Sensors and Actuators A 301 (2020) 111731

Contents lists available at ScienceDirect

Sensors and Actuators A: Physical

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

Implementation and calibration of an odometry system for mobile robots, based on optical computer mouse sensors

A.F.M. Paijens, L. Huang*, A.M. Al-Jumaily

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand

A R T I C L E I N F O

Article history:
Received 16 April 2019
Received in revised form 31 October 2019
Accepted 7 November 2019
Available online 16 November 2019

Keywords:
Optical computer mouse
Optical flow sensor
Odometry
Mobile robots
Multiple sensors
Synchronized data acquisition
Calibration

A B S T R A C T

Although the application of computer mouse sensors as optical flow recorders in mobile robot odometry holds great promise, it is still hampered by inconsistent performance of the sensors in measuring displacements of the robot on different floor surfaces. This issue can be overcome by a re-calibration of the sensitivity of the sensor measurement when the robot moves on a new type of surface. With the calibrated figure for the sensitivity of each sensor, the positions and orientations of the sensors on the robot frame can be derived from the sensor readings of a set of constrained displacements of the robot. Sensitivity, position and orientation of each sensor are the key parameters in the conversion from the optical flow measured by the sensors, into an estimated displacement of the robot. To minimize the occurrence of systematic errors in this conversion, it is essential to calibrate these parameters as accurately as possible.

This paper proposes a novel procedure to perform their calibration based on a micro-controller setup that implements strict synchronization of the acquisition of optical flow data from all mouse sensors. Execution of the procedure is simple and requires no more than one measurement with a yardstick to obtain the calibrated figures for all key parameters. The collected data set can also be used to verify the calibration with a position calculation. The observed inaccuracy of the calculated location represents an excellent benchmark to compare the performance of different robot localization systems. The efficacy of the calibration method is experimentally tested and validated in a number of calibration scenarios.

1. Introduction

Odometry refers to an estimation of the displacement of a robot or vehicle in an incremental way, relative to its previous estimated position. It is used in mobile robot localization in combination with absolute position measurement to periodically recalibrate the vehicle's location [1–3].

Odometry is a field of mature solutions dominated by wheel shaft encoder reading [4]. Unfortunately recording of displacement with wheel encoders is inhibited by errors such as slippage of the wheels and the presence of particles on the surface in the path of the wheels [2,3,5,6]. As the consequence of their non-tactile measurement method, optical mouse sensors avoid these pitfalls and have the added advantage of operating without moving parts. Computer mouse sensors are cheaper than wheel encoders, while performing odometry at a high resolution [7–9].

Although promising, the operation of optical mouse sensors as displacement sensors is not without limitations. The key parameters in its operation are the sensitivity, position and orientation of each sensor on the robot frame. Errors in these parameters will give rise to systematic errors in the calculated position and orientation of the robot. Opposite to random errors that can compensate each other as the consequence of averaging, systematic errors will accumulate with each incremental step.

The sensitivity of an optical computer mouse sensor tends to be inconsistent and depends on the distance between the sensor aperture and the floor surface and on properties of the sensor's motion trajectory such as velocity and curvature [7,10–12]. To remedy these limitations, modifications have been proposed to the sensor optics to render it afocal, effectively mitigating the height dependency of the sensitivity [13]. To further improve the performance several researchers have proposed and successfully demonstrated the use of multiple redundant mouse sensors and combinations with other position measurement methods through sensor fusion [10,14–16].

Under the condition that the displacements recorded by the different mouse sensors are measured strictly coincidentally, the displacement and rotation of the robot can be calculated through

* Corresponding author.
E-mail addresses: antonius.paijens@aut.ac.nz (A.F.M. Paijens), loulin.huang@aut.ac.nz (L. Huang), ahmed.aljumaily@aut.ac.nz (A.M. Al-Jumaily).

https://doi.org/10.1016/j.sna.2019.111731
0924-4247/

linear regression [17,18]. Research towards the optimal geometrical arrangement of multiple mouse sensors on the robot frame for this conversion indicates that placement of the sensors furthest away from the geometrical center of the robot yields the most accurate calculation of position and orientation of the robot [18–22]. The results also demonstrate that the orientation at which the sensor is mounted in the robot frame has no effect on the accuracy of the calculation. Nonetheless, orientation and position of each sensor in the robot frame are the parameters in a regression model. Their exact values are hard to establish in a setup using optical computer mouse sensors, as the position and orientation of the small apertures of the sensors are difficult to measure and the location of the sensors in the robot frame might not coincide with their design value, due to manufacturing tolerances.

This problem can be solved through a one-time calibration measurement of the robot displacement using a separate highly accurate calibration system. The calibration measurements reveal the deviation in the robot's displacement as measured by the mouse sensors. Through its relation with the errors in the sensors' positions, the deviation can then be traced back to improve the accuracy of the sensor positions established [18]. The method does not evaluate the sensors orientations, but assumes them to be aligned with the robot frame coordinate system.

In another approach a robot is moved over an arbitrary homogeneous path comprising of a line or an arc, recorded by the mouse sensors on the robot frame. The recorded displacements of the separate sensors are coupled by the kinematics of the robot, in which the position and orientation of the sensors in the robot frame are parameters. These parameters are estimated by minimization of the difference between the displacement derived from the kinematic model and the displacement as recorded by the sensors [23]. The method still requires an accurate figure for the distance between at least two sensors on the robot, which is difficult to obtain. The calculation of the sensor positions and orientations is a brute force computation solving all parameters for all sensors in one minimization operation, while reconstructing the path of the robot at the same time. The calibration method is validated in an additional experiment by moving the robot in a full circular path, measuring the distance and orientation from the end position as measured by the sensors to the starting position of the path. Although the results reported are satisfactory, a more analytic method could potentially obtain better results more efficiently.

The linear regression model used in these papers to relate the displacements of the optical flow sensors to the translation and rotation of the robot, is purely static. This condition presumes synchronization of the acquisition of displacement data from the sensors in a schedule of strict coincidence. Uncertainty about the times at which the displacement was recorded by the different sensors translates into uncertainty of the calculated position and orientation of the robot. Notwithstanding this prerequisite, previous papers completely omit the aspect of synchronicity in data collection from different sensors or disregard explanation on the design and implementation of a scheduled data acquisition system [17,18,23].

The approach presented in this paper improves on the previous calibration and measurement methods in multiple aspects. It starts with a simple, but cardinal calibration of the sensor sensitivity, using a yardstick only. In a subsequent operation, it separately calibrates the orientation and position of all optical mouse sensors, by subjecting the robot frame to pivoting motions around fixed pivot points, without any prior knowledge of the location of the sensors or the pivot points. By forcing the robot to travel over arcs of constant radius, the orientation and position of the sensors can be derived directly from their displacement as measured, using a model of the kinematics of the pivot motion. Since the pivot point is stationary in each experiment, its displacement as calculated with the calibrated values of the parameters of the odometry system directly reflects the measurement error of the position and orientation of the robot and can serve as a validation for the calibration.

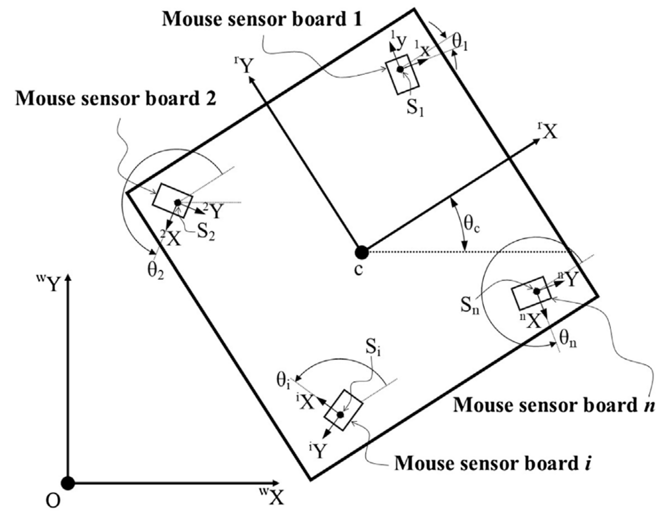

Fig. 1. Contour of a mobile robot with mouse sensors and corresponding coordinates systems.

This approach is innovative in the following three aspects. Firstly, it requires only one measurement to set the length of a straight path to calibrate the sensor sensitivities. Orientations and positions of the sensors can be derived subsequently from the displacements measured by the optical flow sensors on the robot in the pivoting experiments. Hence the operation described is highly suitable to be implemented as a self-calibration procedure.

Secondly, validation of the calibration through calculation of the displacement of the pivot point requires no additional measurements and defines an unambiguous measure to compare the performance of different robots and navigation systems.

Thirdly, the developed micro controller based setup implements strict synchronization of the data acquisition by multiple optical flow sensors over a data bus (I2C) without requiring additional hardware. To the best of our knowledge, no other sources elaborating the synchronization of the data collection by multiple optical mouse sensors are available in the open literature.

The paper is organized as follows: after an introduction to odometry in Section 2, Section 3 describes the electronic design of an odometry with optical mouse sensors that secures proper synchronization of the data acquisition from all mouse sensors. Section 4 explains the design of the calibration procedure. Section 5 describes the execution of the experiments performed as a part of the calibration procedure. The experimental results are validated with calculations of the robot path in Section 6. The results and improvements demonstrated are listed in Section 7.

2. Odometry using optical mouse sensors

An optical computer mouse sensor can measure its linear displacement only, leaving its rotational angle (angular displacement) undecided. Therefore a mobile robot with three degrees of freedom requires at least two mouse sensors at different positions to compute the full planar (linear and angular) motion. With two or more mouse sensors in place, the displacement data measured is redundant and allows for calculation of the robot motion by means of least squares regression [20].

The square in Fig. 1 depicts the contour of a mobile robot frame with n computer mouse sensors at positions s_i, $(i = 1, \ldots, n)$. The robot moves in a planar space spanned by the world coordinate system ${}^{w}X - {}^{w}Y$ with its origin O in a chosen position, whereas the robot has its local ${}^{r}X - {}^{r}Y$ coordinate system which has its origin in the center of mass (or another representative point) $[x_c\ y_c]^T$; its direction is represented by θ_c, the angle between axes ${}^{r}X$ and ${}^{w}X$. Each of the mouse sensors has its own coordinate system ${}^{i}X - {}^{i}Y$ with its origin $[x_i\ y_i]^T$ in the center of the aperture of the sensor; its orientation is represented by θ_i, the angle between ${}^{i}X$ and ${}^{r}X$.

For simplicity, the linear displacements of the robot and the sensors are put in a vector respectively defined as

$$c = [x_c \quad y_c \quad \theta_c]^T$$

$$s_i = [x_i \quad y_i]^T$$

The linear displacement of the sensor s_i, caused by the motion of the robot, can be expressed in the robot coordinate frame as:

$$\Delta s_i = \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = \begin{bmatrix} \Delta x_c \\ \Delta y_c \end{bmatrix} + \Delta\theta_c \begin{bmatrix} -y_i \\ x_i \end{bmatrix} \tag{1}$$

where $[\Delta x_c\ \Delta y_c]^T$ and $\Delta\theta_c$ are the linear and angular displacement of the robot expressed in the robot coordinate frame.

The result can be expressed as a displacement $\Delta^{i}s_i$ in the sensor coordinate frame by space transformation:

$$\Delta^{i}s_i = R(\theta_i) \quad \Delta s_i \tag{2}$$

where the superscript i refers to the coordinate frame of sensor s_i and $R(\theta_i)$ is a rotation matrix corresponding to the angle θ_i of sensor s_i in the robot coordinate system:

$$R(\theta_i) = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \tag{3}$$

Note by default without a left superscript, a vector is expressed in the robot coordinate frame.

Eqs. (1)–(3) can be combined:

$$\Delta^{i}s_i = A_i \quad \Delta c \tag{4}$$

where $\Delta c = [\Delta x_c\ \Delta y_c\ \Delta\theta_c]^T$ and $A_i \in \mathbb{R}^{2\times 3}$:

$$A_i = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & x_i \sin(\theta_i) - y_i \cos(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) & x_i \cos(\theta_i) + y_i \sin(\theta_i) \end{bmatrix}$$

Expression (4) can be converted to a system of $2\,n$ linear equations:

$$\Delta^{s}s = A\,\Delta c \tag{5}$$

Table 1
Location of corners of the robot frame.

Location of corners	1	2	3	4
CAD dimensions [mm]	(0.0, 0.0)	(0.0, 326.0)	(326.0, 326.0)	(326.0, 0.0)
Hand measurement [mm]	(0.0, 0.0)	(−2.3, 326.7)	(324.5, 326.6)	(326.9, 0.0)

where $\Delta^{s}s \in \mathbb{R}^{2n}$ is the vector of displacements registered by all sensors in their own coordinate system and $A \in \mathbb{R}^{2n\times 3}$:

$$\Delta^{s}s = [\Delta^{1}s_1 \quad \Delta^{2}s_2 \quad \ldots \quad \Delta^{i}s_i \quad \ldots \quad \Delta^{n-1}s_{n-1} \quad \Delta^{n}s_n]^T$$

$$A = [A_1 \quad A_2 \quad \ldots \quad A_i \quad \ldots \quad A_{n-1} \quad A_n]^T$$

Eq. (5) can be used to calculate an ordinary least squared estimator $\Delta\hat{c}$ for the robot displacement, when at least three independent displacement figures, comprising Δx_i and Δy_j, $(i, j = 1, \ldots, n)$ are available. The resulting optimal (least squares) estimator for Δc can be calculated as [24]:

$$\Delta\hat{c} = (A^T A)^{-1} A^T \Delta^{s}s \tag{6}$$

It is found that the variance of the solution of the robot displacement according to Eq. (5) is inversely proportional to the number of sensors used [18].

3. Setup for odometry using optical mouse sensors

To assess the accuracy of odometry using optical computer mouse sensors, a robot frame with four mouse sensors has been assembled, sitting on four legs. The frame, depicted in Fig. 2 can make hand powered, guided translations and rotations on rounded ends capping the legs at the bottom. Table 1 lists the locations of the corners of the frame in the CAD design and according to hand measurements of the actual frame produced by a laser cutter. The frame was cut on a UNIVERSAL™, 800*450 mm flatbed industrial grade laser cutter, producing a part of which the supposedly square corners were about 0.4°(0.4 %) off, resulting in a misplacement of over 2 mm of the corners of the frame of 326 mm square.

The sensors used are ADNS9800 optical laser mouse sensors from Pixart Imaging Inc. [25]. As can be observed in the picture in Fig. 2, the robot frame is rigged with four sensors, each mounted on a printed circuit board (PCB). The PCB's are mounted at the bottom side of the robot frame facing down to allow the sensors to sense the optical flow resulted from the relative motion between the robot frame and the floor. The sensors apertures are capped with the standard lenses provided by the manufacturer. They are mounted to keep the tops at 10.75 mm height over the floor surface, in compliance with the recommendations by the sensor manufacturer. The orientation of the sensors at the bottom side of the frame,

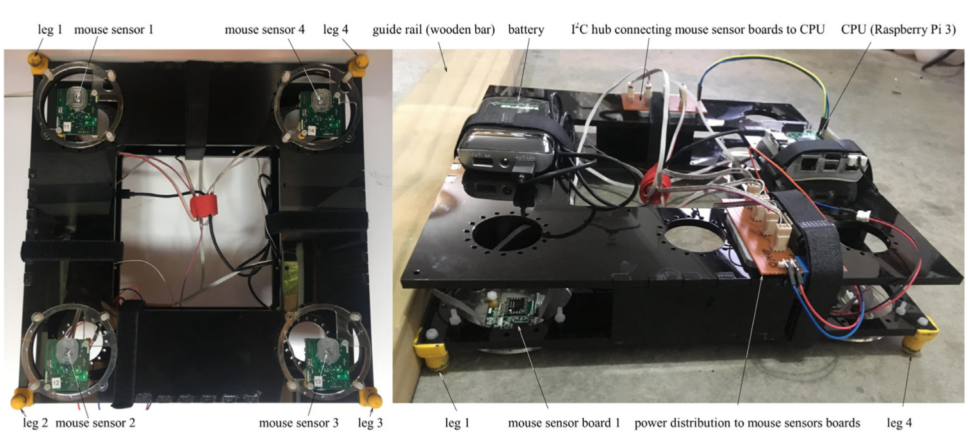

Fig. 2. Bottom (left) and top view (right) of sliding robot frame.

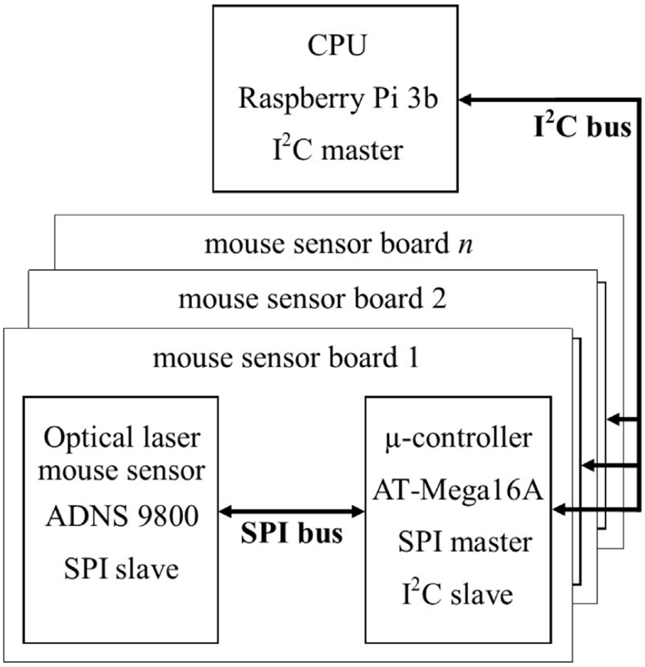

Fig. 3. Schematic of the data acquisition system.

is in alignment with the robot frame orientation, meaning that the ${}^iY$ and ${}^rY$ axes as well as ${}^iX$ and ${}^rX$ axes have the same direction: $\theta_i \approx 0, (i = 1, \ldots, n)$.

The setup of the data acquisition system is shown schematically in Fig. 3. Each sensor is connected to an AT-Mega 16A micro controller ("μC") acting as master at the SPI interface with the sensor. This enables the μC to retrieve the displacement data from the registers of the sensors without interrupting the data recording, preventing jitter. The sensor accumulates the displacement data until it is reset by the data retrieval by the μC. The four AT-Mega μC's are connected as slaves on an I^2C bus to a Raspberry Pi 3b, acting as bus master and CPU. This implementation allows the data acquisition to be synchronized by a "general call" by the CPU over the I^2C bus. A general call is received by all devices on the bus. For the odometry the general call initiates an interrupt routine at all μC's at exactly the same time to retrieve the displacement data from the connected mouse sensors. This marks the transition to a new sampling period for the mouse sensors. Subsequently the CPU requests the μC's one by one to pass the retrieved displacement data over the I^2C bus. When all μC's have passed their data to the CPU, it starts the next sampling interval by generating another "general call". The sampling frequency is not real time controlled this way but has a fairly constant value of 650 ± 5 Hz in the setup described.

The odometry system can be expanded with additional sensor boards, by simply hooking them up to the I^2C bus and addition of their device addresses to the list of boards that is included in the data retrieval loop of the CPU.

4. Calibrations

In this section, an innovative approach for the calibration of the optical computer mouse sensors for the odometry of a mobile robot is described. The terms to be calibrated include sensitivity and accuracy.

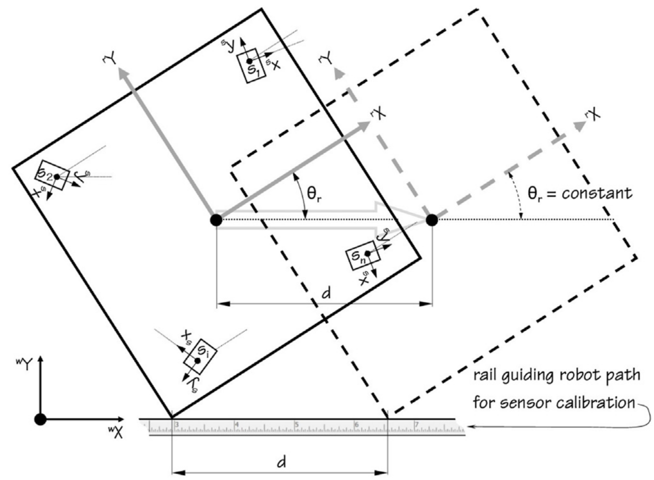

Fig. 4. Motion of the robot frame to measure the sensor sensitivity f_i.

4.1. Sensor sensitivity (resolution)

Sensitivity refers here to the ratio between the reading from the sensor (mostly referred to as "number of pixels") and the corresponding distance (in meters) it moves on the surface. Consequently its unit is "pixels per meter" [ppm].

It has been observed that the sensor sensitivity is strongly dependent on the height of the sensor relative to the surface [7,10,11]. The easiest way to measure the sensitivity of the mouse sensors is illustrated in Fig. 4. The robot frame with sensors is moved in a straight line over a known distance d without rotating it, while recording the $x - y$ displacement in pixels (Δx_{p_i}, Δy_{p_i}) as registered by the mouse sensors s_i.

The sensitivity f_i of the sensor s_i is given by

$$f_i = \frac{1}{d} \cdot \sqrt{(\Delta x_{p_i})^2 + (\Delta y_{p_i})^2} \tag{7}$$

Since it is independent to the motion direction of the sensor, according to the manufacturers specification [25], the solution of Eq. (7) applies to any direction in which the mouse is moved, as long as it is in a straight line.

4.2. Sensor position (location and orientation)

The procedure to obtain the position of a mouse sensor in the robot frame, using multiple pivot points p_k, $(k = 1 \ldots m)$ on the robot, is illustrated in Fig. 5. It is assumed that the sensor sensitivities f_i are available through the simple calibration procedure described in Section 4.1 and that the sensors are mounted in alignment with the robot frame: $\theta_i \approx 0$.

No other information of the robot frame or sensors is required. As such the locations of the pivot points: $p_k = [x_k\ y_k]^T$, the position of ith sensor $s_i = [x_i\ y_i]^T$ and the vector $\vec{p_k s_i}$ (with length $|\vec{p_k s_i}|$ and direction angle β with the ${}^rX$ axis) between them are unknown. The same angle β occurs between the ${}^rY$ axis and the direction (tangent) of arc A_{ki} at the starting point of the path of sensor s_i when the robot frame is pivoted around point p_k. The angle over which the frame is pivoted is equal to α_k and the length of the path (arc A_{ki}) traveled by the sensor is $|A_{ki}|$. The path is recorded by mouse sensor s_i as a displacement $\Delta^i s_i = [\Delta^i x_i \quad \Delta^i y_i]^T$, which can be expressed in the robot coordinates frame as $\Delta s_i = R(-\theta_i)\, \Delta^i s_i$.

The following relations exist between these measures:

$$|A_{ki}| = \alpha_k \quad |\vec{p_k s_i}| \tag{8}$$

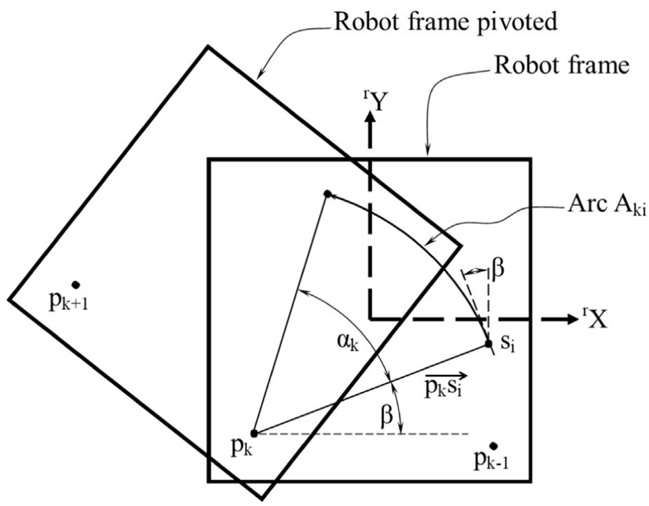

Fig. 5. Robot frame pivoting over angle α_k.

and:

$$\begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = |A_{ki}| \begin{bmatrix} -\sin(\beta) \\ \cos(\beta) \end{bmatrix} \tag{9}$$

From Eqs. (8) and (9), it follows that:

$$\vec{p_k s_i} = |\vec{p_k s_i}| \begin{bmatrix} \cos(\beta) \\ \sin(\beta) \end{bmatrix} = \frac{1}{\alpha_k} R\left(\frac{\pi}{2} - \theta_i\right) \Delta^i s_i \tag{10}$$

Hence, the position of the sensor s_i relative to the pivot point p_k can be established as:

$$s_i = [x_i \quad y_i] = p_k + \vec{p_k s_i} = p_k + \frac{1}{\alpha_k} R\left(\frac{\pi}{2} - \theta_i\right) \Delta^i s_i \tag{11}$$

p_k, α_k and θ_i can be solved by imposing the rigid body and matching pivot point conditions on the displacements measured. To apply these conditions the pivoting experiment has to be conducted for multiple pivot points p_k ($k = 1, \ldots m$), with their locations distributed in a polygonal arrangement over the robot frame. The pivot angle α_k at which the frame is rotated in each pivot experiments should span π minus the internal angle α_i between lines connecting the actual pivot point p_k with the preceding and following pivot points p_{k-1} and p_{k+1}. Under this condition, the sum of all pivot angles is equal to 2π:

$$\alpha_m = \sum_{k=1}^{m} \alpha_k = 2\pi \tag{12}$$

4.2.1. Application of rigid body condition (RBC)

θ_i can be estimated from the RBC [23]. Ideally the RBC can be expressed by the following equation:

$$(\Delta s_i)^T \; (\vec{s_i s_j}) = (\Delta s_j)^T \; (\vec{s_i s_j}) \tag{13}$$

In practice, assume that there is a difference between the left and right hand side of Eq. (13) represented by

$$\varepsilon_{ij} = (\Delta s_i)^T \; (\vec{s_i s_j}) - (\Delta s_j)^T \; (\vec{s_i s_j})$$

which, after considering the calculations of the terms in the equation, can be expanded to:

$$\begin{aligned} \varepsilon_{ij} = \; & \frac{1}{\alpha_{k_1}} (R(-\theta_i)\Delta^i s_{i_2} - R(-\theta_j)\Delta^j s_{j_2}) \\ & \left(R\left(\frac{\pi}{2} - \theta_i\right) \Delta^i s_{i_1} - R\left(\frac{\pi}{2} - \theta_j\right) \Delta^j s_{j_1}\right) \end{aligned} \tag{14}$$

The indices i_1, i_2, j_1, j_2 and k_1 refer to displacement data acquired from sensors s_i and s_j in the experiments 1 and 2 using different pivot points k_1 and k_2. To allow for minimization of ε_{ij} to obtain estimates for θ_i and θ_j, two sets of measurement data for the displacements $\Delta^i s_{i_1}$, $\Delta^j s_{j_1}$, $\Delta^i s_{i_2}$ and $\Delta^j s_{j_2}$, from different experiments with different pivot points have to be used to prevent a trivial solution to occur.

Eq. (14) can be applied to all tuples of subsets of two pivoting experiments using different pivot points and subsets of two sensors. For the present experimental setup with four pivoting points and four sensors this implies application to 36 tuples.

In the present experimental setup, the sensors are mounted in alignment with the robot frame, meaning that the ^iY and ^rY axes as well as ^iX and ^rX axes have the same direction implying that the angle θ_i ($i = 1, \ldots, 4$) is close to zero for all sensors. This allows for further simplifications as the rotation matrix $R(\theta_i)$ can be reduced to matrix $\tilde{R}(\theta_i)$:

$$\begin{aligned} \tilde{R}(\theta_i) & = \lim_{\theta_i \to 0} \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \\ & = \begin{bmatrix} 1 & \theta_i \\ -\theta_i & 1 \end{bmatrix} \end{aligned}$$

Also higher order terms and multiplications of θ_i and θ_j can be neglected in Eq. (14), resulting in the following linear equation:

$$\varepsilon_{ij} = u \cdot \theta_i + v \cdot \theta_j + w, \quad (i, j = 1, \ldots 4; \quad i \neq j) \tag{15}$$

with:

$$\begin{aligned} u & = \Delta x_{i_2} \Delta x_{j_1} - \Delta x_{i_1} \Delta x_{j_2} \\ & + \Delta y_{i_2} \Delta y_{j_1} - \Delta y_{i_1} \Delta y_{j_2} \\ v & = -\Delta x_{i_2} \Delta x_{j_1} + \Delta x_{i_1} \Delta x_{j_2} \\ & - \Delta y_{i_2} \Delta y_{j_1} + \Delta y_{i_1} \Delta y_{j_2} \\ w & = \Delta x_{i_2} \Delta y_{i_1} - \Delta x_{j_2} \Delta y_{i_1} - \Delta x_{i_1} \Delta y_{i_2} \\ & + \Delta x_{j_1} \Delta y_{i_2} - \Delta x_{i_2} \Delta y_{j_1} + \Delta x_{j_2} \Delta y_{j_1} \\ & + \Delta x_{i_1} \Delta y_{j_2} - \Delta x_{j_1} \Delta y_{j_2} \end{aligned}$$

where all terms Δx_i Δy_i and Δx_j Δy_j are displacements measured by the sensors in their $^iX - ^iY$ and $^jX - ^jY$ coordinate frames. The superscripts i and j have been omitted for simplicity in the notation. With Eq. (15), the RBC is applied by finding values for θ_i and θ_j that minimize ε_{ij}. The solution for $\varepsilon_{ij} = 0$ is approximated through linear regression similar to Eq. (6), giving best estimates $\hat{\theta}_i$ and $\hat{\theta}_j$ for θ_i and θ_j:

$$[\hat{\theta}_i \quad \hat{\theta}_j]^T = -([u \quad v]^T \quad [u \quad v])^{-1} \quad [u \quad v]^T \quad w \tag{16}$$

4.2.2. Application of matching pivot point constraints (MPPC)

To derive α_k from the displacements measured by the mouse sensors, at least three pivot experiments have to be conducted, each with a different pivot point. Under these conditions, Eq. (12) applies, providing one constraint to calculate the value of α_k, ($k = 1, \ldots, m$).

With the orientations of the sensors in the robot frame θ_i known, more expressions to estimate α_k can be derived from the matching pivot point constraint (MPPC). The principle of MPPC is illustrated in Fig. 6 and simply states that two different paths between the same pair of pivot points must have the same begin and end point:

$$\vec{p_k s_i} - \vec{p_l s_i} = \vec{p_k s_j} - \vec{p_l s_j} \tag{17}$$

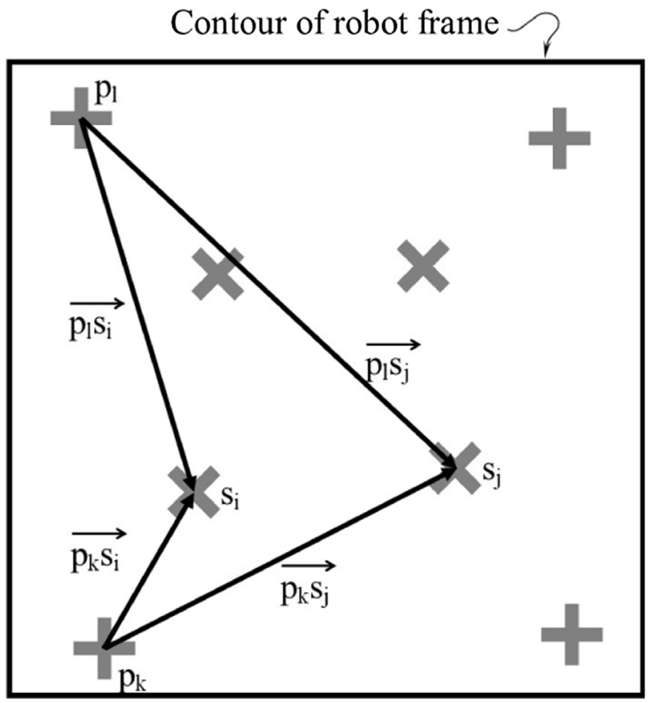

Fig. 6. Illustration of the Matching Pivot Point Constraint (MPPC).

In case of m pivot points and n sensors, there are $\binom{n}{2}$ pairs of paths between a pair of pivot points and there are $\binom{m}{2}$ different pairs of pivot points. Hence, the number of MPPC equations according to Eq. (17) is $\binom{n}{2} \cdot \binom{m}{2}$.

For the present setup described in Section 3, with four sensors and four pivot points there are 36 equations defining MPPC. The terms $\vec{p_k s_i}$, $\vec{p_l s_i}$, $\vec{p_k s_j}$ and $\vec{p_l s_j}$ are expressed by Eq. (10) and comprise the unknowns α_k and α_l for the pivoting movements around point p_k and point p_l respectively.

To solve the pivot angles α_l and α_k, $(l, k = 1, \ldots, m, l \neq k)$, ideally the MPPC equations can be expressed in the following compact form:

$$m_c \ \ \alpha^{-1} = 0$$

where:

$$m_c = \begin{array}{ccccc} k=1 & k=2 & \ldots & k=i & k=m \\ \end{array}$$
$$m_c = \begin{pmatrix} a_{11} - a_{21} & a_{22} - a_{12} & \ldots & 0 & 0 \\ a_{11} - a_{21} & 0 & 0 & a_{2i} - a_{1i} & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & a_{ni} - a_{oi} & a_{om} - a_{nm} \end{pmatrix}$$

$\Delta^i s_{ik}$ is the displacement measured by sensor s_i in the turn around pivot point p_k. In the practical measurements, α^{-1} can be solved by numerical minimization of the norm of $m_c \ \alpha^{-1}$ with α_k as parameters and within constraint (12). The solution $\hat{\alpha}_k$ producing the minimal norm $|m_c \ \alpha^{-1}|$ is the optimal estimator for α_k.

Setting the location of pivot point p_1 to $(x_{p_1}, y_{p_1}) = (0, 0)$, the positions of sensors s_i $(i = 1, \ldots, n)$ and the other pivot points p_k $(k = 2, \ldots, m)$ relative to p_1, can be calculated from Eqs. (11) and (17).

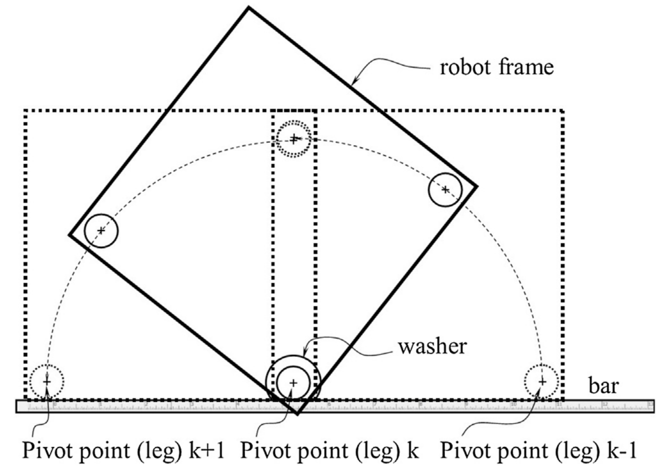

Fig. 7. Top view of the execution of the pivoting experiments.

5. Calibration experiments

With the setup for the mouse sensors as described in Section 3, displacement data from the four mouse sensors are recorded and stored for off-line processing in the experiments conducted in the following two-step procedure:

In the first step, the frame slides 10 times up and down between two stop blocks. The blocks are set at a distance that allows the robot frame to travel 1 m. between them. This is the only measurement applied in the complete calibration procedure. The frame is moved by hand between the blocks with one side of the frame kept against a straight wooden bar fixed on a surface, acting as a guide rail as shown in Fig. 2. The motion is repeated in four orientations of the robot frame each time with another side against the bar. This experiment has been conducted on three different surfaces: a cast concrete floor, a PVC tabletop of a folding table and a smooth wooden tabletop with white matte finish, in which over 80 m of optical flow data was collected on each surface.

In the second step the frame is pivoted around one of its legs acting as center of rotation, with the leg held in place in a washer (ring), fixed to the surface. The washer is of a size that keeps the rounded cap under the pivot leg centered without play. The suspension in the washer constrains the motion of the pivot leg to pure rotation around its own center.

A bar fixed over the surface acts as a limiter for the pivot motion, as illustrated in Fig. 7. The motion starts with legs k and $k-1$ against the bar. With leg k in position p_k acting as pivot leg, the frame is manually pushed around p_k until leg $k+1$ touches the bar after which the frame is pushed back to its starting position with legs k and $k-1$ against the bar. The motion is repeated 10 times for each leg acting as pivot leg ($k = 1, \ldots, 4$). This experiment was performed on the tabletop at which optical flow data was collected over a distance of 26.8 m.

5.1. Calibration of sensitivities

The data collected in the first calibration step are processed according to Eq. (7). The resulting sensitivities are listed in Table 2 together with the standard deviation expressed as a percentage of the average value of the sensitivity, in parenthesis. The following observations can be made:

Firstly, in spite of placement of the sensor according to the manufacturers instructions, the sensitivities measured exhibit considerable deviation from the nominal sensitivity of 1600 dpi specified, corresponding to 62,992 ppm. So the standard resolu-

Table 2
Calibration results of sensor sensitivities on different surfaces in pixels per meter [ppm].

Surface	f_1 [ppm]	f_2 [ppm]	f_3 [ppm]	f_4 [ppm]
Concrete floor	62, 466 (0.02 %)	63, 174 (0.01 %)	72, 649 (0.03 %)	66, 363 (0.02 %)
PVC table top	63, 672 (0.02 %)	64, 521 (0.01 %)	75, 842 (0.02 %)	67, 967 (0.01 %)
Smooth table top	64, 625 (0.20 %)	64, 282 (0.14 %)	74, 871 (0.34 %)	67, 685 (0.19 %)
Average sensitivity.	63, 588 (1.39 %)	63, 992 (0.92 %)	74, 454 (1.79 %)	67, 338 (1.04 %)

Table 3
Calibration results of sensor orientation θ_{s_i} and pivot angle α_k.

k, i	1	2	3	4
θ_{s_i} [rad]	$12.63\ 10^{-3}$	$-8.57\ 10^{-3}$	$2.56\ 10^{-3}$	$-6.62\ 10^{-3}$
α_k [rad]	32.15	31.29	31.10	31.13

tion provided by the manufacturer cannot be used to calculate the displacements of the sensors accurately.

Secondly, the standard deviation is smaller on rough surfaces, like the concrete floor and the PVC table, which produce more features in the image of the surface picked up by the sensor.

Thirdly, the sensitivity is different for each surface. The figure listed for the average sensitivity over all three surfaces in Table 2, has a corresponding standard deviation that is much larger that for the other surface separately. Obviously using one value for the sensitivity on all surfaces impairs the accuracy of the displacement measurement.

5.2. Calibration of sensor orientation and location

Sensor orientations and locations are calibrated with the data collected in the pivot experiments in the second step of the calibration procedure. The optical flow data Δs_{p_i} [pixels] are converted to displacements Δs_i [meter] using the calibrated sensitivities from the previous step:

$$\Delta s_i = \frac{\Delta s_{p_i}}{f_i} [\text{meter}]$$

θ_i can be estimated using the expression for linear regression in Eq. (16). The results in Table 3 exhibit very small orientation angles, as is to be expected for sensors mounted in alignment with the robot frame. The very small values for θ_i provide justification for the use of the simplified rotation matrix $\tilde{R}(\theta_i)$ and elimination of higher order and product terms for θ_i and θ_j in Eq. (15).

Subsequently, the pivot angles α_k can be estimated using the estimated values for θ_i. The pivot angles are needed in the calculation of the positions of pivot points and sensors. Through the minimization of the Euclidean norm of $m_c \alpha^{-1}$, $\hat{\alpha}_k$ is established as listed in Table 3. The constraint applied to the minimization according to Eq. (12) takes into account that the data collected is the accumulation of the displacement data of all 10 sweeps of the robot frame: $\alpha_m = 2 \cdot 10 \cdot 2\pi = 40\pi$. The sum of the calculated pivot angles in Table 3 matches this amount well. With the figures for $\hat{\alpha}_k$, $(k = 1, \ldots, 4)$, the vectors $p_k s_i$ connecting all pivot points p_k with all sensors s_i can be computed using the expression in Eq. (10).

Defining $p_1 = (0, 0)$ as the point of reference, the locations of all other pivot points p_k $(k = 2, 3, 4)$ are computed as the average value of the locations determined by the vector additions: $p_k = p_1 s_i - p_k s_i$, $(k = 2, 3, 4;\ i = 1, \ldots, 4)$:

$$p_k = \sum_{i=1}^{4} (p_1 s_i - p_k s_i)/4 \quad (k = 2, 3, 4)$$

Lastly, the sensor positions are calculated as the average values of the vector additions defined by Eq. (11):

$$s_i = \sum_{k=1}^{4} (p_k + p_k s_i)/4 \quad (i = 1, \ldots, 4)$$

A geometric center of the robot frame c is defined at the intersection of the diagonal between sensor pairs $[\bar{s}_1\ \ \bar{s}_3]$ and $[\bar{s}_2\ \ \bar{s}_4]$. Table 4 gives an overview of the location of pivot points and sensors relative to c as established in the calibration procedure described, compared with the figures from the CAD design and locations estimated from the hand measured figures.

6. Validation of the mobile robot path

With the setup calibrated for the sensitivity of the sensors and the locations of the pivot points and positions and orientations of the sensors in the robot frame, the effects of calibration on the measurement of the mobile robot displacement can be validated through computation of the path of the robot frame.

The path is reconstructed by first order integration of the incremental displacements of the center of the robot frame $\Delta c(t) = [\Delta x_c(t)\ \Delta y_c(t)\ \Delta\theta_c(t)]$ where t represents the time interval at which the displacement occurs. $\Delta c(t)$ is estimated through Eq. (6).

The integration process is computed by:

$${}^w\theta_c(t) = {}^w\theta_c(t-1) + \Delta\theta_c(t)$$

$$\begin{bmatrix} {}^w x_c(t) \\ {}^w y_c(t) \end{bmatrix} = \begin{bmatrix} {}^w x_c(t-1) \\ {}^w y_c(t-1) \end{bmatrix} + R({}^w\theta_c(t)) \begin{bmatrix} \Delta x_c(t) \\ \Delta y_c(t) \end{bmatrix}$$

$R({}^w\theta_c(t))$ is the transformation matrix for the rotation from robot to world coordinates.

Fig. 8 depicts the computed path of the robot center over 10 return sweeps for all four pivot experiments. In each experiment a different leg of the frame is used as pivot point. The robot center travels approximately 6.7 m over each path, without external correction or resetting of its position. The top picture frame detailing the center of the graph shows the robot path at the start and return position of the sweeps. As is to be expected without correction of the position, there is some drift away from the starting point of the robot center, which is the supposed point of return after each sweep. There is no noticeable drift in the orientation of any of the paths however. Even after 10 return sweeps the computed orientation of the paths appears to be highly accurate.

The leg of the robot frame acting as the center of rotation in each experiment is suspended at the pivot point in a washer fixed on the surface as explained in Section 5. Physically, no translation of the pivot leg can occur. This certainty can be used to rate the accuracy of the robot position as measured with the mouse sensors, using the parameters in Table 4. Any translation of the pivot leg observed from the optical flow data, represents an error that provides an objective measure for the accuracy of the computed displacement of the robot. The location of the pivot leg in world coordinates ${}^w p_k =$

Table 4
Overview of the locations of pivot legs p_k and sensors s_i relative to the robot center c.

k,i	1	2	3	4
Calibrated positions				
p_k [m]	(−0.1530, −0.1518)	(−0.1549, 0.1628)	(0.1573, 0.1627)	(0.1576, 0.1556)
s_i [m]	(−0.1107, −0.1086)	(−0.1132, 0.1140)	(0.1155, 0.1133)	(0.1109, 0.1117)
CAD positions				
p_k [m]	(−0.1565, −0.1565)	(−0.1565, 0.1565)	(0.1565, 0.1565)	(0.1565, −0.1565)
s_i [m]	(−0.1155, −0.1155)	(−0.1155, 0.1155)	(0.1155, 0.1155)	(0.1155, −0.1155)
Pivot leg positions estimated from hand-measurements				
p_k [m]	(−0.1558, −0.1569)	(−0.1581, 0.1569)	(0.1557, 0.1568)	(0.1581, −0.1569)

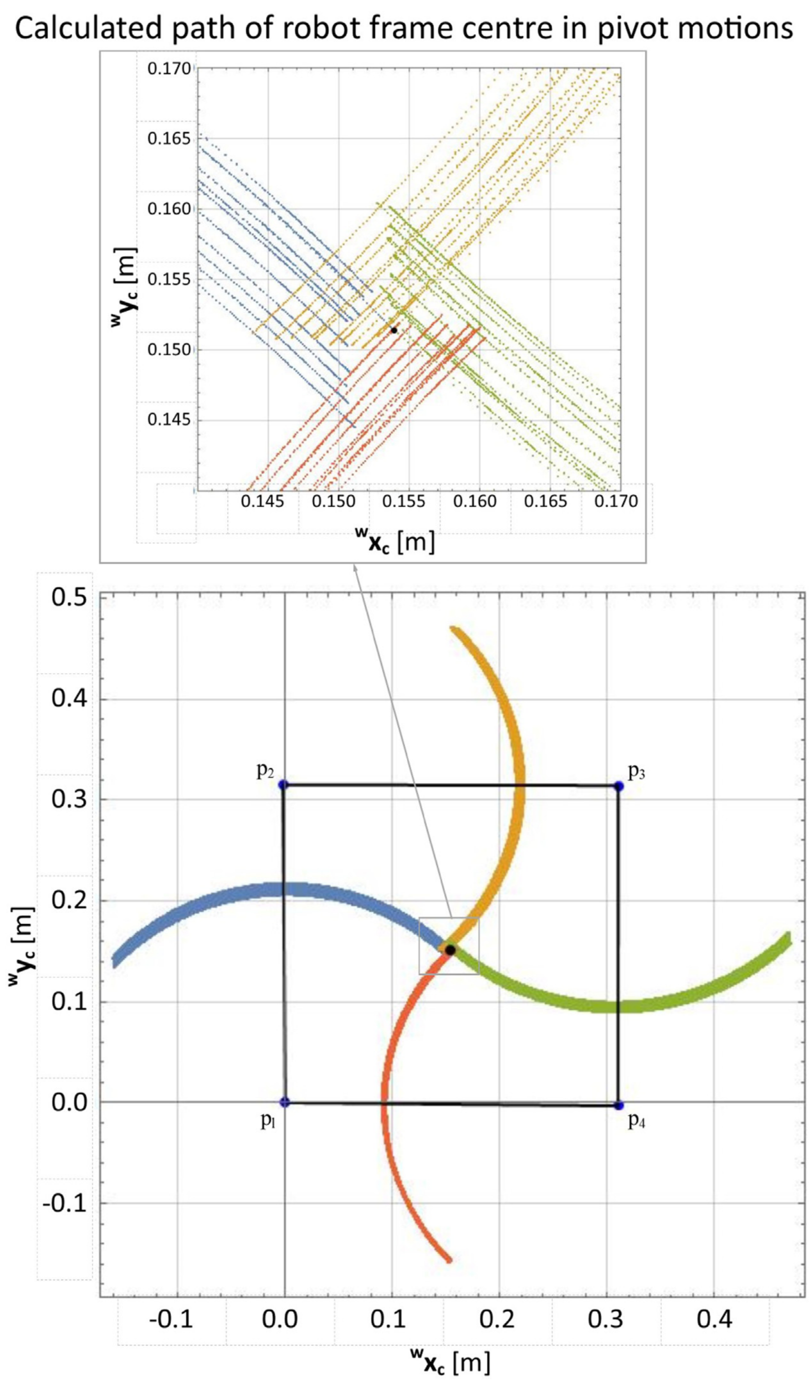

Fig. 8. Paths of the centre of the robot frame in four pivot experiments.

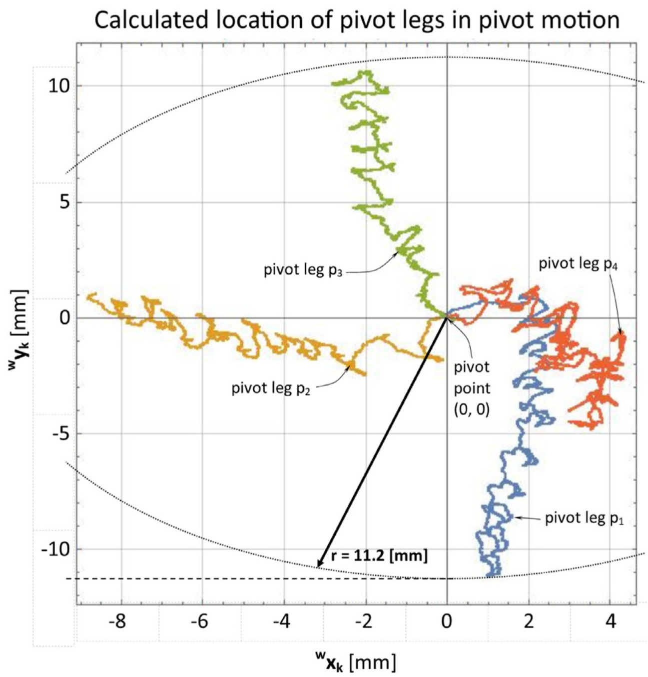

Fig. 9. Calculated position of the pivot legs ${}^w p_k$ relative to the fixed pivot point, for each of the four pivot experiments.

Table 5
Deviation D_p^2 computed for different calibration scenario's.

Included calibration Scenario:	1	2	3	4
Sensor sensitivity f_s	☐	☑	☑	☑
Locations of pivot points p_k and sensors s_i	☐	☐	☑	☑
Sensor orientation θ_{s_i}	☐	☐	☐	☑
$D_p^2 = \sum_{k=1}^{4} \sum_t {}^w p_k^2$ [m^2]	9.63	8.89	2.17	2.17

$({}^w x_k, {}^w y_k)$ can be derived from the robot frame center location ${}^w c = ({}^w x_c, {}^w y_c)$ and orientation ${}^w\theta_c$ with the expression:

$$\begin{bmatrix} {}^w x_k(t) \\ {}^w y_k(t) \end{bmatrix} = \begin{bmatrix} {}^w x_c(t) \\ {}^w y_c(t) \end{bmatrix} + R({}^w\theta_c(t)) \begin{bmatrix} x_k \\ y_k \end{bmatrix}$$

Fig. 9 frames the calculated location of the pivot leg ${}^w p_k$ for each step, relative to its starting point in the origin of the graph representing the actual pivot point. The figure comprises the graphs for the location of each of the pivot legs used in the experiments. Obviously and unavoidably, the pivot legs drift away from the pivot point, up to a maximum distance of 0.011 m, while the center of the robot frame travels over 6.7 m distance. Expressed as a dimensionless figure the drift amounts to $0.011/6.7 \cdot 100\,\% = 0.16\%$, which is a very satisfactory result for dead reckoning and significantly better than the 0.28% error reported in [23], where double the number of optical mouse sensors was used (eight sensors). To evaluate the effectiveness of the calibration procedure applied, the displacement of the pivot leg ${}^w p_k$ is squared and accumulated over all integration steps t of the pivoting procedure for all four pivoting experiments: $D_p^2 = \sum_{k=1}^{4} \sum_t {}^w p_k^2$.

The deviation D_p^2 can be used to compare the accuracy of the position established with computer mouse sensors for different sensor parameters and data processing methods. A lower value for D_p^2 indicates a more accurate computation of the robot position. Table 5 lists the deviation D_p^2 for different calibration scenario's, from not calibrated, to fully calibrated according to the procedure described in this paper. Comparison of the calibration scenario's demonstrates that calibration of the sensor sensitivities and in particular calibration of the location of pivot points and sensors strongly reduce the value of D_p^2. These results validate the significance and effectiveness of the calibration procedure presented.

7. Conclusions

A new approach for implementation and calibration of optical mouse sensors for applications to odometry of mobile robots is proposed. The calibration method determines the orientation and position of the sensors in separate computations from the same data set. The data set comprises of displacement readings from the mouse sensors itself, acquired in a simple experiment in which the robot frame is rotated around pivot points. The certainty that the position of the pivot point is fixed in the experiments allows for validation of the results of the calibration, without additional measurements. This use of a pivot point defines a basis for assessment of the accuracy of the displacement measurement that can be used to compare the performance of different robot localization systems.

The pivoting experiments are preceded by a single measurement of the sensitivity of each sensor by collection of sensor readings in a move without rotation of the robot over a known distance.

Neither calibration of the sensor's sensitivities nor their location and orientation in the robot frame requires any a-priori knowledge about the positions of the sensors or pivot points. The setup used to collect the measurement data from the mouse sensors is designed to enable synchronized acquisition of the displacement measurements from all sensors simultaneously.

After completion of the full calibration procedure, the setup developed using four computer mouse sensors is capable of accurate odometry: the computed position of the pivot point of the robot frame deviated 0.16% of the total distance travelled (6.7 m). This is a significant improvement over the next best result of an accuracy of 0.28% published in the open literature. Moreover, the simple experimental procedure and usage of readings from the mouse sensors only, strongly facilitate implementation of the method in a self-calibration operation.

Conflicts of interest

The authors declare no conflicts of interest.

Acknowledgment

The authors want to thank Jian Huang, John Collins and Akhil Anand from Auckland University of Technology for their support.

References

[1] J. Borenstein, H.R. Everett, L. Feng, D. Wehe, Mobile robot positioning: sensors and techniques, J. Robot. Syst. 14 (4) (1997) 231–249, http://dx.doi.org/10.1002/(SICI)1097-4563(199704)14:4<231::AID-ROB2>3.0.CO;2-R.
[2] K.S. Chong, L. Kleeman, Accurate odometry and error modelling for a mobile robot, Proceedings of International Conference on Robotics and Automation, vol. 4 (1997) 2783–2788, http://dx.doi.org/10.1109/robot.1997.606708.
[3] J. Borenstein, H. Everett, L. Feng, et al., Where Am I? Sensors and Methods for Mobile Robot Positioning, vol. 119(120, University of Michigan, 1996, pp. 27.
[4] E. Eitel, Basics of rotary encoders: overview and new technologies, Mach. Des. Mag. 7 (2014).
[5] M. Shneier, R. Bostelman, Literature Review of Mobile Robots for Manufacturing, 2015, http://dx.doi.org/10.6028/nist.ir.8022, Tech. Rep., May.
[6] A. Martinelli, R. Siegwart, Estimating the odometry error of a mobile robot during navigation, in: European Conference on Mobile Robots (ECMR2003), Warsaw Poland, September 4–6, no. LSA-CONF-2003-005, 2003.
[7] T. Ng, The optical mouse as a two-dimensional displacement sensor, Sens. Actuators A: Phys. 107 (1) (2003) 21–25, http://dx.doi.org/10.1016/s0924-4247(03)00256-5.

[8] M. Tresanchez, T. Pallejà, M. Teixidó, J. Palacín, The optical mouse sensor as an incremental rotary encoder, Sens. Actuators A: Phys. 155 (1) (2009) 73–81, http://dx.doi.org/10.1016/j.sna.2009.08.003.
[9] A. Bonarini, M. Matteucci, M. Restelli, A kinematic-independent dead-reckoning sensor for indoor mobile robotics, 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566) (2004), http://dx.doi.org/10.1109/iros.2004.1389998.
[10] J. Palacin, I. Valgañon, R. Pernia, The optical mouse for indoor mobile robot odometry measurement, Sens. Actuators A: Phys. 126 (1) (2006) 141–147, http://dx.doi.org/10.1016/j.sna.2005.09.015.
[11] U. Minoni, A. Signorini, Low-cost optical motion sensors: an experimental characterization, Sens. Actuators A: Phys. 128 (2) (2006) 402–408, http://dx.doi.org/10.1016/j.sna.2006.01.034.
[12] T. Ng, K. Ang, The optical mouse for vibratory motion sensing, Sens. Actuators A: Phys. 116 (2) (2004) 205–208, http://dx.doi.org/10.1016/j.sna.2004.04.009.
[13] D.-H. Yi, T.-J. Lee, D.-I. Cho, Afocal optical flow sensor for reducing vertical height sensitivity in indoor robot localization and navigation, Sensors 15 (5) (2015) 11208–11221, http://dx.doi.org/10.3390/s150511208.
[14] D. Sekimori, F. Miyazaki, Self-localization for indoor mobile robots based on optical mouse sensor values and simple global camera information, 2005 IEEE International Conference on Robotics and Biomimetics – ROBIO (2005) 605–610, http://dx.doi.org/10.1109/robio.2005.246337.
[15] S. Bell, High-precision robot odometry using an array of optical mice, 2011 IEEE Reg 5 Stud Pap Contest (2011), March.
[16] D.-H. Yi, T.-J. Lee, D.-I. Cho, A new localization system for indoor service robots in low luminance and slippery indoor environment using afocal optical flow sensor based sensor fusion, Sensors 18 (2) (2018) 171, http://dx.doi.org/10.3390/s18010171.
[17] P.-L. Wu, S.-L. Jeng, W.-H. Chieng, Least squares approach to odometry based on multiple optical mouse sensors, 2010 5th IEEE Conference on Industrial Electronics and Applications (2010), http://dx.doi.org/10.1109/iciea.2010.5514712.
[18] S. Kim, H. Kim, Systematic robustness analysis of least squares mobile robot velocity estimation using a regular polygonal optical mouse array, Asian J. Control 14 (2) (2011) 348–358, http://dx.doi.org/10.1002/asjc.454.
[19] M. Cimino, P.R. Pagilla, Optimal location of mouse sensors on mobile robots for position sensing, Automatica 47 (10) (2011) 2267–2272, http://dx.doi.org/10.1016/j.automatica.2011.08.004.
[20] S. Kim, H. Kim, Optimal optical mouse array for accurate mobile robot velocity estimation, 2013 10th IEEE Interational Conference on Networking Sensing and Control (ICNSC) (2013) 83–88, http://dx.doi.org/10.1109/icnsc.2013.6548715.
[21] S. Kim, Isotropic optical mouse placement for mobile robot velocity estimation, 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2013), http://dx.doi.org/10.1109/aim.2013.6584060.
[22] S. Kim, H. Kim, Optimal placement of optical mice for accurate mobile robot velocity estimation, Int. J. Control Autom. Syst. 12 (4) (2014) 861–869, http://dx.doi.org/10.1007/s12555-012-0462-x.
[23] J.-S. Hu, Y.-J. Chang, Y.-L. Hsu, Calibration and on-line data selection of multiple optical flow sensors for odometry applications, Sens. Actuators A: Phys. 149 (1) (2009) 74–80, http://dx.doi.org/10.1016/j.sna.2008.10.003.
[24] A.C. Rencher, W.F. Christensen, Methods of Multivariate Analysis, John Wiley & Sons, Inc., Hoboken, New Jersey, 2012, http://dx.doi.org/10.1002/9781118391686.
[25] Adns 9800 Data Sheet, Pixart Imaging Inc., Taiwan, 2012.

Biographies

Tony Paijens graduated with a Master degree in Mechanical Engineering from Delft University of Technology (TU-Delft) in the Netherlands in 1987. He worked as a process engineer at Akzo Nobel from 1987 to 1990, after which he lectured at TU-Delft from 1990 to 2006. Subsequently he ventured into entrepreneurship in the design, construction and operation of supercritical extraction plants in South East Asia over a period of eight years. Presently he pursues his PhD at Auckland University of Technology. His interest is making and using robotic technologies for manufacturing.

Loulin Huang is an Associate Professor of Mechatronic Engineering in Auckland University of Technology He had about thirty years' research experience in the areas of robotics, mechatronics and control. He has completed more than thirty industrial and government-funded projects in robotics and mechatronics, and published about 100 papers and two books. The systems developed from his projects include international award winning mobile robot systems for soccer games and an automatic road texture depth monitoring system which has been used in the road projects in New Zealand and Australia.

Ahmed Al-Jumaily is currently a Professor of Biomechanical Engineering at the Institute of Biomedical Technologies at the Auckland University of Technology, New Zealand. He is a member of 11 international professional societies. He is the Editor-in-Chief of the ASME Journal of Engineering and Science in Medical Diagnostics and Therapy and of the ASME monograph series-Biomedical and Nanomedical Technologies. Published more than 350 papers in international journals and conference proceedings in Biomedical applications, system dynamics, vibrations and control.