Computers in Industry xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

# Fusion of visual odometry and inertial navigation system on a smartphone

Simon Tomažič *, Igor Škrjanc

*Faculty of Electrical Engineering, Laboratory of Autonomous Mobile Systems, University of Ljubljana, Ljubljana, Slovenia*

ARTICLE INFO

*Article history:*
Received 31 December 2014
Received in revised form 1 May 2015
Accepted 11 May 2015
Available online xxx

*Keywords:*
Visual odometry
Pedometer
Compass
Fusion
Kalman filter
Smartphone
Robot
Pedestrian

ABSTRACT

The paper presents the monocular visual odometry, inertial navigation system and the fusion of both these localization approaches. The visual odometry algorithm consists of four other algorithms, namely the camera calibration algorithm, KLT algorithm, algorithm for the estimation of rigid transformation and RANSAC algorithm. The inertial navigation system is based on a pedometer and digital compass. Both visual odometry and the inertial navigation system can determine the incremental movements and the positions of a robot or a pedestrian according to the world coordinate system. In order to get an even more robust and accurate localization system, the advantages of each mentioned localization approaches were combined by using the Extended Kalman Filter. The algorithms were fully implemented on a smartphone, where they were divided into several threads that could be performed simultaneously on multiple processor cores. The proposed system, which fuses information from the camera and inertial sensors, can convert the smartphone into a powerful mobile sensor unit or the so-called virtual sensor that returns relative position in relation to the starting point. This virtual sensor can be used as an advanced sensor unit on mobile robots or as part of a smartphone application which requires personal navigation system. The operation of the localization system is proved by experimental results which were obtained by attaching a smartphone on a pedestrian who walked along the reference trajectory drawn on the floor. In the experiments the described system showed big potential in many aspects since very good results were obtained.



## 1. Introduction

One of the most important features of all living creatures on Earth is the ability to determine their own position in the environment they live in. With the development of new localization algorithms and systems, there has been a growing tendency to enable robots, autonomous mobile systems, and particularly people who have lost this ability, e.g. due to blindness, to use this ability.

Recently, a lot of studies have focused on localization in the indoor environment, since this represents a major challenge, mainly due to the fact that GPS signals are not available there. As smartphones have become indispensable accessories of a modern man, the possibility of using their hardware for the purpose of indoor localization of persons and autonomous mobile systems has been studied. Smartphones are equipped with multiple sensors such as the accelerometer, gyroscope, magnetometer, altimeter, camera and multiple communication modules (Bluetooth, WiFi, NFC, LTE), which enable the implementation of different algorithms for indoor localization [1]. In addition, smartphones contain increasingly powerful multi-core processors, which allow for the implementation of more complex algorithms.

The development of new algorithms and systems for localization in the indoor environment is especially important due to their usefulness in many fields. Namely, an algorithm that provides accurate localization in the indoor environment can be used for different types of robots, for applications for the blind, for applications which comprise personal navigation system (PNS) for guiding [2] in large shopping malls, museums, airports, public institutions, etc. since people often spend a lot of time finding the desired location in an unfamiliar environment.

Localization of unmanned ground vehicles and robots in the indoor environment is already well developed, since they can be equipped with more powerful hardware and additional sensors, e.g. LIDAR, depth sensor, stereo camera [3]. In connection with the LIDAR sensor, especially simultaneous localization and mapping (SLAM) method [4] has been established, which can also process the information obtained by the camera. Since the camera is a low cost and relatively lightweight sensor, its use is becoming more

* Corresponding author.
*E-mail addresses:* simon.tomazic@fe.uni-lj.si (S. Tomažič), igor.skrjanc@fe.uni-lj.si (I. Škrjanc).

http://dx.doi.org/10.1016/j.compind.2015.05.003
0166-3615/

common for the purpose of localization of unmanned ground and aerial vehicles.

Currently the most widely used approach to indoor localization using a smartphone is based on the measurement of the WiFi, Bluetooth or GSM signal strength [1]. However, because this approach does not enable high positioning accuracy, new approaches based on the fusion of several different sensors and methods are being established. As has been shown in numerous studies, a great potential lies especially in the methods based on inertial sensors and camera, as they provide higher accuracy of localization. High accuracy and robustness of localization can be achieved through the fusion of different localization approaches which complement each other. Sirtkaya et al. [5] proposed the fusion of the information from the camera and inertial sensors by using the Kalman filter.

When camera is used for the localization purposes, there are several different methods and algorithms which can determine the movement of an agent (a vehicle, person, robot) on which the camera is attached. Among the established methods, here belong structure from motion (SFM), simultaneous localization and mapping (SLAM) [4], visual odometry (VO) [6] and image-to-map matching [7]. SFM and SLAM are considered to be compute-intensive methods and they also spend a lot of memory, because they build a 3D map of the environment besides the motion estimation. The approach with the image-to-map matching requires extensive image dataset of the indoor environment which is used for off-line reconstruction (i.e. building a map) presented with a 3D point cloud. This type of localization also requires a powerful hardware with a lot of storage [8]. Since visual odometry estimates only the motion of the camera, it can operate in real time, even on less powerful hardware. The concept of visual odometry was established by Nister et al. [9], who introduced the main concept, which is the basis for the most of the existing visual odometry algorithms. The term was chosen for its similarity to wheel odometry since both approaches incrementally estimate the motion of a vehicle.

In order to obtain an accurate localization technique which would run in real time on a smartphone, a computational optimal system based on the use of camera and inertial sensors was built in this study. The core of the system is visual odometry, which enables accurate determination of incremental movements of the smartphone, i.e. a human or a robot on which the smartphone is attached. Visual odometry is based on the assumption that a smartphone is fixed on a certain height and at an angle relative to the floor [10]. This assumption is often true when using a smartphone on a robot. For determining the movements of (the blind) persons, a smartphone needs to be fixed to the body. For the operation of visual odometry, it is necessary to know the transformation between the camera coordinate system (C.S.) and the ground C.S., which is obtained by the initial calibration. In connection with the visual odometry, RANSAC algorithm [9,11] is also often used, which enables the elimination of the traces of feature points that represent outliers in the determination of the rigid motion model.

For the purpose of testing the algorithms, a Galaxy S4 smartphone based on the operating system Android was used. In the implementation of the visual odometry, an open-source library BoofCV [12] that is written in the Java programming language and combines a multitude of useful functions in the field of machine vision was used. The visual odometry algorithm is performed in real time on the smartphone where the speed of image processing is equal to 10–15 fps. In the application the images were captured at a resolution of $320 \times 240$ pixels. The calibration was also performed on the smartphone by using the BoofCV library [12].

As a complementary system of visual odometry, the inertial navigation system based on three sensors, namely 3-axis accelerometer, 3-axis gyroscope and 3-axis magnetometer, was implemented. The inertial navigation system for pedestrians is basically composed of a digital compass, which is responsible for determining the absolute heading, and a pedometer, which determines the length of the travelled path by counting the steps. Inertial navigation system can also be used on a wheel robot, whereby double integration of acceleration for determining the travelled distance is used instead of a pedometer. This approach is somewhat more susceptible to the errors accumulation but this can be reduced by considering another localization technique such as visual or wheel odometry.

Both visual odometry and inertial navigation system have advantages as well as disadvantages. Both approaches have one common weakness, namely, they belong to dead reckoning localization techniques. The phrase dead reckoning means that a previously determined position is used in the process of calculating the current position. Consequently, the newly computed positions contain cumulative errors. With the fusion of both localization approaches, the size of the accumulated errors can be reduced and also all other errors that may occur within a particular system should be eliminated. In order to get a system that would be as reliable as possible for determining the relative position, the Extended Kalman Filter (EKF), which enables optimal combining of information, was used for the fusion purposes.

The advantage of this system is that it is fully implemented on the smartphone and it does not require additional indoor infrastructure (e.g. WiFi network) for its operation. The proposed localization system, which combines information from hardware sensors (camera and inertial sensors) converts the smartphone into a powerful mobile sensor unit or the so-called virtual sensor that returns incremental movements in relation to the starting point. Therefore, the smartphone just needs to be attached to the robot and the virtual sensor already determines incremental movements of the robot. The result of the virtual sensor can be sent wirelessly, e.g. via ROS messages [13], to the central unit of the robot, where it is used in the control algorithm. The smartphone could work also in a telerobot (remote) control mode [14], wherein the whole control algorithm [15] would run on it and only commands according to the desired task would be sent to the autonomous ground vehicle. The above mentioned virtual sensor can easily be applied also in the application for guiding humans [16,17] in an indoor space.

The operation of the localization system is proved by experimental results which were obtained by attaching a smartphone on a pedestrian who walked along the reference trajectory drawn on the floor. When using a smartphone as an external sensor unit on a robot, equally good or even better results are expected than when using it on a pedestrian due to much less vibrations.

In the following sections the components of the monocular visual odometry are first presented, then visual odometry itself. Afterwards, the inertial navigation system which includes a digital compass and a pedometer is described as well as the fusion of visual odometry and inertial navigation system. Finally, the experimental results of the implemented system functionality are given.

## 2. Monocular visual odometry

Monocular visual odometry is the sequential estimation process of camera motions depending on the perceived movements of pixels in the image sequence. The visual odometry consists of four algorithms, namely: the camera calibration, the

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

feature tracker, the algorithm for the estimation of a rigid motion model and the RANSAC algorithm.

### 2.1. Camera calibration

If it is assumed that the camera has a thin lens, then it can be described with the pinhole camera model [18]. A point located in the image is denoted with $\boldsymbol{m}=[u,v]^T$ and a point located in the 3D space is denoted with $\boldsymbol{M}=[X,Y,Z]^T$ [19]. A projected point $\boldsymbol{m}$ is determined with an optical ray that is reflected from a point $\boldsymbol{M}$ of the observed scene and travels through the optical centre $C$ and then hits the image plane. Points $\boldsymbol{m}$ and $\boldsymbol{M}$ can be written in homogeneous coordinates as $\boldsymbol{m}=[u,v,1]^T$ and $\boldsymbol{M}=[X,Y,Z,1]^T$. The pinhole camera model defines transformation between the 3D point $\boldsymbol{M}$ and its projection on the image-point $\boldsymbol{m}$:

$$s\boldsymbol{m}=\boldsymbol{A}[\boldsymbol{R}\ \boldsymbol{T}]\boldsymbol{M} \tag{1}$$

where $s$ is the scale factor. Rotation matrix $\boldsymbol{R}$ and translational vector $\boldsymbol{T}$ represent the extrinsic parameters which describe the transformation between the world coordinate system and the camera coordinate system. $\boldsymbol{A}$ is a matrix of camera intrinsic parameters and it is defined as:

$$\boldsymbol{A}=\begin{bmatrix}\alpha & \gamma & u_0\\ 0 & \beta & v_0\\ 0 & 0 & 1\end{bmatrix} \tag{2}$$

where $(u_0v_0)$ is the principal point, $\alpha=fs_x$ and $\beta=fs_y$ represent the focal length expressed in pixels according to the coordinate axes $u$ and $v$. $f$ is the focal length expressed in millimetres, $s_x$ and $s_y$ are scale factors according to the coordinate axes $u$ and $v$, which determine number of pixels per millimetre. Skew parameter $\gamma$ represents the distortion of a pixel.

From a plurality of modes for the estimation of the planar homography between the calibration target (the plane is covered with a chessboard pattern) and its image, a method that is based on the criterion of maximum likelihood [19] was chosen. In this case the calibration parameters are estimated analytically in the first step. In the second step this result is optimized by using the non-linear optimization technique based on the maximum likelihood criterion.

It may be assumed for the calibration target that $Z$ component of the point is always equal to zero. Therefore the point $\boldsymbol{M}$ can be written as $\boldsymbol{M}=[X\,Y]^T$ or in homogeneous coordinates as $\boldsymbol{M}=[X\,Y\,1]^T$. Consequently the transformation between points $\boldsymbol{m}$ and $\boldsymbol{M}$ can be written as:

$$s\boldsymbol{m}=\boldsymbol{H}\boldsymbol{M} \tag{3}$$

where $\boldsymbol{H}=\boldsymbol{A}[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{T}]$ is homography, which is defined up to scale factor λ. Further the next relation can be written:

$$[\boldsymbol{h}_1\boldsymbol{h}_2\boldsymbol{h}_3]=\lambda\boldsymbol{A}[\boldsymbol{r}_1\boldsymbol{r}_2\boldsymbol{T}], \tag{4}$$

where $r_i$ is $i$-th column of the matrix $\boldsymbol{R}$ and $h_i$ is $i$-th column of the matrix $\boldsymbol{H}$. The maximum likelihood estimation of homography H is obtained by minimizing the expression:

$$\sum\nolimits_i=(\boldsymbol{m}_i-\hat{\boldsymbol{m}}_i)^T\boldsymbol{\Lambda}_{\boldsymbol{m}_i}^{-1}(\boldsymbol{m}_i-\hat{\boldsymbol{m}}_i) \tag{5}$$

where $\hat{\boldsymbol{m}}_i=\frac{1}{\overline{\boldsymbol{h}}_3\boldsymbol{T}\boldsymbol{M}_i}\begin{bmatrix}\overline{\boldsymbol{h}}_1\boldsymbol{T}\boldsymbol{M}_i\\ \overline{\boldsymbol{h}}_2\boldsymbol{T}\boldsymbol{M}_i\end{bmatrix}$ is the point extracted from the model of the calibration target points, $\overline{\boldsymbol{h}}_i$ is $i$-th row of the matrix $\boldsymbol{H}$ and $\boldsymbol{\Lambda}_{\boldsymbol{m}_i}=\sigma^2\boldsymbol{I}$ is the covariance matrix. Eq. (5) is further written as a nonlinear minimization problem, which is based on the method of least squares $\min_H\sum_i\|\boldsymbol{m}_i-\hat{\boldsymbol{m}}_i\|^2$. To obtain the analytic solution, all rows of the matrix $\boldsymbol{H}$ are further written in matrix $x=\left[\overline{\boldsymbol{h}}_1T,\overline{\boldsymbol{h}}_2T,\overline{\boldsymbol{h}}_3T\right]^T$ and Eq. (3) is transformed in:

$$\begin{bmatrix}\boldsymbol{M}^T & 0^T & -u\boldsymbol{M}^T\\ 0^T & \boldsymbol{M}^T & -v\boldsymbol{M}^T\end{bmatrix}\boldsymbol{x}=0 \tag{6}$$

or $\boldsymbol{L}\boldsymbol{x}=0$, where matrix $\boldsymbol{L}$ has dimensions $2n\times 9$ ($n$ is the number of points). Since the matrix $\boldsymbol{x}$ is defined up to the scale factor, the solution is the right singular vector of the matrix $\boldsymbol{L}$, which is associated with the smallest singular value.

Matrix $\boldsymbol{B}=\boldsymbol{A}^{-T}\boldsymbol{A}^{-1}$ is considered as a symmetric matrix, which can be defined as 6D vector $\boldsymbol{b}=[B_{11},B_{12},B_{22},B_{23},B_{33}]^T$.

If the $i$-th column of the matrix $\boldsymbol{H}$ is written as $\boldsymbol{h}_i=[h_{i1},h_{i2},h_{i3}]^T$, then the next equation is valid:

$$\boldsymbol{h}_i^T\boldsymbol{B}\boldsymbol{h}_j=\boldsymbol{v}_{ij}^T\boldsymbol{b} \tag{7}$$

where $\boldsymbol{v}_{ij}=[\boldsymbol{h}_{i1}\boldsymbol{h}_{j1},\boldsymbol{h}_{i1}\boldsymbol{h}_{j2}+\boldsymbol{h}_{i2}\boldsymbol{h}_{j1},\boldsymbol{h}_{i2}\boldsymbol{h}_{j2},\boldsymbol{h}_{i3}\boldsymbol{h}_{j1}+\boldsymbol{h}_{i1}\boldsymbol{h}_{j3},\boldsymbol{h}_{i3}\boldsymbol{h}_{j2}+\boldsymbol{h}_{i2}\boldsymbol{h}_{j3},\boldsymbol{h}_{i3}\boldsymbol{h}_{j3}]^T$.

If it is considered that the vectors of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthonormal, the equation for the limitation of intrinsic parameters can be written:

$$\begin{bmatrix}\boldsymbol{v}_{12}^T\\ (\boldsymbol{v}_{11}-\boldsymbol{v}_{22})^T\end{bmatrix}\boldsymbol{b}=0 \tag{8}$$

If $n$ images of the calibration target are captured, then also the number of Eq. (8) is equal to $n$, which further can be combined in a new equation as:

$$\boldsymbol{V}\boldsymbol{b}=0 \tag{9}$$

where $\boldsymbol{V}$ is the matrix with dimensions $2n\times 6$. Since a unique solution $b$ is required, the number $n$ has to be $n\geq 3$. The solution to Eq. (9) is obtained by using the singular value decomposition (SVD), as the eigenvector of matrix $\boldsymbol{V}^T\boldsymbol{V}$, which belongs to the smallest eigenvalue (equivalently, the right singular vector of $\boldsymbol{V}$ associated with the smallest singular value). If the smallest eigenvalue of the matrix $\boldsymbol{V}^T\boldsymbol{V}$ is equal to zero, then the eigenvector, which belongs to this eigenvalue, is the exact solution for the equation [13]. This statement is true also when $n=3$ if the matrix $V$ is singular (one of the singular values which are computed with the SVD is equal to zero). In all other cases when there is no singular value which would be equal to zero (the rank of matrix $\boldsymbol{V}$ is equal to six), the eigenvector of matrix $\boldsymbol{V}^T\boldsymbol{V}$, which belongs to the smallest eigenvalue, represents the best solution in a sense of least squares.

When the vector $\boldsymbol{b}=[B_{11},B_{12},B_{22},B_{13},B_{23},B_{33}]^T$ is estimated, all intrinsic parameters can be computed as:

$$\begin{aligned}v_0&=\left(\frac{B_{12}B_{13}-B_{11}B_{23}}{B_{11}B_{22}-B_{12}^2}\right)\lambda=B_{33}-\frac{(B_{13}^2+v_0(B_{12}B_{13}-B_{11}B_{23}))}{B_{11}}\\ \alpha&=\sqrt{\frac{\lambda}{B_{11}}},\beta=\sqrt{\frac{\lambda B_{11}}{B_{11}B_{22}-B_{12}^2}}\\ \gamma&=\frac{-B_{12}\alpha^2\beta}{\lambda},u_0=\frac{\gamma v_0}{\beta}-\frac{\beta_{13}\alpha^2}{\lambda}\end{aligned} \tag{10}$$

When the matrix of the intrinsic parameters $\boldsymbol{A}$ is determined, extrinsic parameters considering Eq. (4) can be calculated (where orthonormality of $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ is taken into account):

$$\begin{aligned}\boldsymbol{r}_1&=\lambda\boldsymbol{A}^{-1}\boldsymbol{h}_1\\ \boldsymbol{r}_2&=\lambda\boldsymbol{A}^{-1}\boldsymbol{h}_2\\ \boldsymbol{r}_3&=\boldsymbol{r}_1\times\boldsymbol{r}_2\\ \boldsymbol{T}&=\lambda\boldsymbol{A}^{-1}\boldsymbol{h}_3\end{aligned} \tag{11}$$

where $\lambda=1/\|\boldsymbol{A}^{-1}\boldsymbol{h}_1\|=1/\|\boldsymbol{A}^{-1}\boldsymbol{h}_2\|$. In the calibration process the radial distortion described by the following model was also taken into account:

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4) \begin{bmatrix} x \\ y \end{bmatrix} \quad (12)$$

where $[x' \; y']$ are coordinates of the distorted point, $[x \; y]$ are coordinates of the undistorted point and parameter $r$ is equal to $r = x^2 + y^2$. The radial distortion coefficients are denoted by $k_i$.

The calibration procedure can be summed up as follows: 1. capturing images of the calibration object from different angles (to achieve good results of the camera calibration at least twenty image shots are required), 2. detecting the corners on the chessboard for all captured images, 3. estimating homographies for all captured images by using Eq. (6), 4. considering all homographies in Eq. (9), which is used for estimation of vector $b$ with SVD, 5. estimating five intrinsic parameters considering vector $\boldsymbol{b}$ and Eq. (10), 6. estimating rotation matrices and translation vectors (with Eq. (11), for all images) which represent extrinsic parameters, 7. optimizing all calibration parameters and radial distortion coefficients, using the Levenberg–Marquardt algorithm:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \| \boldsymbol{m}_{ij} - \boldsymbol{m}'(\boldsymbol{A}, k_1, k_2, \boldsymbol{R}_i, \boldsymbol{T}_i, \boldsymbol{M}_j) \|^2 \quad (13)$$

where point $\boldsymbol{m}'(\boldsymbol{A}, k_1, k_2, \boldsymbol{R}_i, \boldsymbol{T}_i, \boldsymbol{M}_j)$ is a projection of a point $\boldsymbol{M}_j$ in the image $i$ according to Eq. (3).

In the algorithm of visual odometry, it is necessary to know the rigid transformation between ground (plane) C.S. and camera C.S.: $\boldsymbol{g} = (\boldsymbol{R}_{CP}, \boldsymbol{T}_{CP})$. This rigid movement can be calculated by considering the rigid transformation between the chessboard and the camera $\boldsymbol{g} = (\boldsymbol{R}_{CT}, \boldsymbol{T}_{CT})$ (Fig. 1) and the assumption that yaw = 0 between ground and camera C.S.

### 2.2. Optical flow

Optical flow is a way of describing the seeming movement of surface, objects and edges that occurs due to the relative motion between the observer and the scene. The purpose of the optical flow methods (Kanade and Lucas [20], Horn–Schunck, etc.) is to determine the movement between two consecutive images which are captured at time $t$ and $t + \Delta t$. These optical flow methods are considered as differential methods, since they are based on local Taylor approximation of image signal.

In the implementation of the visual odometry algorithm, Kanade–Lucas-Tomasi (KLT) feature tracker was used, which was introduced in three articles, namely Lucas and Kanade [20], Tomasi and Kanade [21] and Shi and Tomasi [22]. The problem solved by the KLT algorithm can be defined as searching moves between two consecutive images, which are denoted by $I$ and $J$. These two images are grayscale images with size $N_x \times N_y$ pixels. Let $I(x,y)$ represent the intensity of a grayscale image $I$ in the point $(x,y)$. If point $\boldsymbol{u} = (u_x, u_y)$ is known in the image $I$, then the aim of the algorithm is to find the point $\boldsymbol{v}$ in the image $J$, where the values of $I(\boldsymbol{u})$ and $J(\boldsymbol{v})$ are similar ($\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{d} = (u_x + d_x, u_y + d_y)$). In the point u the optical flow is equal to $\boldsymbol{d} = [d_x \; d_y]^T$. Current optical flow is evaluated with the estimator $\varepsilon(\boldsymbol{d})$, which is defined as:

$$\begin{aligned} \varepsilon(\boldsymbol{d}) &= \varepsilon(d_x, d_y) \\ &= \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} (I(x,y) - J(x+d_x, y+d_y))^2, \end{aligned} \quad (14)$$

where $\omega_x$ in $\omega_y$ are parameters which determine the size of the integration window: $(2\omega_x + 1) \times (2\omega_y + 1)$ (Fig. 2).

Ordinary algorithm KLT works well only when the movements of the pixels are small. For larger movements the pyramidal implementation of this algorithm is needed [23,24]. In the pyramidal implementation of algorithm KLT, the optical flow is estimated on an image pyramid $I^0 \rightarrow I^1 \rightarrow I^2 \rightarrow I^3 \rightarrow I^{L_m} \ldots (L_m: 2 \sim 4)$, where the optical flow is computed first at the highest level and then at lower levels up to the level zero. On each level a preliminary estimation of the movement from the higher level is considered. The reason to start the calculation on the highest level is that the movement is the smallest at a minimum resolution. The optical flow on the individual level is calculated by using the gradient matrix:

$$\boldsymbol{G} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (15)$$

and image mismatch vector:

$$\boldsymbol{b} = \sum_{x=p_x-\omega_x}^{p_x+\omega_x} \sum_{y=p_y-\omega_y}^{p_y+\omega_y} \begin{bmatrix} \delta I & I_x \\ \delta I & I_y \end{bmatrix} \quad (16)$$

as:

$$\boldsymbol{d}^L = \boldsymbol{G}^{-1} \boldsymbol{b} \quad (17)$$

where $I_x$ in $I_y$ are the partial derivatives of the image $I$ and $\delta I(x,y) = A(x,y) - B(x,y)$ is the difference between two sub-images. The final optical flow is obtained as:

$$\boldsymbol{d} = \sum_{L=0}^{L_m} 2^L \boldsymbol{d}^L \quad (18)$$

where $\boldsymbol{d}^L$ is the estimated displacement on the $L$-th level.

### 2.3. Algorithm RANSAC

The RANSAC (RANdom SAmple Consensus) [18,25,26], is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains a lot of outliers. In statistics, an outlier is an observation point which is distant from other observations due to extreme values of the noise or erroneous measurements. By using algorithm RANSAC, parameters of a model which fits given data can be estimated robustly even when data contain a lot of elements (up to 50%) which greatly differ from the exact values. Roughly speaking, the algorithm works in two steps that are iteratively repeated. In the first step, a set with a minimum number of data elements $m$ (i.e. $m = 3$ for determining the rigid motion) for the estimation of parameters of a model is randomly selected from the data and then a model is fitted to this set of hypothetical inliers. In the second step, when the parameters are estimated, it must be checked how many of the remaining data elements suit to the estimated model. A data element will be considered as an outlier if it does not fit the fitting model within some error threshold that defines the maximum deviation attributable to the effect of noise. If the number of data elements that fit the estimated model well is larger than a certain threshold, then the model is accepted and the fitting data elements are stored,

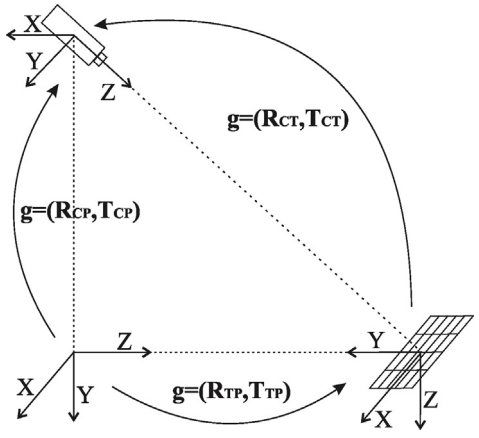


**Fig. 1.** Transformations between ground C.S., chessboard C.S. and camera C.S.

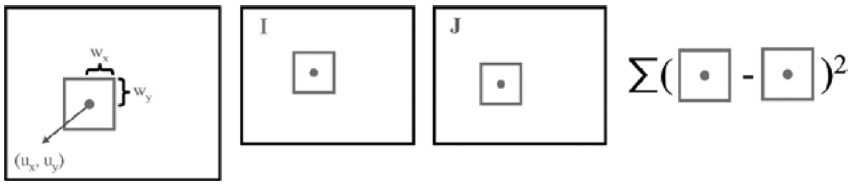


**Fig. 2.** Estimation of the error in searching the shift between images I and J.

otherwise the procedure must be repeated. The probability that the selected model with currently estimated parameters optimally explains given data increases with the number of iterations $K$. This parameter of the algorithm RANSAC can be determined by the following equation:

$$K = \frac{\log(1-p)}{\log(1-w^m)}, \tag{19}$$

where $p$ is the probability that a set of $m$ data elements which does not contain outliers is found (i.e. the probability that the algorithm produces a useful result, e.g. if it is required that algorithm RANSAC is successful in 99% of all the cases, then $p$ must be equal to 0.99) and $w^m$ is the probability that all of $m$ selected data elements are inliers (e.g. if data consist of 60% inliers and 40% outliers then $w^m = 0.6^m$). In practice only a rough estimation of the proportion of inliers in data is needed.

### 2.4. Determining the rigid motion

In the case of a rigid motion [27] the movement of individual points does not need to be described, but only the movement of one point which represents the movement of the entire set of points can be given. In the case of a rigid motion the distance and also the orientation between points are preserved. This means that the norm of the vectors and their vector product remain the same in a rigid motion.

In the visual odometry algorithm a key step is the calculation of a rigid transformation between the two sets of points obtained by using the KLT feature tracker. This transformation between the two sets of points $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{p}_i\}$; $i = 1{:}N$ in the two-dimensional space is determined with the rotation matrix $\boldsymbol{R}_{2\times2}$ and translation vector $\boldsymbol{T}_{2\times1}$, which must minimize the following cost function:

$$f(\boldsymbol{p}) = \frac{1}{N}\sum_{i=1}^{N} \| \boldsymbol{x}_i - \boldsymbol{R}(\upsilon)\cdot\boldsymbol{p}_i - \boldsymbol{T} \|^2, \tag{20}$$

where $\nu$ is the angle of rotation. If the geometric centre of all points from the first set is denoted by $\boldsymbol{c} = 1/N\sum_{i=1}^{N}\boldsymbol{x}_i$ and the geometric centre of all points from the second set is denoted by $\boldsymbol{c}' = 1/N\sum_{i=1}^{N}\boldsymbol{p}_i$, then the translation is defined as:

$$\boldsymbol{T} = \boldsymbol{c} - \boldsymbol{R}\boldsymbol{c}' \tag{21}$$

In order to determine the translation, firstly the estimation of rotation between the two sets of points $\hat{\boldsymbol{p}}_i = \boldsymbol{p}_i - \boldsymbol{c}'$ and $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i - \boldsymbol{c}$ must be solved. The points $\hat{\boldsymbol{p}}_i$ and $\hat{\boldsymbol{x}}_i$ have the geometric centre in the origin. The rotation matrix can be obtained by using the orthogonal Procrustes algorithm [28,29]. Arun et al. [30] presented a way of solving the Procrustes problem by using the singular value decomposition—SVD of correlation matrix in their work. The correlation matrix $\boldsymbol{C}$ of size $2\times2$ (or $3\times3$ in 3D space) can be written as:

$$\boldsymbol{C} = \sum_{i=1}^{N}\hat{\boldsymbol{p}}_i\hat{\boldsymbol{x}}_i T = U\sum V^T \tag{22}$$

where $\boldsymbol{U}\Sigma\boldsymbol{V}^T$ represents a singular value decomposition. The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices and $\Sigma$ is a diagonal matrix with nonnegative elements. The rotation matrix is calculated from the resulting matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ as:

$$\boldsymbol{R} = \boldsymbol{V}\boldsymbol{U}^T \tag{23}$$

The size of matrix $\boldsymbol{R}$ determinant must always be checked, as it must be equal to +1 so that the rotation matrix is correctly determined. When the rotation matrix is known, the translational vector can be calculated by using Eq. (21).

### 2.5. Monocular visual odometry

In the implementation of monocular visual odometry the following assumptions were taken into account: the camera is pointing towards a flat surface and a rigid transformation between the camera C.S. and the ground C.S. is the same at all times. Such assumptions are often true for autonomous vehicles and robots that move on a flat surface and also for pedestrians who walk at a normal speed on a flat surface, if raising and lowering of the body during the walk are disregarded.

As it has been already mentioned, the visual odometry algorithm consists of four basic components which were described in the previous subsections. The first component is the calibration algorithm, which is not performed as a part of the visual odometry, but it is executed only once as an initialization procedure in which the transformation between the camera C.S. and the ground C.S. is determined. The KLT algorithm, the algorithm for determining the

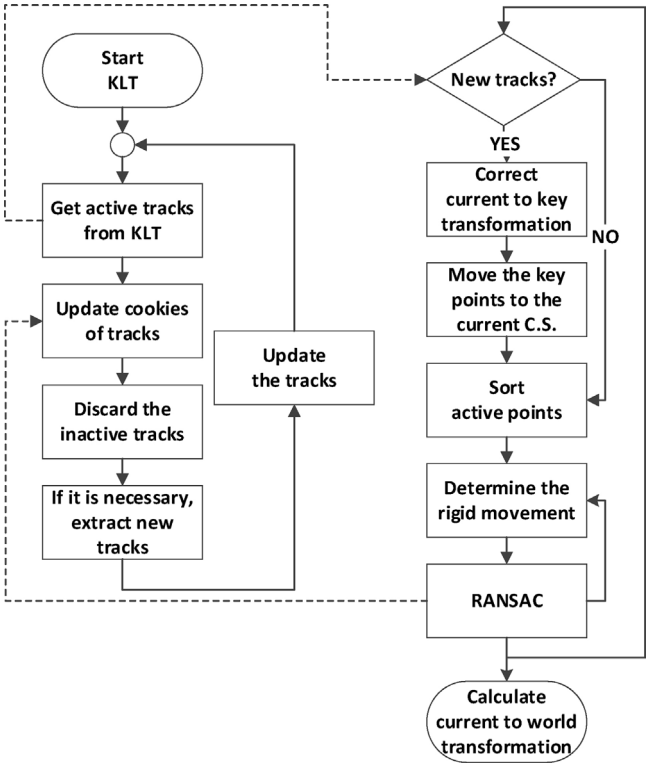


**Fig. 3.** The flowchart of the visual odometry algorithm.

rigid transformation and RANSAC algorithm are three basic components of visual odometry, which must be performed in real time on the selected hardware—on the smartphone. Since modern smartphones have built-in multi-core processors, the visual odometry algorithm has been implemented in a way that compute-intensive parts are performed on separate threads which can be executed on individual processor cores. This means that the individual components of visual odometry can be performed simultaneously. Fig. 3 shows a flowchart of the visual odometry algorithm, which is divided into two parts i.e. two threads. The algorithm KLT or determination of traces is assigned to one thread (left half of the diagram), the algorithm for determining the rigid movement and RANSAC algorithm are assigned to another thread (right half of the diagram) with the purpose of making the two parts of the algorithm approximately equal in the sense of computational intensity. Since KLT algorithm works better for small movements of feature points, it is important that the algorithm runs fast and so more frames per second are processed. All components of the visual odometry algorithm running in separate threads must appear as a whole meaning they must communicate with each other. Thus, the threads exchange data, as it is shown in Fig. 3 with dashed lines. The first thread, which is responsible for the determination of traces of feature points, sends information about the currently active traces to the second thread, and the second thread sends the information to the first thread about traces which were used in determining the rigid transformation and traces which represent outliers according to the selected model (i.e. determined with RANSAC). This information is stored in the so-called cookie of the individual trace. In the KLT thread these cookies are refreshed every time when a new image is captured and then if necessary, according to the provided information, deletion and adding of new traces is carried out.

Below is described a part of the visual odometry algorithm, which is represented by the right half of the diagram in Fig. 3. This part of the algorithm mostly consists of the algorithm for determining a rigid transformation and the RANSAC algorithm by which a rigid transformation from current C.S. to the world C.S. is obtained.

In the visual odometry algorithm, detected movements are calculated in a 2D space, and then the final result is transformed into a 3D space for the purpose of calculating a rigid transformation from the camera C.S. to the world C.S. For determining the rigid movement in 2D, three coordinate systems are important, namely world C.S. (W), key C.S. (K) and the current (plane) C.S. (P), which are aligned at the start-up of visual odometry (Fig. 4). As can be seen from the figure, these coordinate systems lie in the $XZ$ plane of ground C.S., and they have a slightly different orientation in relation to the 3D ground C.S. The axis $Y_{2D}$ of the current C.S. (2D) coincides with the axis—$X$ and the axis $X_{2D}$ coincides with the axis $Z$ of the ground C.S.

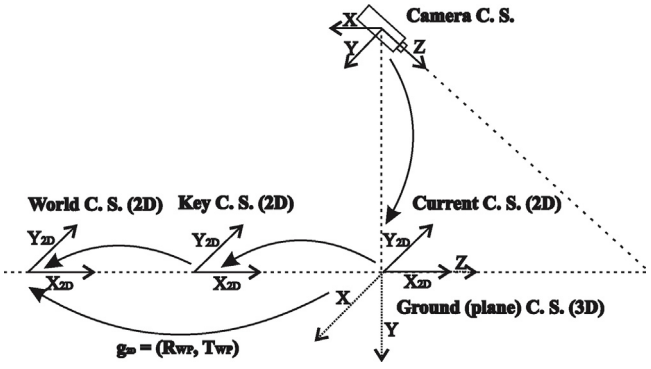


**Fig. 4.** Rigid movement in 2D from the current C.S. to the world C.S. is obtained by considering regular calculations for the rigid transformation from the current C.S. to the key C.S.

With each new image frame the new position of active traces in the current C.S. is obtained. World C.S., which is determined by the first frame, is not moveable and it presents a reference C.S. in which the final result of visual odometry, i.e. current position of the camera, is expressed. The key C.S. is determined by the first frame, and it moves in the current C.S. only when new traces are added (when the number of all traces falls below a certain threshold, the new traces must be extracted). In this case the rigid transformation from the key C.S. to the world C.S. $\boldsymbol{g}_{\mathrm{WK}}^{2\mathrm{D}}$ is refreshed, i.e. it becomes equal to the transformation $\boldsymbol{g}_{\mathrm{WP}}^{2\mathrm{D}}$ and all key points which belong to the active traces are moved to the current C.S.

From the pixels that are defined by the active traces, the points on the normalized image plane can be calculated by considering the intrinsic parameters and radial distortion. These points can then be transformed into the 2D space of the ground (to the floor) so that the coordinates are expressed in the current C.S. Before the points are used for the calculation of the rigid transformation, it must be checked which detected feature points lie on the surface (i.e. denoted as "sort active points" in Fig. 3). This can be done by describing the pixels of the active points in the ground C.S., and then $Y$ component of each point is verified if it is positive (i.e. the ray that passes through this point on the image plane penetrates the floor).

Point $\boldsymbol{x} = (x, y)$, determined by the ray which travels from the focal point towards the ground and passes through the normalized image plane, can be expressed in homogeneous coordinates as $\boldsymbol{x} = (x, y, 1)$. To get the point on the ground $(\lambda x, \lambda y, \lambda)$ in camera C.S., which is determined by a ray passing through $\boldsymbol{x} = (x,y,1)$, scale factor $\lambda$ must be calculated. This factor determines the distance to the point in the Z direction of the camera coordinate system. The problem of searching the scale factor is solved in such a way that the point $(x, y, 1)$ is first rotated using the rotation matrix $\boldsymbol{R}_{\mathrm{PC}}$, with the purpose of getting the point $(X, Y, Z)$ in the coordinate system which has the axis aligned with the ground C.S. If it is assumed that this point lies on the ground, then its $Y$ coordinate must be equal to the height of the camera from the ground. Thus, the point in the plane of the current C.S. (2D) is determined as $\boldsymbol{X}_{\boldsymbol{P}}^{2\mathrm{D}} = (Z \cdot t, -X \cdot t)$, where $t$ = camera height/Y (this ratio determines that the point lies on the floor). The height of the camera is obtained from the known translational vector $\boldsymbol{T}_{\mathrm{PC}}$, which is calculated in the calibration phase. When the corresponding points are mapped from the image plane to the ground in the current C.S. (2D space), the model of rigid movement from the current C.S. to the key C.S. $\boldsymbol{g}_{\mathrm{KP}}^{2\mathrm{D}}$ can be calculated. This rigid transformation represents the core of the entire visual odometry. All 2D rigid movements can be described by means of the translation $\boldsymbol{T} = [X, Y]^T$, and an angle of rotation (yaw), which determines the rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} \cos(\mathrm{yaw}) & -\sin(\mathrm{yaw}) \\ \sin(\mathrm{yaw}) & \cos(\mathrm{yaw}) \end{bmatrix}. \tag{24}$$

The rigid transformation $\boldsymbol{g}_{\mathrm{PK}}^{2\mathrm{D}}$ between key points and current points on the plane is calculated simultaneously using Procrustes analysis. In doing so, the suitability of the model is checked with the algorithm RANSAC. This algorithm checks which model calculated by randomly selected three pairs of points corresponds to the maximum number of remaining pairs of points. The pairs of points consist of the currently active points on the floor (Fig. 5, step 1) and corresponding key points. The resulting model (Fig. 5, step 2) is used in conjunction with a rigid transformation between the ground C. S and the camera C.S. and in this way the current points of the model in the normalized image coordinates from the key points are computed (Fig. 5, step 3). These points are then

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

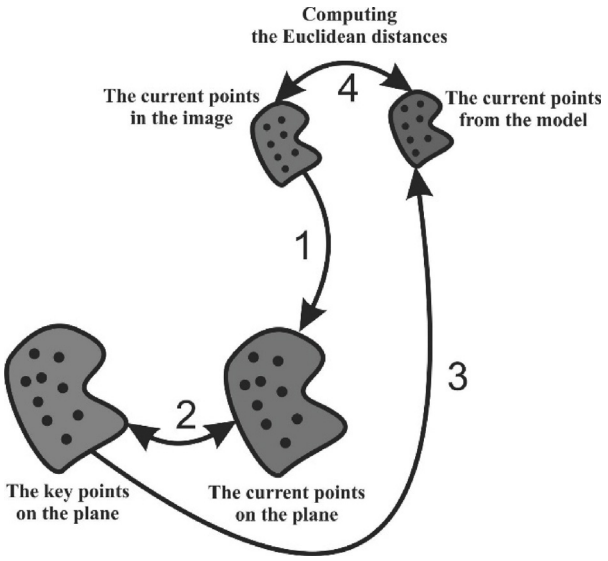


**Fig. 5.** Checking the suitability of the rigid motion model in RANSAC algorithm.

compared to the real current points by calculating the Euclidean distances (Fig. 5, step 4). When the distance between the points is known, the algorithm can determine how many points are described well by the model. The distance between points is expressed in pixels. The points and their traces which do not correspond to the resulting model are added on the list for deletion. Each trace is deleted if it is not used in the last X (the parameter that you set) images. When the number of active traces falls below a certain threshold, new traces are generated by means of KLT algorithm. With each new image frame the active traces of the feature points are updated and transformation from the current C.S. to the world C.S. $\boldsymbol{g}_{\mathrm{WP}}^{\mathrm{2D}}$, which presents a part of the final transformation from the camera C.S. to the world C.S., is calculated as $\boldsymbol{g}_{\mathrm{WP}}^{\mathrm{2D}} = (\boldsymbol{R}_{\mathrm{WP}}, \boldsymbol{T}_{\mathrm{WP}}) = \boldsymbol{g}_{\mathrm{WK}}^{\mathrm{2D}} \boldsymbol{g}_{\mathrm{KP}}^{\mathrm{2D}}$.

In order to calculate the transformation from the world C.S. to the camera C.S., first the rigid movement $\boldsymbol{g}_{\mathrm{WP}}$ in 3D from the current C.S. to the world C.S. (the orientation of C.S. in 3D is the same as in the ground C.S., which is marked with dotted line in Fig. 4) must be calculated: $\boldsymbol{T}_{\mathrm{WP}}^{\mathrm{3D}} = [-T_Y, 0, T_X], \boldsymbol{R}_{\mathrm{WP}}^{\mathrm{3D}} = \mathrm{roty}(-\mathrm{yaw})$ (minus sign is obtained, since the axis of rotation have the opposite direction in 3D space in relation to 2D space).

When the transformations $\boldsymbol{g}_{\mathrm{PW}} = \boldsymbol{g}_{\mathrm{WP}}^{-1}$ and $\boldsymbol{g}_{\mathrm{CP}}$ (in 3D), which are shown in Fig. 6, are determined, the rigid movement from the world C.S. to the camera C.S. can be calculated as: $\boldsymbol{g}_{\mathrm{CW}} = (\boldsymbol{R}_{\mathrm{CW}}, \boldsymbol{T}_{\mathrm{CW}}) = \boldsymbol{g}_{\mathrm{CP}} \boldsymbol{g}_{\mathrm{PW}}$ ($\boldsymbol{g}_{\mathrm{CP}}$ is calculated in the calibration phase). If the inverse of transformation is found: $\boldsymbol{g}_{\mathrm{CW}}^{-1} = \boldsymbol{g}_{\mathrm{WC}} = (\boldsymbol{R}_{\mathrm{WC}}, \boldsymbol{T}_{\mathrm{WC}})$, then the translation vector $\boldsymbol{T}_{\mathrm{WC}}$ defines where the camera is positioned in relation to the world C.S. at any given moment. Thus this vector represents the output of the monocular visual odometry.

## 3. Inertial navigation system

PDR (Pedestrian Dead Reckoning) is a technique that is used in inertial navigation systems (INS) for the purpose of determining the movement of pedestrians (smartphone users) according to the initial position. The movement is given by the number of taken steps and directions in which these steps are taken. The Dead Reckoning technique based on inertial sensors is increasingly present as a complementary system in different localization systems of autonomous mobile systems and robots. The advantages of this approach are small dimensions of sensors and their low cost and above all this approach significantly contributes to improving the accuracy of localization systems. Inertial navigation system is based on three sensors, namely accelerometer, gyroscope and magnetometer. With the signal processing of these sensors, a digital compass for the purpose of determining the orientation and a pedometer for the purpose of calculating the walking distance can be implemented. These two components represent the basis of the inertial navigation system. Pedometer consists of the step counter and the algorithm for the step length estimation. For the autonomous mobile systems, where the usage of a pedometer is not possible for the purpose of calculating the length of the travelled path, double integration of acceleration can be used. However, in this case due to tilt error and noise contained in the accelerometer signal, the error is rising rapidly, and it must be compensated by considering another localization technique, e.g. visual odometry. In case that a global localization technique, such as localization by using WiFi or Bluetooth signals is available, it is most appropriate to use it for the purpose of limiting the increase of the total error caused by dead reckoning technique [31].

### 3.1. Digital compass

One of the key components of inertial navigation system is a digital compass, whose mission is to determine the absolute orientation of the device (and the user) as best as possible according to the world coordinate system (Fig. 7). Digital compass requires essentially two sensors, namely accelerometer, and magnetometer to operate. The accelerometer can determine the direction of gravity (which is parallel to the *Z*-axis of the world C.S.), and the magnetometer can find the direction of the magnetic north. According to these two directions the *X*-axis (Fig. 7) can be calculated with the cross product. For more accurate and stable operation of the digital compass a gyroscope is needed that accurately measures angular velocity (relative rotation) and it is

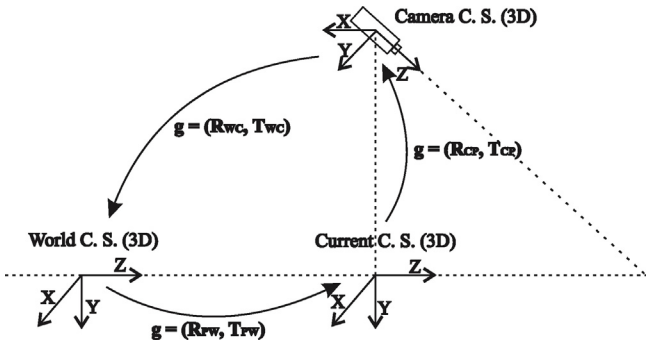


**Fig. 6.** Rigid transformation from the world C.S. to the camera C.S.

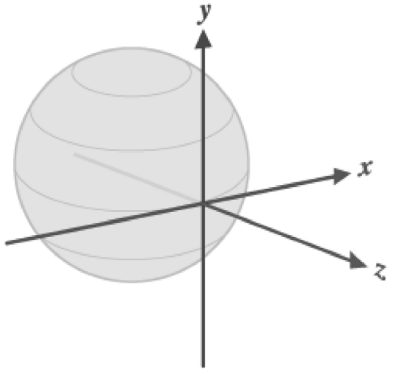


**Fig. 7.** World coordinate system used by the Rotation vector [33].

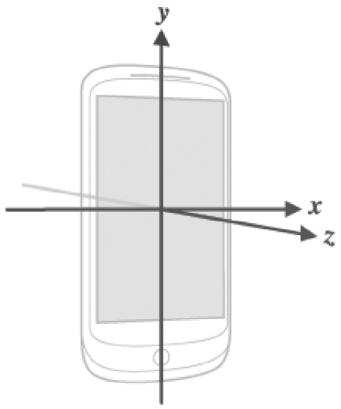


**Fig. 8.** Smartphone coordinate system [33].

also much more responsive (higher sampling rate) than magnetometer. The gyroscope can successfully eliminate false rotations returned by the magnetometer in the presence of a magnetic interference. All mentioned sensors measure its values relative to the device coordinate system (Fig. 8). The inertial sensors are rarely used individually primarily due to noise, drift and bias which are present in the measurements of the sensors. All mentioned sensors' errors can be largely eliminated by the fusion of all three sensors. Thus, the Android operating system already includes multiple virtual sensors or software-based sensors that combine measurements from the hardware-based sensors. One of the software-based sensors is the Android Rotation vector, which combines the accelerometer, gyroscope and magnetometer by using the Kalman filter [32] in one orientation sensor. The output of this sensor is in the form of rotational vector (or quaternion), which describes the absolute orientation of the smartphone in relation to the world C.S. (Fig. 7).

It also depends on the MEMS chip (Microelectromechanical systems), which is built into the smartphone, how effective the sensor fusion, which is included in the Android OS by default, is. In the test of a digital compass which is based only on the Rotation vector, it has been found that the gyroscope has too little role in the default sensor fusion, since magnetic disturbance in the surroundings has big impact on the measurements of rotation. The measurement deviation is especially a big problem when magnetic disturbance is spread over a large part of the route. In this case the direction of orientation slowly drifts toward wrong value returned by magnetometer. Therefore, the second-level fusion of the measurements of the Rotation vector and the calibrated gyroscope, which maintains low drift errors, has been implemented. The calibrated gyroscope is very responsive and precise at measuring the rate of rotation. There is only one problem with gyroscopes, namely they provide only relative rotations. The aim of the proposed fusion is to increase the impact of the gyroscope to get an even more precise virtual orientation sensor. The diagram of the sensor fusion is shown in Fig. 9, where the right window summarizes the second-level fusion.

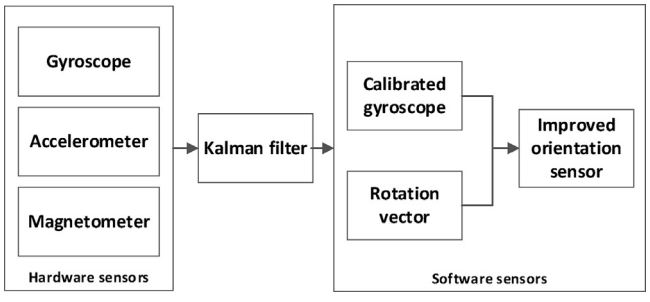


**Fig. 9.** Second-level fusion of measurements of Rotation vector and calibrated gyroscope.

As the gyroscope measures the rate of rotation (angular velocity), the sensor's output must be integrated in order to get a relative rotation. Then, the angle obtained from the gyroscope can be written in the form of a quaternion, while the output from the rotation sensor already has that form. A quaternion is determined by the axis of rotation and the angle of rotation about this axis. The most appropriate way to combine two quaternions (interpolation) is the Quaternion SLERP (Spherical Linear Interpolation) [34] method, in which the contribution of each quaternion is determined by the weights. In this case, the impact of the gyroscope should be increased, while the influence of the Rotation vector can be decreased to the limit which is necessary for eliminating the drift of the gyroscope. Weight which is close to one (e.g. 0.998) has been assigned to the gyroscope. The Rotation vector is used to determine the initial absolute orientation of the smartphone according to the world coordinate system. Therefore, both quaternions are initialized to the same value at the beginning. This means that their dot product is equal to one. If one of the vectors jumps instantaneously, then the dot product is also changed (it is decreased). In case that the dot product falls below a certain threshold (e.g. 0.8) then rotation is determined only by the gyroscope. This could happen in case of magnetic disturbance when the magnetometer returns false orientation of the device. If a device is exposed to vibration the incorrect values can be returned also from the gyroscope. Therefore, in case that the dot product of both vectors is below a certain threshold for a long period of time (both parameters are set), it is necessary to initialize again both rotations to the current value of the Rotation vector. The result of the fusion, which is expressed by a quaternion, is converted into a rotation matrix, and Euler's angles [35]. Only rotation around the *Z*-axis (world C.S.), i.e. yaw is important for the purpose of the digital compass. The methods for converting quaternions to the rotation matrix and Euler angles, which are by default included in the Android SDK, were used.

### 3.2. Pedometer

In the inertial navigation system for pedestrians a double integration of acceleration can be avoided by using the pedometer in determining the travelled path length. In this way, the travelled path length can be calculated more accurately at the known step length since the accumulated error (due to inaccurate stride length) is not rising as fast as in the double integration. Pedometer is a step counter which in most cases is based solely on the accelerometer, but it can use also other sensors for its operation in order to increase the reliability of the step detection. In this paper, the described implementation of the pedometer considers, in addition to the accelerometer, also the magnetometer and the gyroscope, which enable the calculation of the acceleration only in the vertical direction, regardless of the orientation of the smartphone. The measurements of the vertical acceleration (along the *Z*-axis of the world C.S.) describe the body movement during walking in the best manner or they contain most information from which it is possible to determine whether the user takes another step. Fig. 10 shows the key elements of the algorithm for the step detection in the form of a diagram.

The vertical acceleration is obtained by considering the rotation matrix $\boldsymbol{R}_{WS}$ (with size $3 \times 3$), which is determined by the improved virtual rotation sensor. The latter was described in the subsection of the digital compass, where it is used for determining the orientation of the device. Since the matrix $\boldsymbol{R}_{WS}$ describes the transformation between the world C.S. (W), which is shown in Fig. 7, and smartphone (S) C.S., which is shown in Fig. 8, the acceleration in the direction of Z-axis of the world C.S. (Fig. 8) can

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

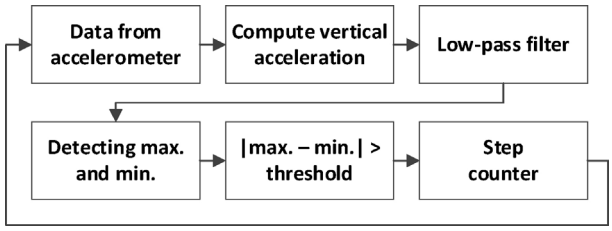


**Fig. 10.** Implementation of pedometer.

be calculated as:

$$\boldsymbol{a}_z = [\boldsymbol{R}_{\mathrm{WS}}(3,1)\boldsymbol{R}_{\mathrm{WS}}(3,2)\boldsymbol{R}_{\mathrm{WS}}(3,3)]\cdot[a_x a_y a_z]^T, \tag{25}$$

where $a_x$, $a_y$ and $a_z$ are accelerations in the direction of the axes of the smartphone C.S. The form of the signal of the vertical acceleration depends on the position of the smartphone (in hand, mounted on a body, etc.) and the walking mode of the user. However, regardless of these effects it is desired that the signal shape for further processing (for detecting local extremes) is as close as possible to the sinusoidal signal. In addition to the white noise which is incorporated in the accelerometer signal by default, high-frequency disturbances due to rapid oscillations of the device during walking (Fig. 11, above) are also presented. With the purpose of eliminating high-frequency disturbances, a low-pass filter with a cut-off frequency of 10 Hz was used, and in this way a signal that is suitable for detecting local maxima and minima (Fig. 11, below) was obtained. As can be seen in Fig. 11 below, the filtered signal is locally monotonically increasing and decreasing. So the local extrema are obtained by checking if there has been a change in the sign of the derivative of the signal. The locally maximum values are detected by checking if the signal has passed from the part of monotonic increasing to the part of monotonic decreasing. The locally minimum values are detected by checking if the signal is passed from the part of monotonic decreasing to the part of monotonic increasing.

The individual steps are detected in the area of signal where acceleration increases from minimum to maximum value. In the process it is checked if the absolute value of the difference between the maximum and the minimum value is greater than the selected threshold:

$$|a_{\max} - a_{\min}| > \text{threshold} \tag{26}$$

In case that this condition is true, the algorithm registers that a new step is detected. Scientific writings also describe other methods for the step detection, e.g. the ZUPT (Zero Velocity Update) method [36], and methods based on frequency analysis of the accelerometer signal. In the described algorithm for the step detection the information (i.e. number of steps per second) obtained by frequency analysis (FFT—Fast Fourier Transformation) is not necessary since it does not contribute to the reliability of detecting steps and it is quite compute-intensive.

### 3.3. Estimation of stride length

In order to use the pedometer as part of the inertial navigation system, it is necessary to know the length of each step. The most simple way of determining the step length is to assume that it is equal to an average value which is the same all the time. Leppakoski et al. [31] have suggested that the stride length varies linearly relative to the average step intervals (the higher step frequency means the longer step). For described pedometer a model based approach that describes the stride length by the maximum and minimum values of the vertical acceleration at current detected step was used [35]. The stride length defines the equation:

$$L = K\sqrt[4]{a_{\max} - a_{\min}} \tag{27}$$

where $a_{\max}$ is the maximum vertical acceleration and $a_{\min}$ is the minimum vertical acceleration. $K$ is a constant value which is different for each pedometer user. It can be calculated at startup by using the visual odometry. So, if the pedometer detects steps and each step length (which must be within the expected interval, e.g. between 0.3 m and 1 m) is provided by visual odometry, then the conditions are met so that the constant $K$ can be calculated for each

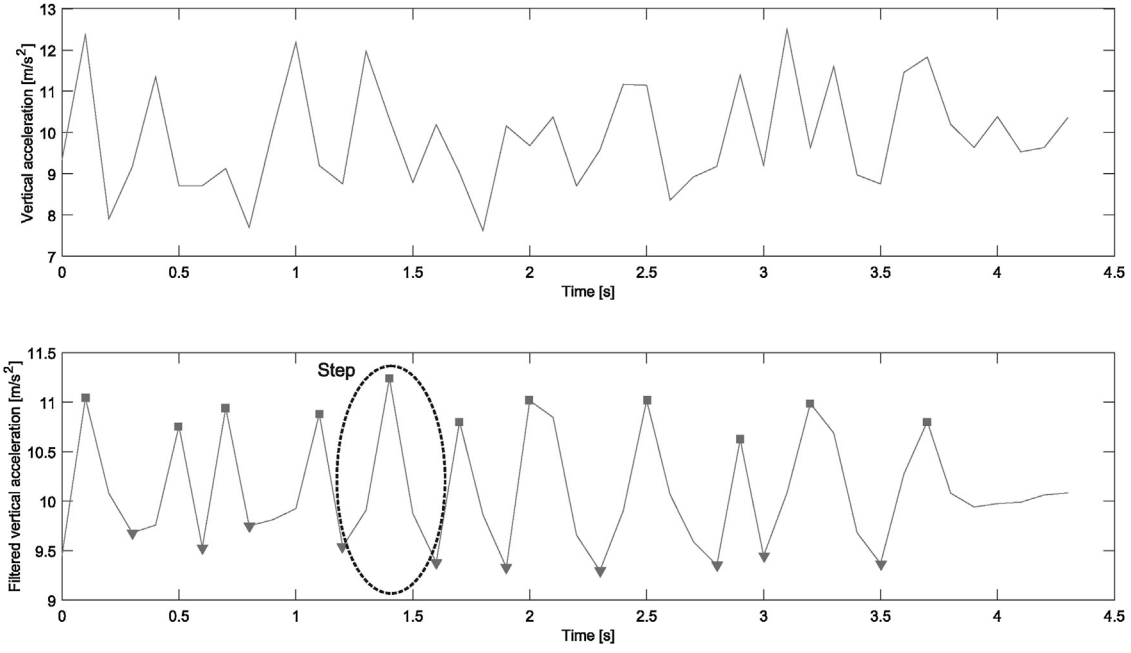


**Fig. 11.** Above: raw vertical acceleration; below: filtered vertical acceleration.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

step by using the following equation:

$$K = \frac{L}{\sqrt[4]{a_{\max} - a_{\min}}} \tag{28}$$

where $L$ is the step length measured by visual odometry as: $L_k = \sqrt{(x_k - x_{k-1})^2 (y_k - y_{k-1})^2}$. $(x_k,y_k)$ and $(x_{k-1},y_{k-1})$ are the positions of a smartphone in 2D world C.S. (see Section 2.5) at time $k$ and $k-1$ respectively. After five detected steps, the average value of the obtained constants $K$, which is then used in the calculation of stride length using Eq. (27), can be calculated. This process of setting the parameter $K$ can be repeated several times while the application is running, and in this way the value of the constant $K$ can be corrected to the new average value.

### 3.4. Fusion of visual odometry and inertial navigation system

Fusion is the combining of measurements (data) obtained by various sensors or methods in order to achieve more accurate and reliable information than when sensors or methods are used individually. In this study, the aim was to combine the information obtained by visual odometry and inertial navigation system using the Extended Kalman Filter (EKF) [31,37]. A technique of loosely coupled Kalman filter [5] was used, in which the visual odometry was calculated independently of the inertial navigation system. Both results were then combined with the already mentioned filter where inertial navigation system (compass and pedometer) was used as a prediction (to propagate the state of the filter) and visual odometry (2D transformation from the current in the world C.S.) as a correction (or measurement update). Since the Kalman filter is computationally efficient, it is suitable for the implementation on a smartphone. The Extended Kalman Filter is executed on its own thread, so that the operation of visual odometry is not slowed down at the fusion of information.

The fusion of visual odometry and inertial navigation system was performed in order to eliminate weaknesses of individual approaches and achieve a more accurate and robust tracking of the smartphone user. The visual odometry is especially important for determining the travelled distance, since the model for the stride length estimation is also based on visual odometry (to determine the constant $K$). In the visual odometry it has turned out that problems arise in the case of poor lighting conditions or high-speed turns when the traces of feature points are lost due to image blurring. In this case, the digital compass, which is a part of the inertial navigation system, offers an appropriate solution for correcting inaccurate rotations obtained by odometry. When the camera is directed towards the ground which has a very monotone texture, the features tracker provides traces with difficulty since there are not enough feature points. This means that traces are limited in number and a lot of them are incorrectly determined (e.g. in straight movement a trace appears which points to the left or right). Therefore, in such cases, it is desired that prediction (in the EKF) of smartphone position and heading is performed using inertial navigation system, which allows the reduction of the impact of incorrect results of visual odometry on the final estimation of the user's location. Inertial navigation system is not as accurate in determining the length of the route as visual odometry is, but it can satisfactorily compensate for the error that would otherwise arise in the case when the visual odometry breaks down. By the fusion (filtering) of headings from both systems the influence of magnetic disturbances and gyroscope drift are also reduced. Both visual odometry and inertial navigation system belong to the dead reckoning localization technique, which means that the position error in relation to the starting point increases over time. With the fusion of both systems, the error increasing is reduced but it cannot be completely eliminated due to the dead reckoning approach of these systems. On the other hand, the error in determining the heading does not increase over time, since the digital compass, which determines the absolute rotation by considering a magnetometer, is incorporated in the fusion of systems.

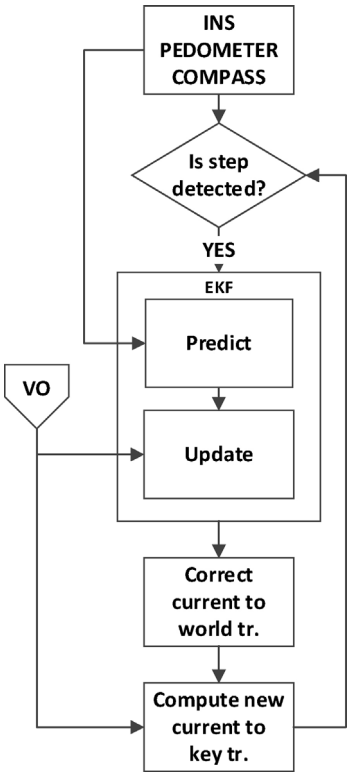


**Fig. 12.** The diagram of the fusion of visual odometry and inertial navigation system.

The crucial steps of fusion are illustrated with flowchart in Fig. 12, where it can be seen that the combining of information is carried out when the pedometer detects a new step. As the diagram shows, the output of inertial navigation system is used in the prediction phase and the output of visual odometry in the correction phase of the EKF. The outcome of the fusion is a corrected transformation from the current to the world C.S. In order to determine the new transformation from the key to the world C.S. correctly, after new traces have been added in the visual odometry algorithm, correction transformation needs to be calculated, which is determined by the difference between the corrected transformation from the current to the world C.S. and transformation that is returned by the visual odometry. This correction transformation is then considered in the correction of the transformation from the current to the key C.S. determined by visual odometry. Adding of new traces is performed with each detected step or when the number of active traces falls below a certain threshold.

The estimation of the position of a smartphone user based on the PDR technique starts at the initial coordinates $x_0$ and $y_0$ and initial heading $\theta_0$. The new position (after the next step) in 2D space $(x_k,y_k)$ and heading $\theta_k$ are then calculated using a non-linear state model:

$$\begin{bmatrix} \theta_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} \theta_{k-1} + & \Delta\theta_k \\ x_{k-1} + & l_k \cos\theta_k \\ y_{k-1} + & l_k \sin\theta_k \end{bmatrix}, \tag{29}$$

where $\Delta\theta_k$ is the change of the heading between the steps at discrete time $k-1$ and $k$ respectively and $l_k$ is the current stride length. Since the state model is non-linear, the ordinary Kalman

filter is not suitable and therefore an Extended Kalman Filter was used [5].

The elements of the state vector $x_k$ are as follows: $x_1$ = heading, $x_2 = x$ coordinate, $x_3 = y$ coordinate. Filtering or fusion with an Extended Kalman Filter started at the initial estimation of state $\hat{x}_0$ and the initial covariance $\boldsymbol{P}_0$. Initial state estimation is calculated at the first detected step using the current transformation from the current to the world C.S. which is determined by visual odometry algorithm. A Motion model which is used in prediction phase of an Extended Kalman Filter determines where the smartphone user would be at the moment $k$. Thus, the state propagation is defined as:

$$\hat{x}_k^- = \hat{x}_{k-1} + \begin{bmatrix} \Delta\theta_k \\ l_k \cos\hat{x}_{1_{k-1}} \\ l_k \sin\hat{x}_{1_{k-1}} \end{bmatrix}, \tag{30}$$

where $\hat{\boldsymbol{x}}_{k-1}$ denotes the posterior estimate after the measurement update using the measurement samples derived at $k-1$ time step. $\hat{\boldsymbol{x}}_k-$ represents the prior estimate for $k$-th time step, and $\hat{x}_{1_{k-1}}$ represents the previous posterior estimate of heading. Change of heading $\Delta\theta_k$ is determined by the difference between the current value of the compass and previous posterior estimate of heading $\hat{x}_{1_{k-1}}$. The stride length $l_k$ is computed by the model described in pedometer subsection.

The state matrix $\boldsymbol{F}_k$, which is necessary for the covariance propagation, is obtained by taking a partial derivative of (30):

$$\boldsymbol{F}_k = \begin{bmatrix} 1 & 0 & 0 \\ -l_k \sin\hat{x}_{1_k}^- & 1 & 0 \\ l_k \cos x_{1_k}^- & 0 & 1 \end{bmatrix} \tag{31}$$

The state noise $Q_k$ is calculated on each new step as:

$$\boldsymbol{Q}_k = \mathrm{diag}\left( \begin{bmatrix} \sigma_{\Delta\theta}^2 \\ \cos^2(\hat{x}_{1_k}^-)\sigma_l^2 \\ \sin^2(\hat{x}_{1_k}^-)\sigma_l^2 \end{bmatrix} \right) \tag{32}$$

where $\sigma_{\Delta\theta}^2$ is a variance of heading measurements obtained by the compass and $\sigma_l^2$ is a variance of step length estimates obtained by the pedometer. In prediction phase the prior covariance $\boldsymbol{P}_k^-$ is calculated as:

$$\boldsymbol{P}_k^- = \boldsymbol{F}_k \boldsymbol{P}_{k-1} \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{Q}_k, \tag{33}$$

where $\boldsymbol{P}_{k-1}$ is the posterior covariance from the previous time step.

The measurement input $\boldsymbol{z}_k = [\theta_{\mathrm{VO}_k} x_{\mathrm{VO}_k} y_{\mathrm{VO}_k}]^T$ is provided by the visual odometry and it consists of heading $\theta_{\mathrm{VO}_k}$ (i.e. yaw) and coordinates $x_{\mathrm{VO}_k}$ and $y_{\mathrm{VO}_k}$, which are determined by the transformation (in 2D space) from the current to the world C.S.

The measurement matrix $\boldsymbol{H}$ is equal to the unit matrix of size $3 \times 3$: $\boldsymbol{H} = \boldsymbol{I}_{3\times3}$. The measurement update of state $\hat{\boldsymbol{x}}_k$ and covariance $\boldsymbol{P}_k$ can be calculated as:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H}\boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R})^{-1} \tag{34}$$

$$\hat{\boldsymbol{x}}_k = \hat{x}_k^- + \boldsymbol{K}_k (\boldsymbol{z}_k - \boldsymbol{H}\hat{x}_k^-) \tag{35}$$

$$\boldsymbol{P}_k = (\boldsymbol{I}_{3\times3} - \boldsymbol{K}_k \boldsymbol{H}) \boldsymbol{P}_k^- \tag{36}$$

where $\boldsymbol{R}$ is the covariance of heading and position estimation with visual odometry and $\boldsymbol{I}_{3\times3}$ is a unit matrix. $\boldsymbol{K}_k$ represents the gain of the Kalman filter. The covariance matrix $\boldsymbol{R}$ is a diagonal matrix, which is determined by experimentally estimated variances of heading and position. The state $\hat{\boldsymbol{x}}_k$ provides the corrected transformation in 2D space from the current to the world C.S.

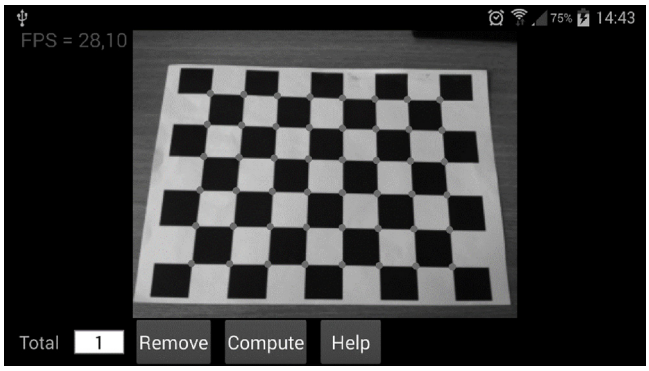


**Fig. 13.** Corners detection in chessboard.

## 4. Results

### 4.1. Calibration results

Calibration was performed on an Android smartphone by using the Java library BoofCV. The camera has a CMOS image sensor (sensor size is 1/3.06 or $4.69 \times 3.52$ mm) with the resolution of 13.25 MP and the lens with the focal length $f = 31$ mm. The camera calibration algorithm, which is described in detail above, is based on the work of Zhang [19]. In the calibration phase firstly the calibration object—chessboard was captured from different angles (20 images). In the process the calibration algorithm detected the corners (Fig. 13) by which the intrinsic camera parameters and the coefficients of the radial distortion were calculated.

When the smartphone was placed on the height and slope under which it was used for the purpose of visual odometry, another chessboard image was captured. From this image the extrinsic camera parameters and also the transformation $g_{CP}$ were calculated (Fig. 14).

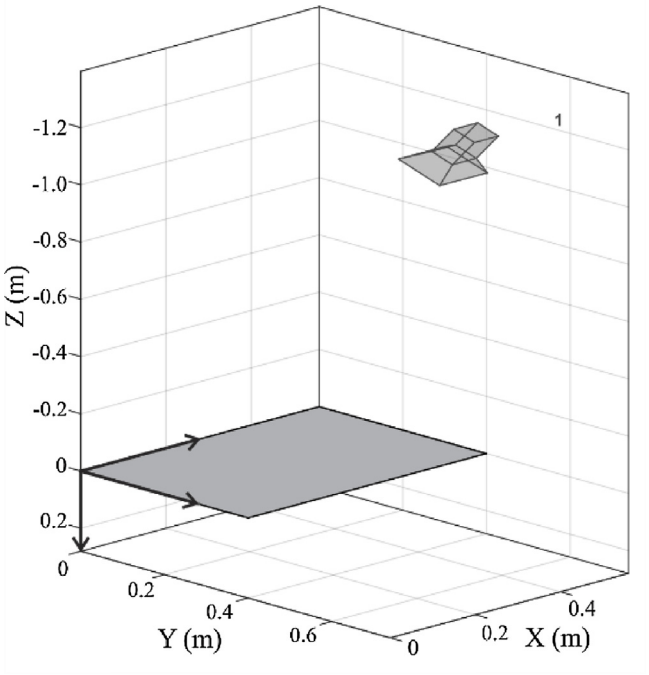


**Fig. 14.** Extrinsic parameters visualization.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

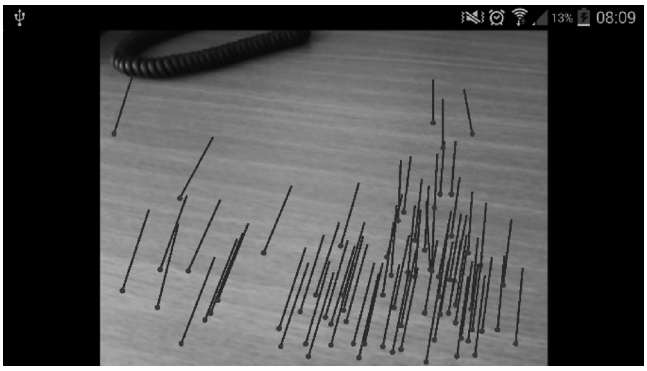


**Fig. 15.** Optical flow on the smartphone.

### 4.2. Optical flow on the smartphone

Fig. 15 shows the optical flow which provides for finding the traces of points in the visual odometry algorithm which determine the movements between the sequential image frames. The points on each image frame represent the corners which are detected by the Shi–Tomasi algorithm.

For proper functioning of the visual odometry it is important that the feature tracker provides a sufficient number of good traces in real time. Therefore, it was necessary to adjust the parameters of KLT tracker according to the computing power of the smartphone. The maximum number of traces is limited to 80, because if there are too many traces, then RANSAC algorithm with a limited number of iterations (e.g. 100 iterations) fails to determine the correct model of a rigid movement. For fast performance of visual odometry the number of RANSAC iterations must be as small as possible.

### 4.3. Determination of pedestrian movement through the visual odometry

The visual odometry algorithm is running on the smartphone in real time where it can be processed from 10 to 15 fps and this is entirely satisfactory for normal walking speed (2–3 steps per second). Faster operation of the algorithm was achieved primarily by splitting visual odometry algorithm into two parts which were carried out simultaneously on separate processor cores. For the purpose of testing the visual odometry, a smartphone was mounted on a pedestrian who walked along a 12 m reference trajectory which was drawn on the floor. The back of the smartphone was facing the ground so that the optical axis of the camera had an approximate inclination of 45° in relation to the ground. Fig. 16 shows the result obtained in one of the experiments with the visual odometry. Although this result is one of the worst obtained by using only visual odometry, it is most appropriate for the explanation of possible errors in the visual odometry. As can be seen in Fig. 16, the trajectory obtained with the visual odometry very accurately follows the reference trajectory at the beginning. However, at some point the trajectory starts to deviate from the reference trajectory due to incorrectly determined rigid transformation from the current to the key C.S. The problem is particularly the inaccurately determined heading, which causes that positional error quickly increases with time. This situation was due to the very monotone texture of the floor on which the experiment was performed. Thus, the algorithm KLT determined a lot of bad traces and, consequently, RANSAC algorithm failed to identify the correct model of rigid movement. Since at the same time this incorrect determined rigid transformation was considered also for update of the transformation from the key to the world C.S., all further positions were incorrectly defined. However, as can be seen in

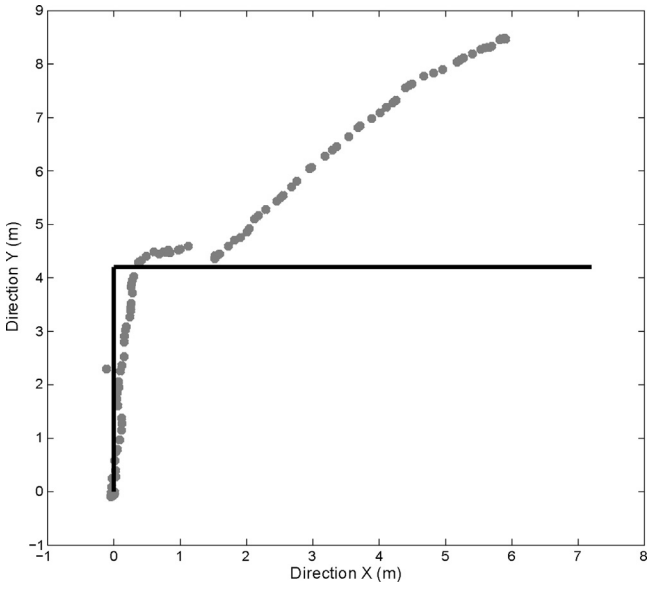


**Fig. 16.** An example of error at determining trajectory using visual odometry.

Fig. 16, the visual odometry worked properly from the defect onward, as the trajectory is almost as straight as the reference trajectory. So, if the heading had been suitably corrected at the moment when the wrong transformation was considered, the trajectory would have followed the reference trajectory accurately. The solution to the described problem lies in the fusion of visual odometry with inertial navigation system as described below.

### 4.4. Determination of pedestrian movement through inertial navigation system

In examining the inertial navigation system for pedestrians based on the PDR technique, a digital compass was considered to determine the heading and a pedometer was used for the purpose of detecting steps and measuring their length. Using the motion model which is described in the subsection of the fusion with the Extended Kalman Filter, the trajectory shown in Fig. 17 was obtained. The dots in the figure determine the position of the user

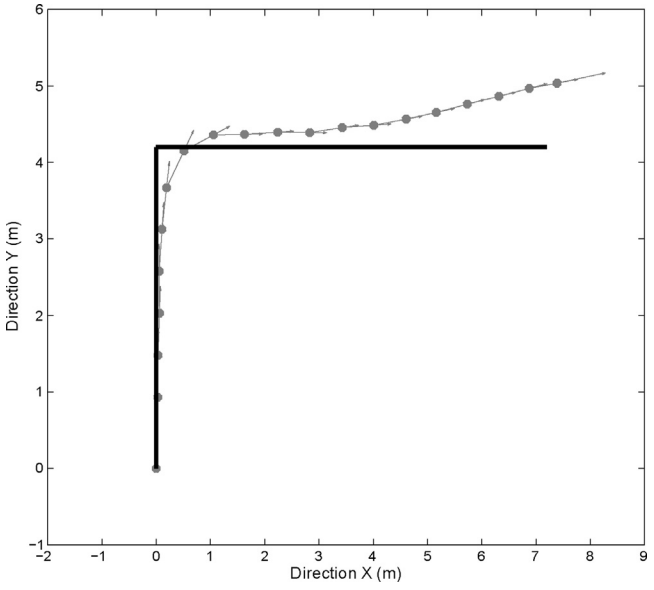


**Fig. 17.** Determination of pedestrian trajectory through inertial navigation system.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

and the arrows determine the heading at individual detected steps. As can be seen in Fig. 17, the trajectory follows the reference trajectory accurately at the beginning, but then slightly deviates due to not entirely accurate heading in the last part of the route. The length of the steps is well determined by the model (27) as the error of the total path length is equal to 2%. The error of the entire path length does not only depend on inaccurate stride length but also on whether the pedometer detects all the steps. At the stage of experimentation, a false step detection was not observed, since the smartphone was constantly fixed to the body. This prevents the shaking of the device which could lead to incorrect step detection.

In the space where the inertial navigation system was tested, especially in the last part of the path, a strong magnetic disturbance was present. Due to this disturbance, the ordinary digital compass, which is based only on magnetometer and accelerometer, turned in a completely wrong direction. If the arrows in Fig. 17 are examined closely, we can see that the compass described in this article works well even in the presence of a strong magnetic disturbance. Namely, the heading slightly deviates from the reference trajectory only in the last part of the path. However, the heading drifts away from the true value in any case if a long-term magnetic disturbance is present. This can happen due to the magnetometer, which returns incorrect values, or the gyroscope drift.

### 4.5. Determination of pedestrian movement with fusion of visual odometry and inertial navigation system

Both visual odometry and inertial navigation systems can operate independently and in certain cases they offer completely satisfactory results in determining the trajectory of a smartphone user. With the purpose of taking a step forward, these two relatively good localization approaches with pros and cons are combined in this study to eliminate potential weaknesses (which are mentioned in the analysis of the experimental results where systems are used individually) and to get more reliable and accurate results. In Fig 18, the results of the fusion of visual odometry and inertial navigation system using the Extended Kalman Filter are shown. If the obtained trajectory is compared with the reference trajectory in Fig. 18, a very small deviation (of about a few centimetres) can be seen. Furthermore, the fact that

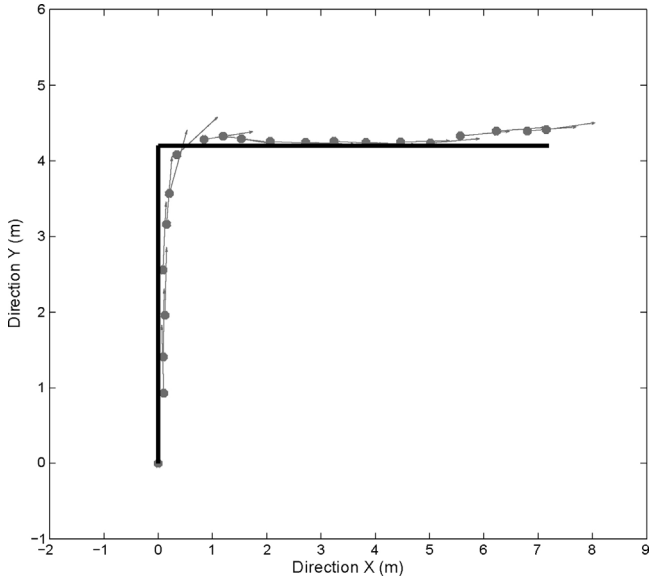


**Fig. 18.** Determination of pedestrian trajectory with fusion of visual odometry and inertial navigation system.

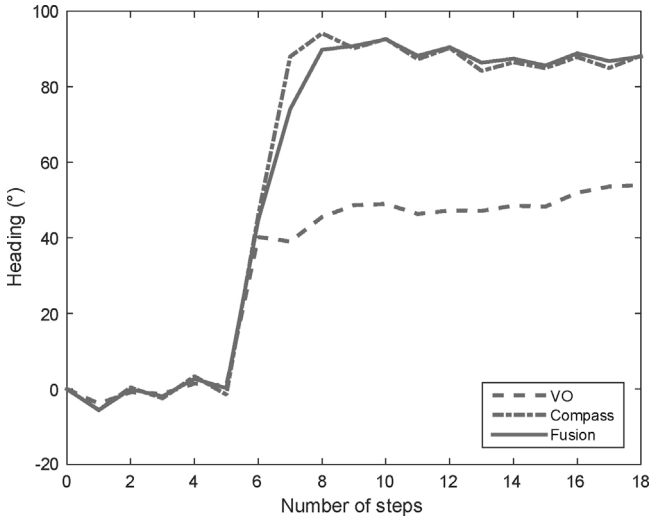


**Fig. 19.** Heading changes during the walk along the reference trajectory drawn in Fig. 18.

the test user has difficulty following the drawn reference trajectory accurately must be taken into account.

Fig. 19 shows the results of one of the first five experiments (experiment number four) where three curves present the results of determining the heading with visual odometry, digital compass and fusion of both approaches. According to the reference trajectory shown in Fig. 18, the heading values should be near to 0° at the beginning (approximately five steps from the starting point) and near to 90° towards the end of the reference trajectory (approximately the last eight steps). Although the pedestrian cannot be entirely aligned with the reference trajectory during the walk, the angles 0° for the first part and 90° for the second part of reference trajectory present the best approximation to the true heading.

To achieve as good results as they are shown in Fig. 18, the heading and the distance to the starting point must be very accurately determined. In order to show in details how heading and distance errors can be bounded by the implementation of the EKF based fusion of visual odometry and inertial navigation system, ten experiments were carried out. The first five experiments were made with the purpose of observing heading changes during the walk along reference trajectory such as shown in Fig. 18 and the second five experiments were made to analyse distance errors which occur by using the visual odometry, the inertial navigation system and fusion of both approaches. During the second five experiments, the pedestrian walked five times along straight reference line with the length of 10.2 m.

To evaluate the accuracy of determined heading values obtained with visual odometry, digital compass and fusion of both approaches, the mean squared errors (MSE) of heading values (which were modified to percentages) were computed for all five experiments (Fig. 20). In each experiment, the first column presents the MSE of heading obtained with visual odometry, the second column the MSE of heading obtained with digital compass and the third column the MSE of heading obtained with fusion of both approaches. In the calculation of the mean squared error, the heading values for the first five and last eight steps were considered, since during pedestrian's turning to the left the true heading values are not known.

In Fig. 20 can be seen that the MSE of heading values determined with visual odometry are very big for all five experiments since the visual odometry fails during the turn to the left (this can be seen in Fig. 19) and consequently the

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

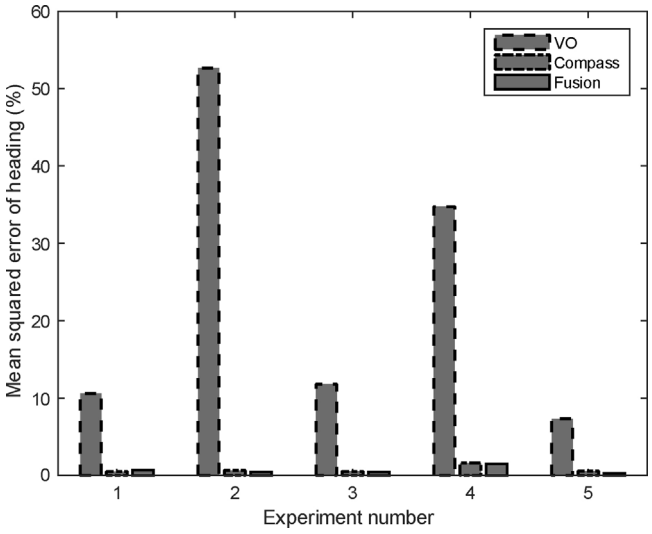


**Fig. 20.** The mean squared error of the heading (determined with VO, compass and fusion of both approaches) modified to percentages for five experiments in which the heading was changed as can be seen in Fig. 19.

measurements of heading for the second part of the reference trajectory are false. On the other hand, the MSE of heading values determined with digital compass are very small for all five experiments. This confirms that implemented digital compass can very accurately determine the heading also when high-speed turns occur. Since digital compass is included in the prediction part of the EKF, the heading is suitably corrected when the visual odometry fails and consequently the heading values obtained with fusion are also very accurately determined, i.e. the MSE of heading values are very small in all experiments.

In order to observe how the distances to the starting point determined with visual odometry, pedometer and fusion of both approaches are changed at each step, five experiments were carried out in which a pedestrian walked along a 10.2 m straight reference line. The results of one of the experiments (experiment number four) which can be seen in Fig. 21 show that the distances to the starting point are equal for all three approaches up to step five. Namely, for the first five steps the calibration procedure is performed in which constant $K$ (which is a part of the model for the step length estimation in the pedometer) is computed by using visual odometry measurements according to Eq. (28) and consequently each step length is equal to the measurements obtained by visual odometry. When the calibration procedure is finished, the step length is estimated according to the model (27). As can be seen in Fig. 21, the model very well describes the step length as all three curves very similarly increase with the number of steps.

For all five experiments and all three approaches, the relative error of the walking distance was estimated as shown in Fig. 22. The true distance (10.2 m) was known since the pedestrian walked straight on from the known starting point to the known finish point. The first columns of each experiment present relative errors of distances obtained with visual odometry, the second columns present relative errors obtained with pedometer and the third columns present relative errors of estimated distances obtained with fusion of both approaches. According to all relative errors shown in Fig. 22, the approach with fusion of visual odometry and pedometer manifests as the best solution in the long term, since visual odometry can be very accurate only at determing the walking distance in appropriate conditions (e.g. appropriate lighting). But in some cases when it fails (e.g. experiment number two) substantially worse results would be obtained as in the case of considering the fusion approach. Therefore in general the pedometer is not as accurate in determing the walking distance as visual odometry, but it is very important for the achievement of a long term accuracy. As can be seen in Fig. 22, all relative errors computed for distances obtained with fusion are very small, i.e. less than 3.5%.

In order not to slow down the operation of the visual odometry due to the implementation of the fusion, the Extended Kalman Filter is assigned to its thread, which can run on another processor core than odometry (which occupies two processor cores) does. The smartphone which has been used for the purpose of the testing contains four processor cores, which means that three cores have approximately the same burden. The fourth processor core is used by the Android OS to run the user interface (UI) thread.

In order to demonstrate the performance of the described localization system that runs in real time on a smartphone and in doing so does not require additional external infrastructure (e.g. WiFi network), an experiment in which a pedestrian walked the closed-loop path (clockwise) with length of 27 m (Fig. 23), was carried out. In addition to the six perpendicular turns which are on the test route, there is also a strong magnetic disturbance in the

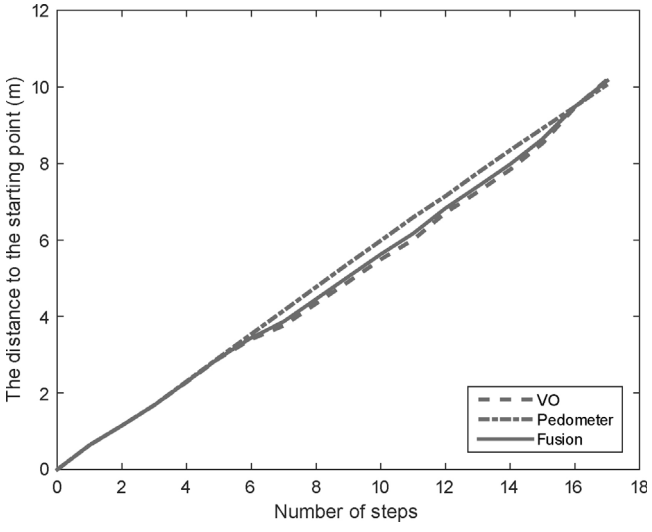


**Fig. 21.** The distance to the starting point during the walk along a 10.2 m straight reference line.

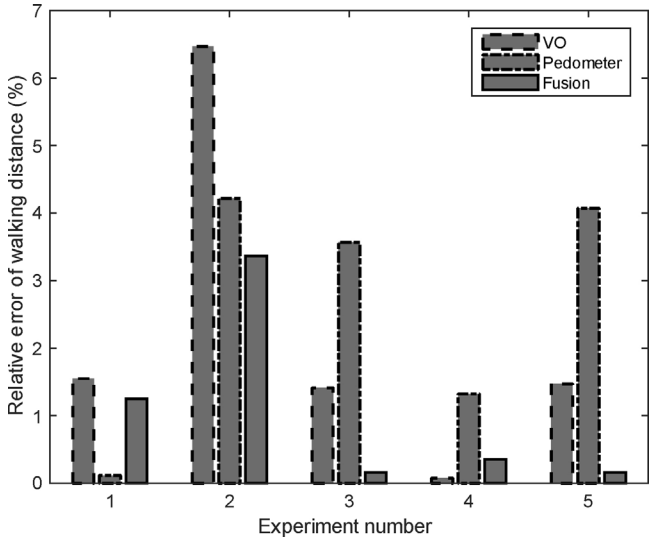


**Fig. 22.** The relative error of walking distance for five experiments.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

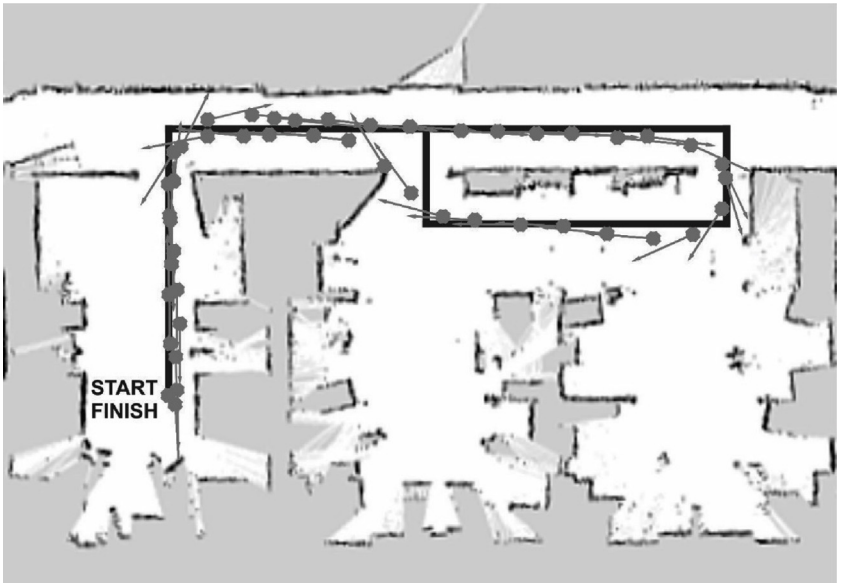


**Fig. 23.** Determination of pedestrian closed-loop trajectory with fusion of visual odometry and inertial navigation system.

main part of the room (where the trajectory forms a loop), which represents a major challenge for the digital compass. Ideally, the circles (in Fig. 23) should overlap with the ground truth trajectory but in reality this is not reasonably expected since even the pedestrian cannot follow the line so accurately. As can be seen in Fig. 23, the end point of the trajectory is away from the starting point for only few centimetres. This result is above and beyond all expectation for the dead reckoning technique.

## 5. Conclusion

In this paper the monocular visual odometry, inertial navigation system and the fusion of both these localization approaches are presented. The whole localization system was implemented on a smartphone, where it was divided into three parts which were carried out in separate threads. These three threads can run on separate processor cores, enabling the execution of several parts of the algorithms simultaneously. In this manner, the system can process from 10 to 15 fps. This is sufficient speed (at normal walking speed) for the visual odometry algorithm to operate as expected. With the implementation of this localization system, a new virtual sensor which measures incremental movements is obtained on the smartphone. This virtual sensor can be used in various applications on a smartphone (e.g. for guiding the blind) or as an independent (or additional) sensor unit on mobile robots.

The experiments have shown that both visual odometry and inertial navigation system can operate completely independently from each other. The advantage of the visual odometry is in accurately determining the relative positions, unless there are too many sharp turns on the route. The advantage of the inertial navigation system is mainly reflected in the possibility of accurately measuring the absolute heading by using a digital compass. The study showed that the problems of the visual odometry lie in the monotone texture of the floor or in poor lighting conditions.

In the inertial navigation system, the pedometer proved to be very reliable, since in testing there were no false step detections. Digital compass also works reliably if it is not exposed to long-term magnetic disturbance. Otherwise, the heading drifts slightly away from the true value, but this error can be eliminated by fusion. In order to get an even more robust and accurate localization system, the advantages of each mentioned approaches were combined by using the Extended Kalman Filter. This filter also eliminates most of the errors that occur in an individual system.

All localization experiments were carried out with a pedestrian on whom a smartphone was mounted. Instead of a wheel robot, a pedestrian was chosen for doing experiments since this way of testing represents a major challenge for the visual odometry. While walking, the camera shakes much more than in the case when it is attached to a robot. The results of the testing of the fusion of both systems have shown a high accuracy in determining the relative position in indoor spaces. An especially good result was achieved in the experiment when the pedestrian walked the path in the shape of a closed loop.

The disadvantage of the visual odometry, as well as of the inertial navigation system, is the inability to determine the absolute position in space if the position of the starting point is not given.

Therefore, the localization system will be improved with one of the global localization techniques (e.g. the positioning with WiFi or Bluetooth signals) in the future. Since precise relative movements between the global positions can be determined by the described system, high-accuracy global positioning is expected from this fusion. Using one of the global localization techniques can also limit errors that occur due to dead reckoning approach in visual odometry and inertial navigation system.

## References

[1] V. Honkavirta, T. Perala, S. Ali-Loytty, R. Pich'e, A comparative survey of wlan location fingerprinting methods, Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on. IEEE 2009 (2009) 243–251.
[2] J. Ramos, A. Costa, P. Novais, J. Neves, Interactive guiding and localization platform, Int. J. Artif. Intell. 12 (1) (2014) 63–78.
[3] J. Weichselbaum, C. Zinner, O. Gebauer, Pree, W, Accurate 3D-vision-based obstacle detection for an autonomous train, Comput. Ind. 64 (9) (2013) 1209–1220.
[4] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robot. Autom. Mag. 13 (2) (2006) 99–110.
[5] S. Sirtkaya, B. Seymen, A.A. Alatan, Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation, 16th International Conference on, Information Fusion (FUSION) 1 (2013) 219–226.
[6] B. Silva, A. Burlamaqui, L. Goncalves, On monocular visual odometry for indoor ground vehicles, Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), Brazilian, 2012 pp. 220–225.
[7] J. Straub, S. Hilsenbeck, G. Schroth, Huitl, R, et al., Fast relocalization for visual odometry using binary features, Proceeding of: IEEE International Conference on Image Processing (ICIP) (2013) 2548–2552.
[8] G. Navarro, M. Manic, FuSnap: fuzzy control of logical volume snapshot replication for disk arrays, IEEE Trans. Ind. Electron. 58 (9) (2011) 4436–4444.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003

[9] D. Nister, O. Naroditsky, J. Bergen, Visual odometry, Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on (2004) 652–659.
[10] J. Campbell, R. Sukthankar, I.R. Nourbakhsh, A. Pahwa, A robust visual odometry and precipice detection system using consumer-grade monocular vision, IEEE International Conference on Robotics and Automation (ICRA 2005) (2005) .
[11] M. Agrawal, K. Konolige, Rough terrain visual odometry, Proceedings of the International Conference on Advanced Robotics (ICAR 2007) (2007) .
[12] BoofCV, Calibration. <http://boofcv.org/index.php?title=Tutorial_Camera_Calibration>, 2014 (1.6.2014).
[13] Robot Operating System – ROS. <http://www.ros.org/>, 2014 (1.12.2014).
[14] H. Zhong, J.P. Wachs, S.Y. Nof, Telerobot-enabled HUB-CI model for collaborative lifecycle management of design and prototyping, Comput. Ind. 65 (4) (2014) 550–562.
[15] R. Precup, S. Preitl, M. Radac, E.M. Petriu, et al., Experiment-based teaching in advanced control engineering, IEEE Trans. Edu. 54 (3) (2011) 345–355.
[16] J. Ramos, P. Novais, K. Satoh, T. Oliveira, et al., Speculative orientation and tracking system, Int. J. Artif. Intell. 13 (1) (2015) 94–119.
[17] A. Sanchis, V. Julián, J.M. Corchado, Billhardt, H, et al., Improving human-agent immersion using natural interfaces and CBR, Int. J. Artif. Intell. 13 (1) (2015) 81–93.
[18] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, 4th ed., Cengage Learning, Stamford, USA, 2014, pp. 598–601.
[19] Z. Zhang, A flexible new technique for camera calibration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1330–1334.
[20] B.D. Lucas, T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence – Volume 2, (1981) pp. 674–679. Morgan Kaufmann Publishers Inc. Canada.
[21] C. Tomasi, T. Kanade, Detection and Tracking of Point Features. Int. J. Comput. Vision. Technical Report, Carnegie Mellon University (1991).
[22] J. Shi, C. Tomasi, Good features to track, Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on (1994) 593–600.
[23] J.-Y. Bouguet, Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm. Technical Report, Intel Corporation Microprocessor Research Labs (2001).
[24] J.K. Suhr, Kanade–Lucas–Tomasi (KLT) Feature Tracker. Computer Vision Laboratory. Seoul, Korea (2009).
[25] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[26] D.A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, 2003, pp. 365–375.
[27] Y. Ma, S. Soatto, J. Kosecka, S.S. Sastry, An Invitation to 3-D Vision: From Images to Geometric Models, SpringerVerlag, 2004, pp. 19–37.
[28] D.W. Eggert, A. Lorusso, R.B. Fisher, Estimating 3-D rigid body transformations: a comparison of four major algorithms, Mach. Vis. Appl. (1997) 272–290.
[29] R. Szeliski, Computer Vision Algorithms and Applications, Springer, London, 2011, pp. 320–321.
[30] K.S. Arun, T.S. Huang, S.D. Blostein, Least-squares fitting of two 3-D point sets, IEEE Trans. Pattern Anal. Mach. Intell. 9 (5) (1987) 698–700.
[31] H. Leppakoski, J. Collin, J. Takala, Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals, IEEE International Conference on, Acoustics, Speech and Signal Processing (ICASSP) (2012) 1569–1572.
[33] Android Coordinate System. <http://developer.android.com/reference/android/hardware/SensorEvent.html#values>, 2014 (20.11.2014).
[34] K. Shoemake, Nimating rotation with quaternion curves, Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '85, ACM, New York, NY, USA, 1985, pp. 245–254.
[35] T. Zengshan, Z. Yuan, Z. Mu, L. Yu, Pedestrian dead reckoning for MARG navigation using a smartphone. EURASIP J. Adv. Sig. Pr., 65(1) (2014). Springer.
[36] M. Hardegger, G. Tröster, D. Roggen, Improved actionSLAM for long-term indoor tracking with wearable motion sensors, Proceedings of the 2013 International Symposium on Wearable Computers, ISWC '13, ACM, New York, NY, USA, 2013, pp. 1–8.
[37] T. Gallagher, E. Wise, L. Binghao, A.G. Dempster et al, Indoor positioning system based on sensor fusion for the blind and visually impaired, International Conference on, Indoor Positioning and Indoor Navigation (IPIN) 2012 (2012) 1–9.

**Simon Tomažič** received bachelor's degree in electrical engineering in 2012 at the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. He is currently a young researcher at the same faculty. His main research interests include indoor positioning and navigation in combination with Android-based smartphone. His research is based on the use of smartphone cameras for the monocular visual odometry purposes. He is also interested in using inertial sensors and digital compass and sensor fusion with different types of Kalman filters. He is enthusiastic about unmanned ground vehicles and their use for different purposes in housekeeping, army and health care. He has experience in machine vision, inertial sensors, digital compass, sensor fusion, fuzzy control, simulations, mathematical and computer modelling. He is the author of four conference papers and three journal papers.

**Igor Škrjanc** received B.S., M.S. and Ph.D. degrees in electrical engineering, in 1988, 1991 and 1996, respectively, at the Faculty of Electrical and Computer Engineering, University of Ljubljana, Slovenia. He is currently a Full Professor with the same faculty. He is lecturing the basic control theory at graduate and advanced intelligent control at postgraduate study. His main research areas are adaptive, predictive, neuro-fuzzy and fuzzy adaptive control systems. His current research interests include also the field of autonomous mobile systems in sense of localization, direct visual control and trajectory tracking control. He has published 79 papers with SCI factor and 25 other journal papers.He is co-author and author of 8 chapters in international books and co-author of scientific monograph with the title Predictive approaches to control of complex systems published by Springer. He is author of 6 university books, 24 international and domestic projects and 4 patents. In 1988 he received the award for the best diploma work in the field of Automation, Bedjanič award, in 2007 the award of Faculty of Electrical Engineering, University of Ljubljana, Vodovnik award, for outstanding research results in the field of intelligent control, in 2012 the 1st place at the competition organized by IEEE Computational Society, Learning from the data in the frame of IEEE World Congress on Computational Intelligence 2012, Brisbane, Australia: solving the sales prediction problem with fuzzy evolving methods, and in 2013 the best paper award at IEEE International Conference on Cybernetics in Lausanne, Switzerland. In 2008 he received the most important Slovenian research award for his work in the area of computational intelligence in control—Zois award. In year 2009 he received a Humboldt research award for long term stay and research at University of Siegen. He is also a member of IEEE CIS Standards Committee and Slovenian Modelling and Simulation society and Automation Society of Slovenia. He also serves as an Associated Editor for IEEE Transaction on Neural Networks and Learning System, IEEE Transaction on Fuzzy Systems and the Evolving Systems journal.

Please cite this article in press as: S. Tomažič, I. Škrjanc, Fusion of visual odometry and inertial navigation system on a smartphone, COMIND (2015), http://dx.doi.org/10.1016/j.compind.2015.05.003