# Estimation of Mobile Robot Pose from Optical Mouses

Lenka Mudrová, Jan Faigl, Jaroslav Halgašík, and Tomáš Krajník

The Gerstner Laboratory for Intelligent Decision Making and Control,
Department of Cybernetics, Faculty of Electrical Engineering
Czech Technical University in Prague
{mudrol1,halgajar}@fel.cvut.cz,
{xfaigl,tkrajnik}@labe.felk.cvut.cz

**Abstract.** This paper describes a simple method of dead-reckoning based on off-the-shelf components: optical mouses and a laptop. The problem is formulated as finding a transformation of mouses positions to position of the robot. The formulation of the transformation is based on a method already used in range-based localization. Beside a solution of the transformation, the paper provides description of practical application of mouse based localization for a home made robot. The paper considers identification and mouse data reading procedures as well. The presented approach has been evaluated in several real experiments and the proposed localization provides competitive results to the odometry based on high-precision stepper motors.

## 1 Introduction

An optical computer mouse used to control a cursor on the screen can inspire to create an open-loop system, which estimates position of a mobile robot. The mouse provides local changes of its sensor position in two directions, so it is clear that estimation of robot pose (position and orientation) cannot be based just on one mouse. Either two or more mouses have to be used or additional information (robot kinematic constrains, compass data) must be taken into account. An application of the mouse to the self-localization problem of a mobile robot came out almost immediately with introduction of an optical mouse. Since then the problem has been studied for several years and successful applications of mouses-based robot self-localization have been reported by many authors. The basic problem formulation for one mouse can be based on a velocity equation [4]. If several mouses are used, detection of wrong measurements can be considered [9]. Precision of localization can also be increased by combination with other sensors [8]. The precision of the optical mouse depends on a surface on which a mouse is moved. Authors of [6] examined several surfaces and reported problems of non-straight displacements. To increase precision of pose estimation, several mouses can be combined and rigid-body constraints can be considered to verify consistence of data measurements [2].

D. Obdržálek and A. Gottscheber (Eds.): EUROBOT 2010, CCIS 156, pp. 93–107, 2011.

In this paper we formulate the self-localization as a problem of finding transformation between two robot positions. The transformation is composed from rotation and translation, which is principally same as a method already used in the aforementioned papers. The main difference of the proposed solution resides in computation of the transformation parameters that does not require calculation of inverse (pseudo-inverse) matrix, which can suffer from numerical issues. Our formulation leads to a solution already used in localization methods that are based on range measurements. It allows straightforward combination of several mouses as well as detection of possibly wrong measurements that are discarded. To our best knowledge such approach has not been applied for optical mouses before. An additional contribution of this paper is a presentation of a practical approach, which shows how off-the-shelf components can be combined in a simple and straightforward manner in order to create a cheap independent localization system for a mobile robot. Such a system can be used in EUROBOT competitions, because the playfield is typically flat and uniform, which is suitable for an optical mouse.

The rest of the paper is organized as follows. The next section briefly describes basic principle of mouse sensors and used optical mouses. Problem formulation and pose estimation equation are presented in Section 3. A procedure for identification of sensor parameters is described in Section 4. Practical application of the procedure and realization of mouse based localization system are considered in Section 5, where possible issues are described. Experimental results are presented in Section 6. Described methods and results are discussed and future work is proposed in Section 7.

## 2 Basic Principle of Optical Mouse Sensor and Used Optical Mouses

The main function of computer mouse is measuring translations in two orthogonal axes. A basic scheme of the optical mouse principle is depicted in Fig. 1. The method is based on correlation of two consecutive images captured by an optical sensor, which is principally low-resolution camera (tens of pixels). Camera resolution together with its optics defines a visible area by the sensor. Two consecutive images have to overlap, otherwise a correlation between two consecutive images cannot be found and the traveled translations are not estimated. This limits maximal measurable speed of a mouse. Precision of a particular mouse is affected by resolution of the sensor and capability to recognize surface details. Surface details recognition can be enhanced by proper lighting, which is usually provided by a light or laser emitting diode. Mouses with laser diodes usually provides higher precision, therefore they are more suitable for mobile robot localization.

Regarding precision, price and size, we decided to use four "Genius NetScroll+ Mini Traveler laser" mouses for our localization system. The mouses use the PAW3601DH-NF optical sensor that provides resolution of 1600 counts per inch and maximal speed of 28 inches per seconds (around 0.7 $\mathrm{m.s^{-1}}$) with acceleration

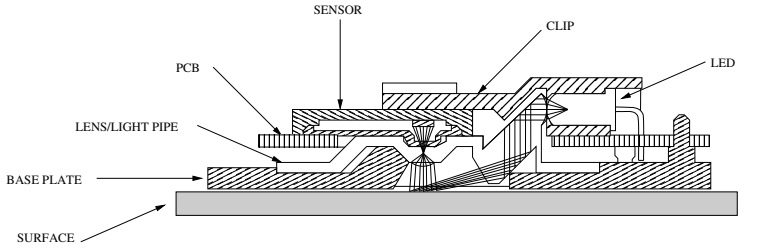



**Fig. 1.** Mechanism of optical mouse

rates up to 20g. The sensor contains a DSP chip that operates at 27 MHz and processes 6600 frames per second. The mouse can be plugged to USB interface and it does not require a special driver in commonly used operating systems.

## 3 Estimation of Robot Pose - Dead Reckoning

An optical mouse provides information about position changes in two orthogonal directions defined by the mouse sensor. The changes are denoted as $\Delta x_m$ and $\Delta y_m$. A summation of these changes can be used to estimate position of the mouse. Thus if a mouse is moved from position $(0, 0)$ then after several mouse movements (without rotation) the position of the mouse can be estimated as $(\sum_{t=1}^{n} \Delta x_m(t), \sum_{t=1}^{n} \Delta y_m(t))$, where $n$ is the number of position changes sent by the mouse. In order to estimate position of a robot together with its orientation several mouses can be used. Beside the approach described in [8] the basic equation can be formulated in a slightly different way leading to a problem formulation that can be solved by another localization approach called Iterative Closest Point (ICP) [5]. The idea is based on consideration of optical mouse sensor output as a measurement of a position of a point in a plane like range measurements of laser scanners or infrared sensors. The main advantage of mouse sensors is that the correspondence problem is already solved, as data are received from particular mouse.

Assume following basic equation of the robot movement composed of turning about an angle $\omega$ and translation about a vector $T$ between time interval $\langle t, t+1\rangle$:

$$\begin{aligned} \varphi(t+1) &= \varphi(t) + \omega \\ X(t+1) &= R_{\varphi(t+1)}T + X(t) \end{aligned} \tag{1}$$

where $\varphi$ denotes orientation of the robot, $X$ denotes position of the robot with respect to global coordinate system, $R_{\varphi(t+1)}$ is the rotation matrix with the angle $\varphi(t+1)$. The problem is to find values of $\omega$ and $T$ from particular movements of mouses.

### 3.1 Robot Movement

To describe the idea of the robot pose estimation assume a robot with a local coordinate system $O_r$. Two mouses are attached to the robot in general position,

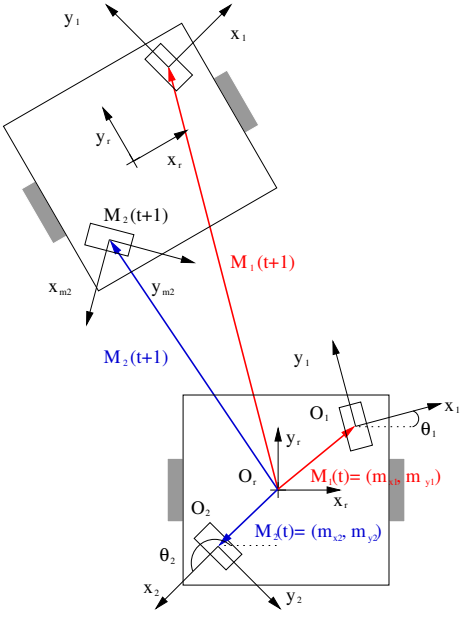



**Fig. 2.** Robot movement

see Fig. 2. The first and the second mouses define coordinate systems $O_1$ and $O_2$ respectively. The centers of the mouses are at positions $M_1(t)$, resp. $M_2(t)$ in $O_r$, i.e. coordinate system of the robot. The orientation of the mouses with respect to orientation of the robot is $\theta_1$, resp. $\theta_2$. When the robot is moved, mouses centers according to $O_r$ become $M_1(t+1)$ and $M_2(t+1)$. The mouse provides changes of position according to mouse coordinate system, thus position of mouse center $i$ can be estimated by

$$M_i(t+1) = M_i(t) + R_{\theta_i} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}. \tag{2}$$

A pair of positions $M_i(t)$ and $M_i(t+1)$ is provided for each mouse and the problem is to find a common transformation (i.e. values of $\omega$ and $T$) between points in the pairs. For $k$ mouses a quadratic criterion can be formulated:

$$E(\omega, T) = \sum_{i=1}^{k} |R_\omega M_i(t) + T - M_i(t+1)|^2, \tag{3}$$

which is exactly the same criterion as in the ICP algorithm [5]. A solution of Eq. (3) can be found in an analytical form [5]:

$$\begin{aligned} \omega &= \arctan \frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}} \\ T &= \begin{bmatrix} \overline{x}' \\ \overline{y}' \end{bmatrix} - R_\omega \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix}, \end{aligned} \tag{4}$$

where

$$\begin{aligned} \overline{x} &= \frac{1}{k}\sum_{i=1}^{k} x_i(t), & \overline{y} &= \frac{1}{k}\sum_{i=1}^{k} y_i(t), \\ \overline{x}' &= \frac{1}{k}\sum_{i=1}^{k} x_i(t+1), & \overline{y}' &= \frac{1}{k}\sum_{i=1}^{k} y_i(t+1), \\ S_{xx'} &= \sum_{i=1}^{k}(x_i - \overline{x})(x_i(t+1) - \overline{x}'), & S_{yy'} &= \sum_{i=1}^{k}(y_i - \overline{y})(y_i(t+1) - \overline{y}'), \\ S_{xy'} &= \sum_{i=1}^{k}(x_i - \overline{y})(y_i(t+1) - \overline{y}'), & S_{yx'} &= \sum_{i=1}^{k}(y_i - \overline{x})(x_i(t+1) - \overline{x}'). \end{aligned} \tag{5}$$

The estimation of the angle $\omega$ and the translation vector $T$ can be performed for each step and new robot position $X(t+1)$ in the global coordinates is then found from Eq. (1).

### 3.2 Detection of Outliers

If more than two mouses are used, detection of possible wrong measurements can be based on evaluation of error for each particular mouse measurement - $\Delta x_m$ and $\Delta y_m$. The mouse measurement with the highest value of the error, according to Eq. (3), can be discarded and transformation can be recomputed from the remaining values. Alternatively several combinations of measurements from two or more mouses can be computed and a transformation with the lowest overall error $E(\omega, T)$ can be used to estimate the robot pose.

## 4 Parameters Identification

Each mouse attached to the robot has three basic parameters that have to be identified to use Eq. (4): orientation of the mouse $\theta_i$ and position of the center of the mouse sensor to the center of the robot $M_i = (x_i, y_i)$. Besides, a distance conversion parameter from the mouse units to the metric units should be estimated to provide position of the robot in more human intuitive fashion, e.g. in meters or inches. The following identification procedure assumes a robot with differential nonholonomic drive that is capable of straight forward movement by a given distance and rotation by a given angle.

Values of $\theta_i$ can be estimated from the forward movement of the robot by a certain distance:

$$\theta_i = \arctan \frac{\sum_{t=1}^{n} \Delta y_i(t)}{\sum_{t=1}^{n} \Delta x_i(t)}, \tag{6}$$

where $\Delta y_i(t)$, resp. $\Delta x_i(t)$, are particular measurements in time $t$ provided by the $i^{th}$ mouse.

If the traveled distance $d$ (e.g. in meters) is known then the distance conversion parameter can be estimated from the forward movement. Mouses can be placed in general orientation, therefore the traveled distance in the $y$ axis should be estimated from the rotated values:

$$\textit{counts per meter for the } i^{th} \textit{ mouse} = \frac{1}{d}\sum_{t=1}^{n}(\Delta x_i(t)\cos\theta_i - \Delta y_i(t)\sin\theta_i). \quad (7)$$

Values of $M_i$ can be estimated if a robot with the differential nonholonomic drive is turned about defined angle $\omega$. The orientation of the mouse $\theta_i$ is already known, therefore each measurement can be rotated to the robot coordinate system. Kinematic constraints of a differential drive allow following computation, because a robot is assumed to move only in a direction perpendicular to wheel axis [4]. The position of the robot is not changed during rotation, therefore for a single movement about a small angle $\Delta\omega$ positions of mouses with respect to the center of the rotation can be described by:

$$\Delta\omega \begin{bmatrix} -m_{x_i} \\ m_{x_i} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}. \quad (8)$$

For a rotation about the angle $\omega$ the positions of the mouses can be estimated from the sum of rotated particular changes:

$$\begin{aligned} m_{x_i} &= \frac{1}{\omega}\sum_{t=1}^{n}(\sin\theta_i\Delta x_i(t) + \cos\theta_i\Delta y_i(t)), \\ m_{y_i} &= -\frac{1}{\omega}\sum_{t=1}^{n}(\cos\theta_i\Delta x_i(t) - \sin\theta_i\Delta y_i(t)). \end{aligned} \quad (9)$$

Though precise identifications are presented in [6,4], the above described procedure is straightforward and relatively easy to implement. The procedure is used to estimate parameters of the mouses in experiments presented in Section 6.

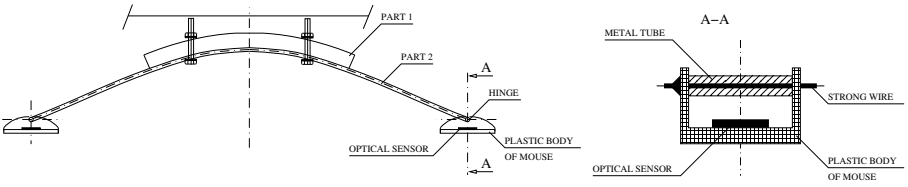
## 5 Practical Issues

This section is dedicated to practical issues that can be considered prior to development of mouse based localization system for a mobile robot. First of all, mouses are primarily used to control a cursor on the screen where a human eye-hand system provides a feedback control loop. The precision of mouse-hand-eye system can lead to an imagination that precise optical mouse can be easily used for robot localization. However, mouses-based self-localization is an open loop system incapable to correct wrong measurements and a naïve application of mouses can lead to a disappointment. This can be avoided by proper consideration of particular assumptions.

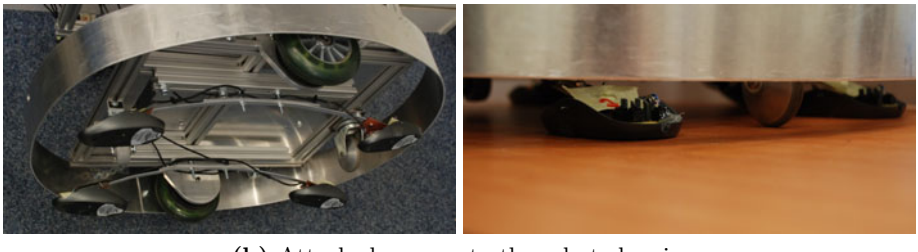
### 5.1 Mounting Mouses at the Robot

One of the fundamental constraints of the mouse application is the height of the sensor above the floor. In order to have the sensor at a constant height,

a spring mechanism should press the mouse to the floor. The pressure cannot be too high, because it have to allow the mouse to traverse small bulges. A schema of our pressure system is depicted in Fig. 3a. It is made from thin long plate of aluminium alloy. The *part 1* is made from stronger plate that is bended. It forms a supporting construction for the *part 2* that is the main spring part. The mouses are fixed by a rotating hinge with one degree of freedom. One spring holds two mouses, therefore two springs are mounted to the bottom of the robot, see Fig. 3b. Because the plastic body of the original mouse is used to hold the mouse electronic board, the sensor is at the correct height. The board has to be precisely placed in the plastic body, therefore a glue is applied to fix the board with the body.




**(a)** Schema of the pressure system



**(b)** Attached mouses to the robot chassis

**Fig. 3.** Mounting mechanism for the mouses

Although usage of four mouses have been planned for the experiments, one of the mouses was not used because of following issue. Accidentally an electronic board of the mouse was twisted and the mouse provided noisy and inaccurate data even if it was placed on the perfect surface, because of wrong height of its sensor.

### 5.2 Data Acquisition

An additional issue comes out when one tries to read raw data from the mouse. A standard operating system abstracts hardware devices. In a unix based system the raw access to the hardware device is possible by reading from the particular

device file in the `/dev` directory. In operating systems based on the Linux kernel these devices are located in `/dev/input/mouseX`, where `X` denotes particular connected mouse. The USB mouses are processed by the *mousedev* driver, which provides emulation of the PS/2 mouse protocol[1]. The protocol uses a sequence of 3 bytes in which particular changes in the $x$ and $y$ directions (i.e. $\Delta x_m$ and $\Delta y_m$) are stored in one signed byte. That means values from -127 to 127 are returned. These values are provided only if the mouse is moved.

If a laptop (or a netbook) is used as the robot on-board computer, the robot movement will cause movement of the cursor in a graphical computer desktop environment, because mouses are typically used to control the cursor. The newly added mouse to the system is automagically attached to the running graphical desktop environment. If the `hal` daemon is used as a device manager, undesirable cursor movement can be avoided by a policy rule to remove the mouse X-driver, see Listings 1.1. Such a rule can be stored in the file and placed into `/etc/hal/fdi/policy` directory.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<deviceinfo version="0.2">
   <device>
      <match key="info.product" contains="Genius Laser Mouse">
         <remove key="input.x11_driver"></remove>
      </match>
   </device>
</deviceinfo>
```

**Listing 1.1.** An example of policy rule to disable mouse movement of the cursor

Despite the fact that all mouses should provide measurements at the same moment, so the mouses can be read sequentially, the `poll` function, which examines a set of file descriptors for an activity, can be used to notify the program about new ready measurement. High speed of nowadays computers causes the `poll` function is return and data reading is performed only for one mouse at a time. The method described in Section 3 is not suitable for computing robot position change from a single mouse measurement. Therefore data from all mouses are collected before performing position estimation. Alternatively we can wait a certain time period, while particular changes of the mouse positions are accumulated. An example of the reading program is depicted in Listings 1.2, where the real timer is used to notify the program by the signal `SIGALRM` that data should be processed.

[1] The driver can be switch into ImPS/2 or EXPS/2 protocol by sending special magic sequence to the device that can be found in the kernel source in the file `mousedev.c`.

```
struct pollfd ufdr[nbr_mices];
struct itimerval timer;
for (int i = 0; i < nbr_mices; i++) {
    ufdr[i].fd = mouse[i];
    ufdr[i].events = POLLIN | POLLRDNORM;
}
timer.it_interval.tv_sec = period / 1000; // period is in ms
timer.it_interval.tv_usec = (period % 1000) * 1000;
timer.it_value.tv_sec = period / 1000;
timer.it_value.tv_usec = (period % 1000) * 1000;
signal(SIGALRM, alarm_handler);
setitimer(ITIMER_REAL, &timer, 0);
while (!quit) {
    if (poll(ufdr, nbr_mices, 10)>0) {//poll with 10 ms
        timeout
        for (int i = 0; i < nbr_mices; i++) {
            if (ufdr[i].revents & (POLLIN | POLLRDNORM)) {
                read(mouse[i])
    } } }
    if (timer_expires) {
        timer_expires = false;
        process_data(mouse, nbr_mices);
        signal(SIGALRM, alarm_handler);
        setitimer(ITIMER_REAL, &timer, 0);
} } //end read loop
```

**Listing 1.2.** An example of mouse readings

The period can affect precision of the pose estimation, that is why several experiments have been performed to verify precision of the localization for different values of period.

### 5.3 Precise Rotation of the Robot

The identification procedure described in Section 4 assumes that the robot is moved in a forward direction by a given distance and it is rotated by the given angle. The traveled distance can be measured manually with sufficient precision, but precise robot rotation is problematic. The following procedure has been performed to rotate the robot about 360°.

1. A laser pointer is attached at the body of the robot.
2. A label of the pointer is placed on a wall that is far enough from the robot.
3. The robot is turned approximately about an angle less than 360°.
4. Then the robot is manually turned to match the laser pointer with the previously placed label on the wall.

The procedure provides sufficient precision for identification of mouses positions $M_i$. A scene of the robot, label, and laser pointer is shown in Fig. 4.

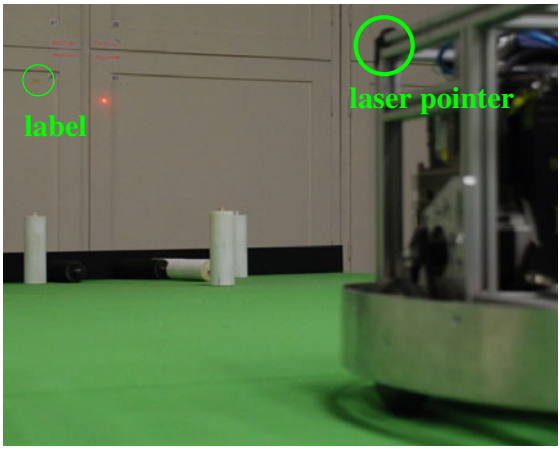



**Fig. 4.** Mounted laser pointer at the robot and label on the wall during precise rotation about 360°

## 6 Experiments

Three optical mouses are used in the self-localization system, because the fourth mouse with the twisted electronic board provided too noisy data. All experiments have been performed with the experimental robotic platform called G$^2$BOT developed at the Gerstner Laboratory. The platform is based on the ER1 kit from the Evolution Robotics [1] with two stepper motors and the RCM control unit. The robot movements are controlled by the Player program [7]. It should be noted that the RCM is connected to the on-board computer by the USB and together with the Player software leads to a significant transport delay, which causes robot movement to be imprecise. For example, the robot does not move exactly forward when commanded to do so, because one stepper is started a bit earlier than the second one. The imperfections of the robot control loop causes real paths to deviate from desired ones, however a path calculated from RCM board odometry looks like the desired one. This property is not an issue in the performed experiments, because the optical mouses are independent system and the true (real) final positions of the robot have been measured manually.

The proposed method of pose estimation is verified in a set of experiments where the robot is navigated along the path with square shape that is approximately two meters long. The robot moves in turn-move manner, which means the robot is turned in the desired directions at first. After that, a forward motion by the given distance is performed. In our experiments, robot forward velocity was around 0.05 m.s$^{-1}$, like in [3], and the radial velocity was around 0.05 rad.s$^{-1}$.

The error of the proposed localization method is evaluated by the distance of the real position from position provided by the localization system at the final point of the path. The average value of the distance from several measurements

is denoted as $\overline{d}$. The repeatability of the localization precision is measured as the sample standard deviation of the distances that is denoted as $s_n$.

The pose estimation is also examined for several reading periods in which particular changes are summed before the position is estimated. In all other cases the estimation is calculated from the data record that contains changes of all used mouses. The robot is primarily moved on a flat surface that is located in the main corridor of the university building. The surface represents typical indoor floor and does not contain any significant disturbances. The identified parameters are then used for the playfield that is made to match the standard surface of the EUROBOT competition, as it is created from the clipboard and the surface is relatively rough.

### 6.1 Pose Estimation

At first the parameters of the used mouses are identified by the procedure described in Section 4. The robot is moved around 55 cm and turned about 360°. Identified parameters are depicted in Table 1. To simplify transformation from sensor units to cm the average value 662 is used in both sensor axes and for all mouses.

**Table 1.** Estimated parameters of the optical mouses

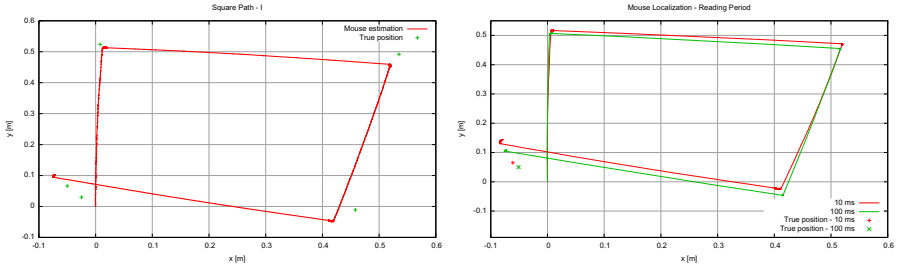
| **nbr.** **Mouse** | $\theta$ [°] | $m_x$ [cm] | $m_y$ [cm] | **sensor units**[2] **per cm** |
|---|---|---|---|---|
| 1 | 179.7 | 7.82 | -12.94 | 642 |
| 2 | 179.6 | -8.03 | -12.98 | 663 |
| 3 | 179.8 | -8.46 | 19.10 | 682 |

Two selected paths obtained from the mouse localization system are shown in Fig. 5a, where the true positions of the robot are visualized as small '+'.

The precision of the localization from the mouses and inbuilt odometry system of the RCM is depicted in Table 2, where 10 ms and 100 ms are reading periods. It should be noted that estimated path depends on the initial position of the robot, which has been set manually and only approximately to the same place. The deviation is more important than the average value, because it represents repeatability of the localization.

The reading period decreases the precision of the localization. Notice that $s_n$ is similar for both examined periods. It indicates possible same results for higher velocity of the robot, e.g. if the robot will be moved ten times faster. The worse precision can be caused by the naïve summation of measurements without rotation about the angle $\theta_i$. From this perspective estimated position of the robot is not so bad, which can be affected by the turn-move navigation together with the almost parallel orientation of the mouses with the forward direction of the robot. An example of estimated paths for both periods are shown in Fig. 5b.

[2] In $y$ direction.




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

**Fig. 5.** Selected paths with true positions

**Table 2.** Precision of the localization method

| **Method** | $\overline{d}$ [cm] | $s_n$ [cm] |
|---|---|---|
| mouses, *immediate processing* | 2.09 | 0.77 |
| RCM odometry | 5.02 | 0.68 |
| mouses, period 10 ms | 4.02 | 2.23 |
| mouses, period 100 ms | 2.54 | 2.21 |

### 6.2 Playfield Surface

It is a known fact that resolutions of the mouses depends on the surface [6]. To evaluate influence of different surfaces, position of the robot is determined with identified parameters from the prior experiments and compared with the new set of parameters for the playfield that are shown in Table 3. The new average value of the transformation constant of the sensor units per cm is 660.

An example of a path obtained from the proposed method with the prior found parameters and a path determined for the new identified parameters are shown in Fig. 6. The overall localization errors and standard deviations are summarized in Table 4. The standard deviations are very similar, therefore the higher value of $\overline{d}$ is cased by the systematic error that is reduced by the new identification of parameters.

**Table 3.** Estimated parameters of the optical mouses for the playfield surface

| **nbr. Mouse** | $\theta$ [°] | $m_x$ [cm] | $m_y$ [cm] | **sensor units$^2$ per cm** |
|---|---|---|---|---|
| 1 | 179.7 | 8.40 | -13.67 | 645 |
| 2 | 179.3 | -8.80 | -13.65 | 647 |
| 3 | 179.7 | -9.69 | 21.07 | 685 |

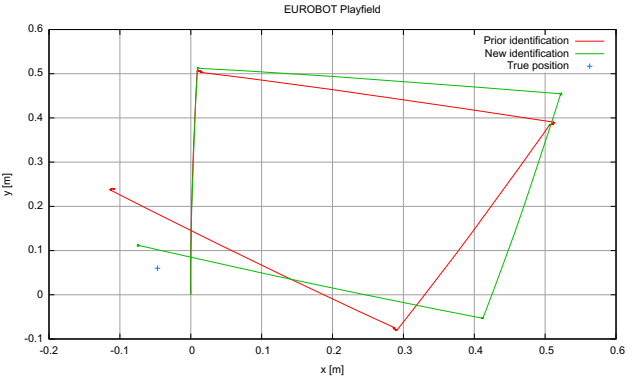



**Fig. 6.** Differences of the estimated robot position with prior and new identified mouses parameters

**Table 4.** Overall precision of the localization methods for the EUROBOT playfield

| **Method** | $\overline{d}$ [cm] | $s_n$ [cm] |
|---|---|---|
| mouses - *prior identification* | 18.69 | 0.27 |
| mouses - *new identification* | 5.45 | 0.31 |
| RCM odometry | 4.88 | 0.34 |

### 6.3 An Example of Outlier Detection

The advantage of using several mouses is the possibility to detect a wrong measurement, i.e. detection of outliers. The effect of outlier detection is demonstrated in Fig. 7. One of the three used mouses is incorrectly identified and

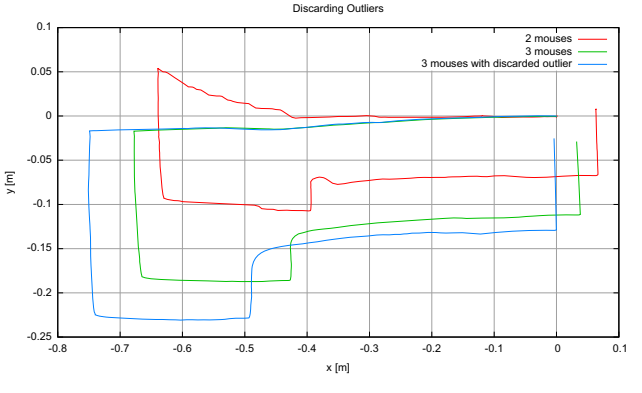



**Fig. 7.** Estimated robot position from three optical mouses with one incorrectly identified mouse

provides wrong measurements. The red path represents position of the robot computed from one incorrectly and one correctly identified mouse. If all three mouses are used, the position estimation precision is improved, but it is still affected by wrong measurements. The estimation is improved significantly if the outlier detection is used and outliers are excluded from the position calculation.

## 7 Conclusion and Future Work

The presented results show that used optical mouses provide competitive estimation of the robot position to the precise odometry provided by the RCM. Regarding to the cost of the RCM, the mouses are cheaper, therefore the optical mouses are considered to be useful for a localization of a robot on flat surfaces like in the EUROBOT 2010 competitions. The main advantage of mouse based localization system is its independence on other parts of the robot system, so it can be used in cases where a slippage or a collision affects robot position.

Despite the fact that feasibility of the proposed method and problem formulation has been verified in a set of preliminary experiments, additional experiments should be performed. In these experiments, the following aspects should be considered. A more general path has to be examined instead of the turn-move navigation, e.g. a path composed circular segments. Additional experiments should be performed to estimate the effect of robot velocity on the precision. The pose estimation might be improved if resolution of each mouse is considered individually instead of single conversion parameter, which can be also useful if various types of sensors are combined. The experiments with the reading period should be reconsidered for a more precise model of robot movement, however it seems that immediate data processing provides more accurate estimation of the robot position.

Even though the optical mouses can be easily plugged to an ordinary laptop, an embedded solution based on the set of sensors and microcontroller can be advantageous. Such a system can be smaller and it can require only single (serial) connection to the on-board computer.

**Acknowledgments.** The support of the Ministry of Education of the Czech Republic, under the Project No. 2C06005, to Jan Faigl is gratefully acknowledged. The support of EU and the Ministry of Education of the Czech Republic, under project No. 7E08006 and ICT-2007-1-216240 to Tomáš Krajník is also acknowledged.

## References

1. Evolution Robotics - ER1 Personal Robot System, http://www.evolution.com/er1
2. Hu, J.S., Chang, Y.J., Hsu, Y.L.: Calibration and Data Integration of Multiple Optical Flow Sensors for Mobile Robot Localization. In: IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, SUTC 2008, pp. 464–469 (2008)

3. Lee, S.: Mobile robot localization using optical mice. In: IEEE Conference on Robotics, Automation and Mechatronics, vol. 2, pp. 1192–1197 (December 2004)
4. Lee, S., Song, J.B.: Mobile robot localization using optical flow sensors. International Journal of Control, Automation, and Systems 2(4), 485–493 (2004)
5. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems 18, 249–275 (1994)
6. Palacin, J., Valgaon, I., Pernia, R.: The optical mouse for indoor mobile robot odometry measurement. Sensors and Actuators A: Physical 126(1), 141–147 (2006)
7. The Player Project - Free Software tools for robot and sensor applications, http://playerstage.sourceforge.net/
8. Sekimori, D., Miyazaki, F.: Self-localization for indoor mobile robots based on optical mouse sensor values and simple global camera information. In: IEEE International Conference on Robotics and Biomimetics (ROBIO 2005), pp. 605–610 (2005)
9. Sekimori, D., Miyazaki, F.: Precise dead-reckoning for mobile robots using multiple optical mouse sensors. In: Informatics in Control, Automation and Robotics II, pp. 145–151 (2007)