33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Absolute position calculation for a desktop mobile rehabilitation robot based on three optical mouse sensors

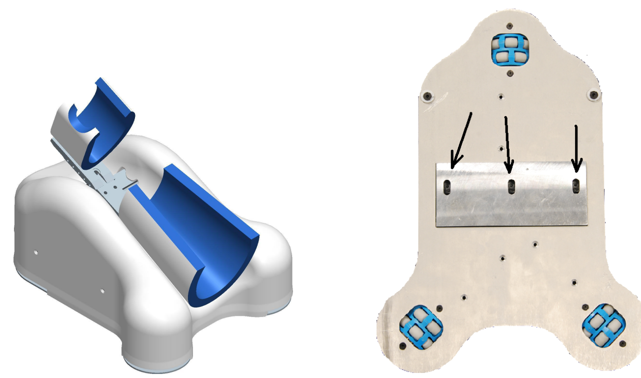
Haritz Zabaleta, David Valencia, Joel Perry, Jan Veneman, Thierry Keller
Tecnalia Research and Innovation
Paseo Mikeletegi, 1, 20011 Donostia
Email: haritz.zabaleta@tecnalia.com

***Abstract*—ArmAssist is a wireless robot for post stroke upper limb rehabilitation. Knowing the position of the arm is essential for any rehabilitation device. In this paper, we describe a method based on an artificial landmark navigation system. The navigation system uses three optical mouse sensors. This enables the building of a cheap but reliable position sensor. Two of the sensors are the data source for odometry calculations, and the third optical mouse sensor takes very low resolution pictures of a custom designed mat. These pictures are processed by an optical symbol recognition algorithm which will estimate the orientation of the robot and recognize the landmarks placed on the mat. The data fusion strategy is described to detect the misclassifications of the landmarks in order to fuse only reliable information. The orientation given by the optical symbol recognition (OSR) algorithm is used to improve significantly the odometry and the recognition of the landmarks is used to reference the odometry to a absolute coordinate system. The system was tested using a 3D motion capture system. With the actual mat configuration, in a field of motion of 710 x 450 mm, the maximum error in position estimation was 49.61 mm with an average error of 36.70 $\pm$ 22.50 mm. The average test duration was 36.5 seconds and the average path length was 4173 mm.**

I. Introduction

ArmAssist is a wireless robot for after stroke upper limb rehabilitation [1]. This project aims at the development of a inexpensive, portable, large-workspace, modular, mobile robotic system for upper-limb functional rehabilitation. The system implements task-oriented therapy, combining motor recovery and functional recovery. The robot is attached to the patient forearm, and the patient will perform pre-defined exercises moving it on a table. The device will be used for the treatment of gravity induced disturbance of coordination between shoulder abduction and elbow flexion. Patients using this device will be able to start the rehabilitation program in the care centers and to continue it at home. The system enables the assessment of patient residual motor/functional ability, implement the physical means for the therapy, and finally assess the results of the therapy. Two important requirements of the robot are: *i*) Accurate measurement of the robot in a flat surface and *ii*) Technologically easy and cheap to implement.

In our device, we wanted to avoid installing any accessory electronic subsystem (beacons, IR landmarks, webcams, external cameras, etc) which lead to odometry as the chosen positioning technique. Odometry is widely used method to estimate both position and orientation in mobile robots. The core idea is the integration of incremental motion information over time [2][3]. The motion information is typically data from incremental wheel encoders in case of wheeled robots.



(a) Drawing (b) Position of the three mice

Fig. 1: ArmAssist device

Many techniques have been proposed in order to reduce errors [4]. Odometry is considered an accurate short term position estimation method, but as errors accumulate, the uncertainty of position estimation inevitably increases over time. The artificial landmark recognition technique uses the information of location of distinctive landmarks at known locations to obtain position information.

The robot's positioning system is equipped with three optical mouse sensors. Two mice provide information of their displacement in two dimensions to the odometry. The third mouse is used as a CCD camera to obtain pictures of the surface in order to extract and recognize visual landmarks of the mat placed below the robot.

The idea of using optical mouse sensors for dead-reckoning for mobile robots is not new [5] [6] [7] [8] [9]. The optical mouse is a very low-cost sensor and has the advantage that the measured displacement is independent from the kinematics of the robot because the optical sensor uses external natural microscopic ground landmarks to obtain the effective relative displacement [10].

The device will be used mostly in game based rehabilitation.

Therefore the position estimation should be good enough in order not to intefear the visionproprioception interactions and also provide sufficient quantitative feedback to the therapists for monitoring therapy progress. The precission needed for game based rehabilitation will be further evaluated during the one year patient trials currently in progress.

II. Method

This method comprises optical mouse based odometry, and a 3rd mouse used as CCD camera with the algorithm to extract and recognize the landmarks on a custom mat and the corrective strategy to detect the misclassifications of the landmarks and fusing of the odometry calculations with the absolute position data.

A. Odometry calculation

Odometry calculation consists in solving the differential drive kinematics for a two wheel robot. This has been previously solved also in non-holonomic wheeled robots with differential steering systems [3]. In our case, the two wheel encoder information has been substituted by two optical mice sensors [10]. The two mice are placed parallel and in a known distance from each other. Mice data are recorded at 125Hz. At this sampling rate, the mice move very little (8 mm at highest speed). It is considered as normal working speed (0.1m/s up to 0.05m/s at slow working speed). This would lead to 0.4 mm changes per reading. This can produce very strong changes in estimated trajectory direction. Therefore a new position is estimated every 40 ms in order to avoid too abrupt changes in estimated trajectory direction.

B. Absolute position estimation

We propose an absolute position estimation method, to determine, in discrete time stamps, the position of the sensor within a custom mat using artificial landmark recognition.

1) Custom printed mat design: The mat spans over the whole field of motion of the arm and contains 16 cells each of which is labeled with a unique landmark.

2) Optical symbol recognition algorithm: The Avago Technologies ADNS-3080 optical mouse sensor has a programmable frame rate over 6400 frames per second which enables high shutter speeds.

The algorithm is divided in the following steps.

- Flat field correction.
- Normalization
- Angle extraction
- Bilinear image rotation and resampling
- Landmark extraction and recognition
- Cell position identification

- **Flat-field correction:** Uneven illumination, the LED orientation, dirt/dust on lenses and other factors can result in a poor quality image. The flat field image should, ideally, be a field of view of the mat without any symbols in it.
- **Normalization:** The flat-field corrected image is then normalized to numbers between 10 (nearly black) and 240 (nearly white). This is done so in order NOT to have pixels with value zero, as the zero value is reserved for not computed for the rotation process.
- **Angle extraction:** The landmarks were designed in sucha a way, that when repeated, the draw a grid of continuous black lines in the captured frame. The algorithm to detects these lines is an extension of the Hough transform [11][12]. Instead of searching and voting for a single line this algorithm searches for multiple lines. The main lines are parallel to each other in a known distance from each other. If the 10 most voted candidate's result are withing a maximum error of 10 degrees, the most voted solution is accepted and the pictures is used for landmark detection. Otherwise, the image is rejected.
- **Bilinear image rotation and resampling:** The final goal is to rotate around the center of the image for landmark extraction and recognition. As the footprint of the image remains constant, the size (in pixels) of the landmark is known.
- **Landmark extraction and recognition:** It is necessary to re-shift the image into the correct position. This is done using the information landmark design. All landmarks start with a black column. This is used to determine which column is the first one. Similarly the row re-shifting is done. Once the landmark picture is re-shifted is easy to locate the areas where the symbol bits are and classify them as black or white.
- **Cell position identification:** The cell position in the mat is known using a lookup table.

C. Odometry and absolute position correction fusing strategy

Mice work better in less homogenius surfaces (e.g. porous surfaces, uneven surfaces) as the chip is able to detect more features of the surface. The Avago Technologies ADNS-3080 optical mice provide a parameter of the surface quality (SQUAL). Its average value on white paper is 75 (in optimal hight adjustment) and the maximum value is 169 [13]. Due to the mat design, the number of features with which the mouse chip estimates its motion is very high (SQUAL = 131.32 ± 6.84).

There are many strategies to correct systematic odometry errors [14] based on redundancy of the mice data. As the mice are solidly attached to each other, the readings of the mice along the axis that joins the centers of the two sensors should be equal. This redundancy can be used to change some parameters of the odometry calculation algorithm for better odometry estimation [7].

On the first version of the 3 optical mouse sensor based positioning system, non systematic odometry error correction strategy had been implemented to fine tune the paramters. The angle estimation errors are the core source of position estimation error calculated by odometry. The odometry correction we present focuses on orientation error correction. The method utilizes redundant sources of orientation changes.

The OSR algorithm does not always return a value (section II-B2), thus the absolute orientation information θ_A has an asynchronous sampling rate. First, the relative orientation values sampled at 25Hz ($\theta_R(n)$) are subsampled to get the information in the OSR data time slots ($\theta_R(\hat{n})$). Once this is done, the consistency between the absolute orientation $\theta_A(\hat{n})$ and the relative orientation $\theta_R(\hat{n})$ is checked. If the information of the relative orientation variation matches, within boundaries ($Th\theta$), with the changes of the absolute orientation variation for the last four valid images (1), the absolute orientation information is defined as compliant, and used to determine the actual orientation error as shown in (2).

$$[|\Delta\theta_R(\hat{n}-i) - \Delta\theta_A(\hat{n}-i)|]_{i=0,1,2,3} < Th\theta \tag{1}$$

$$\epsilon(\hat{n}) = \frac{\sum_{i=1}^{4}\theta_R(\hat{n}-i) - \sum_{i=1}^{4}\theta_A(\hat{n}-i)}{4} \tag{2}$$

The same method can be used for position data consistency checking and correction if extended to two dimensions. First, the relative position values $P_R(n)$ are subsampled to get the information in the OSR data time slots $P_R(\hat{n})$. The positions given by the OSR algorithm are stored in the $P_A(\hat{n})$ vector.

In case of perfect odometry calculations, there is a displacement vector **D** that fullfills the condition that every point of the $P_R(\hat{n})$ fits into the $P_A(\hat{n})$ vector with an error below the half of the cell width (for the X axis ϵ_X) (3) and below the half of the cell height (for the Y axis ϵ_X) (4). In reallity this is not true and the optimal value of **D** is calculated in order to make the maximum number of points fullfill the condition. The vector **E** is calculated every time that new information of OSR is provided.

$$X_A(\hat{n}) - \epsilon_X \leq X_R(\hat{n}) + D_x \leq X_A(\hat{n}) + \epsilon_X \tag{3}$$

$$Y_A(\hat{n}) - \epsilon_Y \leq Y_R(\hat{n}) + D_y \leq Y_A(\hat{n}) + \epsilon_Y \tag{4}$$

III. Results

A. Optical mouse reliability

In the characterization of the optical mouse behavior, straight trajectories were carried out manually in slow, medium and fast movements. As the robot aims after stroke upper limb rehabilitation the maximum working speed is estimated at 1m/s. The tests were carried out in 0.1 m/s, 0.5 m/s and 1m/s speeds.

TABLE I: Spacial Resolution at Different Speeds

Speed [m/s]	CPI mouse # 1	CPI mouse # 2
0.1	1435.6 ± 14.8	1251.2 ± 21.8
0.5	1452.1 ± 31.3	1185.6 ± 47.4
1	1501.7 ± 22.3	1229.6 ± 28.2

B. Orientation correction

The orientation detection algorithm with OSR has been tested. The images of a database of 100 images corresponding to a random trajectory of 30 seconds were used to determine the accuracy of the OSR orientation data. For each of the images an optimal solution was given and compared with the output of the OSR algorintm. The error in absolute value of the results of the OSR algorithm after the consistency check was 2.81 ± 1.57 degrees of an angle (maximum was 5.18 and minimum was 0.04 [deg]). The number of frames with processed angle data was 90 and 70 frames were consistent in relative and absolute orientation changes.

As the most important corrective action concerns orientation correction, a test has been carried out to quantify the odometry errors due to orientation drift. The robot was manually moved following a square shaped trajectory finishing at the starting point. The following parameters have been measured: Test Duration (TD), length of the trajectory with orientation correction (OC) and the trajectory without orientation correction (NOC), mean root square difference between the positions of the trajectory with and without orientation corrections (MRSD), the total length of the trajectories (Length), the final points distance to the initial point (Distance to SP), and the final point's estimation improvement due to orientation correction factor (Improvement) (table II). All values are given in millimeters.

TABLE II: Orientation Correction Effect Test Results

Test #	Length		MRSD	Distance to SP		Improvement
	OC	NOC		OC	NOC	
1	1226	1226	12.42	81.14	168.80	87.66
2	1225	1225	36.16	89.44	191.57	102.13
3	1237	1237	40.31	40.07	178.64	138.56
4	1048	1232	24.70	180.38	291.90	111.52
5	1121	1364	28.84	145.80	267.01	121.20
6	1141	1212	33.16	114.50	217.65	103.16
7	1238	1238	30.02	45.44	161.03	115.59
8	1213	1213	32.03	52.50	160.07	107.57
9	1213	1213	38.86	39.38	167.91	128.53
10	1216	1288	42.16	44.51	183.28	138.76
Mean	1187.8	1244.8	31.9	83.3	198.8	115.5
SD	63.3	47.4	8.7	49.5	46.1	16.5

C. Position correction

The position estimation of the OSR algorithm was tested a 3D motion capture system. The mat had a 720 x 450 mm field of motion divided into 16 cells of 480 x 112.5 mm. The test consisted in moving the device manually thoughout the whold mat field of motion randomly during aproximatelly 30s. The test was repeated 5 times. The motion was captured at a rate of 100 Hz and the data was post-processed in order to obtain the devices angle and the trayectory of the center point.

This data was compared with the presented absolute position calculation estimation method. The maximum error in position estimation was 49.61 mm with an average error of 36.70 ± 22.50 mm. The average test duration was 36.5 seconds and the average path length was 4173 mm.

As an example the trajectory of the test#4 is presented in the following figure.

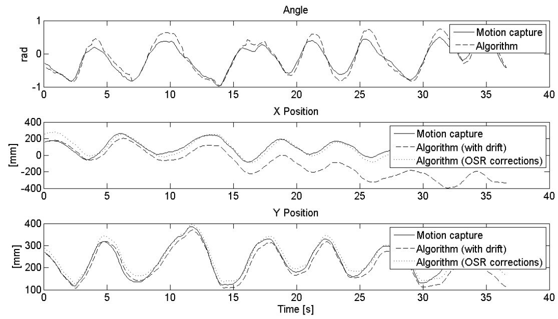


Fig. 2: Position and orientation differences

D. Position initialization

The algorithm is reseted when the mice are loose contact with the surface. This is detected very acuratelly by monitoring the surface quality (SQUAL) paramter provided by the mice controller. All the parameters are reseted to their initial values, and the position is set as *unknown*. Inmediatelly after initiallization (at turn-on or after lift up) the user hast to move the device slightly. If the mice detect movement, the acquired images are processed, and after four valid images the first position, refered to the absolute coordinate system is calculated. This procces takes about 1 second to complete.

E. Computational power

The algorithm has not been jet opticed in order to be able to run on an on-board microprocessor. Nevertheless, it has been running without perceptive delay in a Dell Latitude D630, 2.2GHz Intel Core2 Duo, 2GB RAM. This makes the device completely portable, as no high end PC is needed to run the algorithm. Once the algorithm is ready, it could be programmed into a Field Programmable Gate Array (FPGA).

IV. Conclusion

The study showed similar behavior in slow, medium and fast movements in spacial resolution but quite different from one sensor to another. However, the analysis concluded that even a very cheap optical mouse sensors can be sufficiently accurate, precise and reliable to perform a optical mouse sensor based short term odometry for the ArmAssist robot.

Concerning OSR performance in orientation detection, the modified Hough transform correction strategy, and consistency check showed very good results. The OSR orientation is very reliable as 100 % of the results after the concistency check rule have an error below 5 degrees. Additionally, if this data is used to correct the odometry, the position error (in $\frac{mm}{m}$) decreases significanly by 58.10 %.

Regarding the OSR performance in landmark recognition the accuracy is over 95 % in a 16 group classifier. The raw images have very poor quality and contrast. On the other hand the consistency check rule improves the accuracy to over 99.5 %, but the number of pictures that fullfill the consistency check rule is 72 %, so the time between two absolute position corrections could be rather large (an average of 2.4 corrections per second).

We can conclude that even with very few corrective strategies have been implemented until now, the proposed navigation system can be used in desktop mobile robots where the range of motion of the robot is limited. This method enables the construction of a cheap global positioning system where the main drawback of unbounded error accumulation of odometry is solved.

Future work will target the design of systematic error correction strategies. As the system has sources of redundant position information, strategies for odometry parameter tuning can be implemented. The modification of the odometry parameters will reduce the drift and therefore the overal error. The aim is to increase the accuracy of the system into a level that is satisfactory to cualitative training at home.

Acknowledgment

This work was supported in part by the FIK Project, San Sebastian, Spain.

References

[1] J. Perry, H. Zabaleta, A. Belloso, and T. Keller, “ARMassist: A low-cost device for telerehabiltation of post-stroke arm deficits,” in *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer, 2009, pp. 64–67.

[2] J. Borenstein, H. Everett, and L. Feng, “Where am I? Sensors and methods for mobile robot positioning,” *University of Michigan*, vol. 119, p. 120, 1996.

[3] J. Crowley, “Control of translation and rotation in a robot vehicle,” in *Proceedings of the IEEE Conference on Robotics and Automation*, 1989.

[4] Y. Tonouchi, T. Tsubouchi, and S. Arimoto, “Fusion of dead-reckoned positions with a workspace model for a mobile robot by Bayesian inference,” in *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 2. IEEE, 2002, pp. 1347–1354.

[5] D. Sekimori and F. Miyazaki, “Precise dead-reckoning for mobile robots using multiple optical mouse sensors,” *Informatics in Control, Automation and Robotics II*, pp. 145–151, 2007.

[6] A. Roskilly and N. Tunwattana, “Investigations into the effects of illumination and acceleration on optical mouse sensors as contact-free 2D measurement devices,” 2009.

[7] A. Bonarini, M. Matteucci, and M. Restelli, “Automatic error detection and reduction for an odometric sensor based on two optical mice,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2006, pp. 1675–1680.

[8] J. Palacin, I. Valganon, and R. Pernia, “The optical mouse for indoor mobile robot odometry measurement,” *Sensors and Actuators A: Physical*, vol. 126, no. 1, pp. 141–147, 2006.

[9] S. Lee, “Mobile robot localization using optical mice,” in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 2. IEEE, 2005, pp. 1192–1197.

[10] A. Bonarini, M. Matteucci, and M. Restelli, “A kinematic-independent dead-reckoning sensor for indoor mobile robotics,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2005, pp. 3750–3755.

[11] R. Duda and P. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[12] “Method and means for recognizing complex patterns,” 1962, uS Patent 3,069,654.

[13] A. Datasheet, “ADNS-3080 Data Sheet Avago Technologies,” 2008.

[14] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 6, pp. 869–880, 2002.