to a lower degree.

6 Conclusion

This work describes a method for Pedestrian Visual Odometry that tracks the position of a person based on visual data. The person carries a camera, mounted to its chest, which produces a video stream. The image data is processed using the SURF-algorithm. Based on the SURF-attributes *orientation* and x, the pedestrians steps and the heading at the steps are estimated. The evaluation of the method with three scenarios has shown that it can produce reasonably accurate results for indoor and outdoor scenarios, especially for shorter distances. Just like other Pedestrian Dead Reckoning or Visual Odometry systems, the errors accumulate with increasing distance from a known position. The heading estimation proved to be a great challenge, as the camera is permanently swinging from left to right and as its accuracy additionally depends on the environment. For indoor scenarios, an environment with mostly white walls contributes to a high error of the heading estimation, as well as the step detection. The method is a good candidate to extend existing visual positioning systems, while such systems could also be used to reset the position of the pedestrian, therefore negating the effect of accumulating errors for long distances. The overall quality of the trajectory could be improved by implementing a loop-closure procedure, that finds locations the pedestrian has been before for a retroactive trajectory- and parameter-optimization. Overall it has been shown that a visual pedometer based on the SURF-attribute *orientation* very reliably detects the pedestrians steps. The heading estimation using solely the SURF-attribute x on the other hand is not as accurate. Nonetheless, this method is capable to produce trajectories corresponding to the pedestrians movement with a cumulating error drift at increasing distances.

References

1. Agrawal, M., Konolige, K., Blas, M.R.: CenSurE: Center surround extremas for realtime feature detection and matching. In: Computer Vision - ECCV 2008. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
2. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: fast retina keypoint. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517. IEEE, June 2012
3. Alzantot, M., Youssef, M.: UPTIME: ubiquitious pedestrian tracking using mobile Phones. In: IEEE wireless Communications and Networking Conference 2012 (2012)
4. Bauer, J., Sünderhauf, N., Protzel, P.: Comparing several implementations of two recently published feature detectors. IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 6, pp. 143–148 (2007)
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)

6. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Computer Vision - ECCV 2006, vol. 3951, pp. 404– 417 (2006)
7. Beauregard, S.: A helmet-mounted pedestrian dead reckoning system. In: Applied Wearable Computing, pp. 1–11 (2006)
8. Bebek, Ö., Suster, M.A., Rajgopal, S., Fu, M.J., Huang, X., Cavusoglu, M.C., Young, D.J., Mehregany, M., van den Bogert, A.J., Mastrangelo, C.H.: Personal navigation via high-resolution gait-corrected inertial measurement units. IEEE Trans. Instrum. Measur. **59**(11), 3018–3027 (2010)
9. Bekele, D., Teutsch, M., Schuchert, T.: Evaluation of binary keypoint descriptors. In: 2013 IEEE International Conference on Image Processing, pp. 3652–3656. IEEE, September 2013
10. Brajdic, A., Harle, R.: Walk detection and step counting on unconstrained smartphones. In: Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp 2013, p. 225 (2013)
11. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Proceedings of the European Conference on Computer Vision, ECCV 2010, pp. 778–792 (2010)
12. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **8**(6), 679–698 (1986)
13. Canny, J.F.: Finding Edges and Lines in Images. Technical report (1983)
14. Corke, P., Strelow, D., Singh, S.: Omnidirectional visual odometry for a planetary rover. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), vol. 4, pp. 2–7 (2004)
15. DiVerdi, S., Hollerer, T.: Heads up and camera down: a vision-based tracking modality for mobile mixed reality. IEEE Trans. Vis. Comput. Graph. **14**(3), 500–512 (2008)
16. 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, IPIN 2011 (2011)
17. Groves, P., Pulford, G., Littlefield, C., Nash, D., Mather, C.: Inertial navigation versus pedestrian dead reckoning: Optimizing the integration (2007)
18. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference 1988, pp. 23.1–23.6. Alvey Vision Club (1988)
19. Heinly, J., Dunn, E., Frahm, J.M.: Comparative evaluation of binary features. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS (PART II), vol. 7573, pp. 759–773 (2012)
20. Jiménez, A.R., Seco, F., Prieto, C., Guevara, J.: A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU. In: WISpp 2009–6th IEEE International Symposium on Intelligent Signal Processing - Proceedings, pp. 37–42 (2009)
21. Jiménez, A.R., Seco, F., Zampella, F., Prieto, J.C., Guevara, J.: PDR with a foot-mounted IMU and ramp detection. Sensors **11**(12), 9393–9410 (2011)
22. Jin, Y., Toh, H.S., Soh, W.S., Wong, W.C.: A robust dead-reckoning pedestrian tracking system with low cost sensors. In: 2011 IEEE International Conference on Pervasive Computing and Communications, PerCom 2011, pp. 222–230 (2011)
23. Kang, W., Nam, S., Han, Y., Lee, S.: Improved heading estimation for smartphone-based indoor positioning systems. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, pp. 2449–2453 (2012)

24. Kawaji, H., Hatada, K., Yamasaki, T., Aizawa, K.: Image-based indoor positioning system. In: Proceedings of the 1st ACM International Workshop on Multimodal Pervasive Video Analysis - MPVA 2010, p. 1 (2010)
25. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2004)
26. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 486–492 (2010)
27. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for EasyLiving. In: Proceedings Third IEEE International Workshop on Visual Surveillance, pp. 1–8 (2000)
28. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: binary robust invariant scalable keypoints. In: 2011 International Conference on Computer Vision, pp. 2548–2555. IEEE, November 2011
29. Lindeberg, T.: Detecting salient blob-like image structures and their scales with a scale-space primal sketch: a method for focus-of-attention. Int. J. Comput. Vis. **11**(3), 283–318 (1993)
30. Liu, D., Shan, Q., Wu, D.: Toward a visual pedometer. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing - SAC 2012, p. 1025 (2012)
31. Lowe, D.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
33. Maimone, M., Cheng, Y., Matthies, L.: Two years of visual odometry on the Mars Exploration Rovers. J. Field Robot. **24**(3), 169–186 (2007)
34. Mair, E., Hager, G.D., Burschka, D., Suppa, M., Hirzinger, G.: Adaptive and generic corner detection based on the accelerated segment test. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS (PART II), vol. 6312, pp. 183–196 (2010)
35. Marouane, C., Maier, M., Feld, S., Werner, M.: Visual positioning systems - an extension to MoVIPS. In: 5th International Conference on Indoor Positioning and Indoor Navigation (IPIN 2014), (Section III) (2014)
36. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
37. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1615–1630 (2005)
38. Moravec, H.P.: Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover. Ph.D. thesis (1980)
39. Mulloni, A., Wagner, D., Barakonyi, I., Schmalstieg, D.: Indoor positioning and navigation with camera phones. IEEE Pervasive Comput. **8**(2), 22–31 (2009)
40. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, CVPR 2004, vol. 1, pp. 652–659 (2004)
41. Nourani-Vatani, N., Roberts, J., Srinivasan, M.V.: Practical visual odometry for car-like vehicles. In: Proceedings - IEEE International Conference on Robotics and Automation, (JUNE 2009), pp. 3551–3557 (2009)

42. Oskiper, T., Zhu, Z., Samarasekera, S., Kumar, R.: Visual odometry system using multiple stereo cameras and inertial measurement unit. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8, December 2015
43. Pratama, A.R., Widyawan, Hidayat, R.: Smartphone-based pedestrian dead reckoning as an indoor positioning system. In: Proceedings of the 2012 International Conference on System Engineering and Technology, ICSET 2012 (2012)
44. Randell, C., Djiallis, C., Muller, H.: Personal position measurement using dead reckoning. In: Seventh IEEE International Symposium on Wearable Computers, 2003, Proceedings (2003)
45. Rosin, P.L.: Measuring corner properties. Comput. Vis. Image Underst. **73**(2), 291–307 (1999)
46. Rosten, E., Drummond, T.: Fusing points and lines for high performance tracking. In: Tenth IEEE International Conference on Computer Vision (ICCV 2005), vol. I and II, pp. 1508–1515, vol. 2. IEEE (2005)
47. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). LNCS, vol. 3951, pp. 430–443 (2006)
48. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 International Conference on Computer Vision, pp. 2564–2571. IEEE, November 2011
49. Scaramuzza, D., Fraundorfer, F.: Visual odometry - Part I: the first 30 years and fundamentals. IEEE Robot. Autom. Mag. **18**, 80–92 (2011)
50. Scaramuzza, D., Fraundorfer, F., Siegwart, R.: Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: 2009 IEEE International Conference on Robotics and Automation, pp. 4293–4299 (2009)
51. Werner, M., Kessel, M., Marouane, C.: Indoor positioning using smartphone camera. In: 2011 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2011, pp. 21–23, September 2011