High-Precision Robot Odometry Using an Array of Optical Mice

Steven Bell *Student Member, IEEE*

Oklahoma Christian University

24 March 2011

ACKNOWLEDGEMENTS

I am grateful for Dr. David Waldo's guidance during this project and for his support of the Oklahoma Christian IEEE student branch, and to Dr. Bill Ryan for his help in the manufacturing aspects of this work. I would also like to thank Avago Technologies, Marshall Electronics, and M12 Lenses, Inc, for supplying free and discounted parts for this project.

CONTENTS

Abstract

This paper describes a robot position tracking system based on a set of six optical mice. Each mouse sensor uses a lens which results in higher tracking speeds and allows the sensor to be placed centimeters, rather than millimeters, from the floor. The sensors' tracking and data reporting characteristics are discussed, and an algorithm is presented to determine a robot's motion from the sensor readings. Techniques for error detection and correction are presented, along with methods to use the sensor for line following and edge detection.

I. INTRODUCTION

FOR decades, dead-reckoning has been a staple of mobile robot navigation. For ground-based robots, this has typically been accomplished by using a set of shaft encoders mounted to the robot's wheels to measure speed and distance. While this method is simple and well-tested, it suffers from one primary drawback: any time the wheel slips on the surface it will cause an error in the position measurement. This is a major handicap for omnidirectional drivetrains based on omniwheels or mecanum wheels, which by their nature slide easily and can slip in any direction [1]. The encoder approach also suffers in competition environments where the robot is likely to physically collide with obstacles or even be pushed by other robots, causing wheel slip.

Traditionally, computer mice have been plagued with many of the same problems. Achieving accurate motion with a mechanical ball is difficult due to surface differences, dust, and wear. Optical mice based on high-speed optical flow solve these problems, and they have completely supplanted traditional ball mice over the last ten years. The underlying operation of optical mouse sensors has been discussed extensively in prior research [2]–[4], and is summarized in Appendix A.

This paper presents a means of using optical mice as position sensors by pointing them at the floor and reading motion data from them. A robot navigation system based on optical mice has several notable advantages over wheel-based encoders. First, it has no moving parts and does not need to be in contact with the floor, which ultimately reduces friction and wear on the system. Second, all of the mouse chips on the market track motion in two dimensions simultaneously, and do so at increasingly high resolutions - up to five thousand counts per inch (CPI) [5]. Mouse chips typically come in 8 or 16-pin integrated circuit packages, which means they are both small and easy to mount. Finally, the sensors are abundant and inexpensive: Avago Technologies has shipped over 0.8 billion mouse sensors [6], and typical chips cost less than a dollar apiece in bulk.

A. *Prior research*

Researchers, undergraduates, and hobbyists have been experimenting with optical mice as sensors for robot navigation for the last decade. Most of the published papers and reports take one of three approaches: One set of projects has focused on the validation of the optical mouse sensor as a means of robot position tracking and the development of algorithms to achieve robust 3-degree-of-freedom (3-DOF) tracking with an array of sensors [7]–[12]. That is, they combine the data from multiple sensors to measure a robot's translation in two dimensions plus rotation. Nearly all of these projects use the mouse chips in their intended configuration - they are placed within millimeters of the tracking surface and utilize the commercial-off-the-shelf (COTS) lens and illumination LED. Many of these systems use complete circuit boards and even plastic shells taken from commercial optical mice.

A second area of research has been to carefully characterize particular sensors. Several findings are particularly relevant: although the sensor measures motion very precisely and repeatably [4], it has limitations.

While the sensor is within its working height range, the resolution (in CPI) is a linear function of height. When the sensor is raised further from the surface, it sees a larger patch of the tracking surface, and each pixel represents a larger distance. Because the sensor tracks movement using pixel displacement, one inch of motion will give a higher distance count when the sensor is close than when it is far away [3]. When the sensor is outside of its working focal range, the measured distance drops off sharply.

The measured displacement appears to vary depending on the orientation of the sensor relative to the direction of motion [3], [4]. There are two reasons for this. Many common surfaces - including paper, which is a popular test surface - exhibit a grain. That is, they have a directional surface pattern that may result in better tracking in some dimensions versus others. Moreover, [4] and [13] show that one axis consistently reports higher measurements than the other - a surprising find, given that the chip's internal algorithm is presumably indifferent to the movement direction. This is caused by the fact that the standard design for optical mice includes a single light source along one axis, causing it to track more accurately than the other [13]. Using two orthogonal light sources eliminates this effect.

Finally, the sensor readings are strongly dependent on the surface, which means that the sensor must be calibrated for each new environment, and it will not work reliably in situations where the surface is not uniform [3].

The third research area is the application of optical mice to a variety motion tracking problems where the robot is not within millimeters of the tracking surface. This requires the use of a non-standard lens to refocus the light onto the sensor at a different distance. This approach has been shown to work across a wide range of distances, from mobile robots less than 2 cm above the surface [14] to robots flying tens of meters above the ground [15].

This paper synthesizes these threads of research to create a system based on a set of optical mice which can be used for robust and precise robot navigation.

B. Conditions for mouse use

Based on the research discussed above, there are several constraints which inform the use of optical mice as planar 3-DOF position sensors. First, the tracking surface should have a uniform and unstructured texture - that is, it should not have a grain or other directional pattern which will favor tracking in one direction over another. Second, the environment must allow calibration of the sensors prior to their use. Finally, the surface should be flat, so that the distance from the sensor to the surface is essentially constant.

These constraints make the optical mouse unsuitable for many applications, but in other domains - such as small scale robotics contests - these constraints are not a problem. With these limitations in mind, we proceed with the design of our system.

II. Sensor design

The proposed system is a small, 2-wheel differential drive robot for the IEEE Region 5 robotics competition which uses a set of six optical mouse sensors for navigation. Four sensors are located across the front and two are placed in the back corners. A rendering of the robot base is shown in Figure 1. Encoders are also mounted on the wheels for speed control; their integration is described in Section IV-B. The competition playing field is made of fiberboard painted flat white, which provides a uniform tracking surface for the mouse sensors.

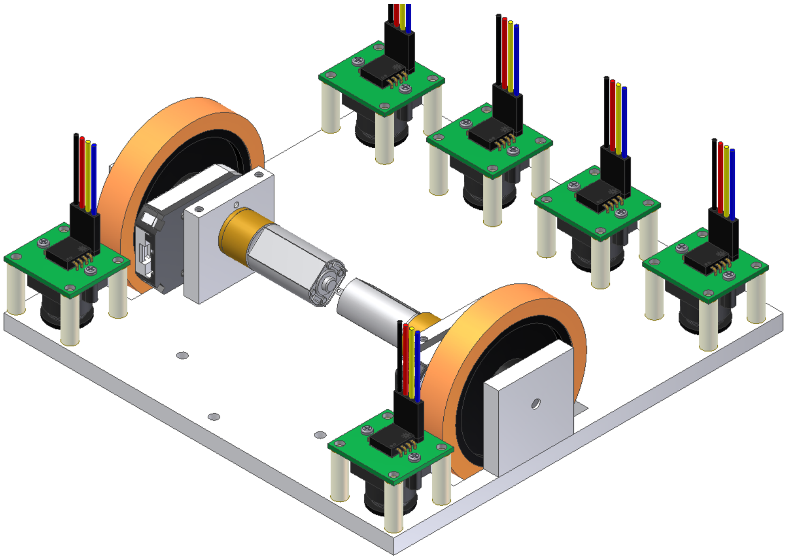
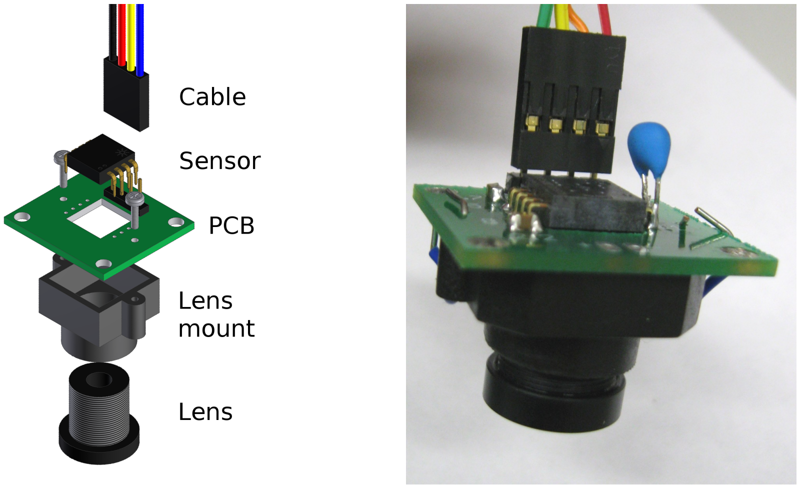


Figure 1: Simplified rendering of the robot base showing the location of the six sensors.

[16] and [2] show mathematically that for optimal performance, the array of sensors should be evenly spaced, located as far from the center of the robot as possible. Because of the additional sensor uses described in Section V, our placement deviates from the optimum. However, the sensors are all located near the edge of the robot, which will still provide very good tracking.

A. Hardware design

The sensor system is based on the Avago Technologies ADNS-2610 and the pin-compatible ADNS-5060. Both sensors have the same form factor and communication interface; the primary difference is in the rated tracking speed. The mouse chip is mounted on a custom printed circuit board which contains a header for connection to a microcontroller and decoupling capacitors as suggested by the sensor datasheet. A COTS lens holder is screwed onto the bottom of the PCB and a standard M12 lens is threaded into it. By using a standard lens mount, it is simple to connect a variety of lenses to the system for testing. A lens with a 3.6 mm focal length was selected for the ADNS-2610; a detailed comparison of lens focal lengths and selection criteria is given in Appendix B. An exploded view of the sensor assembly is shown in Figure 2a and a photograph of the assembled unit in Figure 2b.



(a) Exploded view of the sensor assembly

(b) Photograph of the sensor assembly

Figure 2: Rendering and photograph of the sensor assembly. 3-D models of mouse chip and lens mount courtesy of Avago Technologies and Marshall Electronics respectively.

The surface beneath each sensor is illuminated by a single Avago HLMP-ED80 LED. To reduce the measurement bias discussed in [13], some of the LEDs are placed parallel to the robot's direction of motion and others perpendicular to it. Ideally, two LEDs would be used for each sensor, but this was impractical given the size and electrical current constraints on the system.

To determine the best location for the LED, we measured the surface quality measured by the mouse with the LED placed at varying distances and angles. The result is shown in Figure 3. From the plot, it is clear that the best lighting is essentially parallel to the surface and less than an inch away - which is essentially the configuration in a standard optical mouse.

Although the surface remains well illuminated at steeper angles, the surface quality degrades strongly beyond 20°, because the light washes out the surface rather than causing shadows and bringing out small surface features. An angle of 15° was chosen as a compromise between mechanical constraints and optimal sensing.

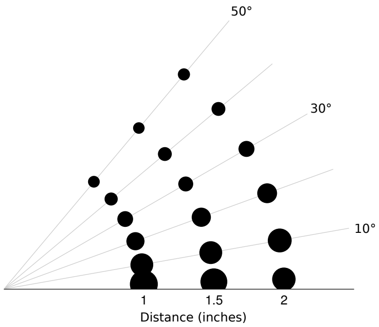


Figure 3: Plot of measured surface quality as a function of LED distance and angle. The dot area is proportional to the surface quality.

B. Software driver

Most of Avago's mouse chips share a common synchronous serial interface to a set of internal registers which contain the tracking data. In a typical mouse, a microcontroller interfaces with the mouse chip, integrates the buttons and mouse wheel, and provides a USB connection to the host PC. In our application, the control hardware (an Arduino microcontroller board) is wired directly to to the mouse sensor so that we have direct access to all of the chip's registers, not merely the motion reports. The chip provides several pieces of data:

- X and Y differential motion values, which report the number of "counts" since the last time the register was read.
- A Surface Quality (SQUAL) measurement, which gauges how suitable the surface is for tracking. A higher SQUAL value indicates a higher number of tracking features, and generally corresponds to a more accurate distance measurement. Lower SQUAL values indicate that the surface is especially smooth and featureless or that the camera is not properly focused on the tracking surface.
- The imager shutter speed, which can be used to derive the brightness of the tracking surface. High (longer) shutter values indicate a darker surface, low values indicate a bright surface.
- The sensor also provides a means of capturing image data directly from the sensor, which was used to obtain the mouse images presented in the Appendix.

The software provides methods for reading each of these values, and communicates with a GUI running on a PC which can display the data in real time.

III. Motion tracking

There are three important frames of reference in the motion tracking problem, which are shown in Figure 4. The first is the absolute coordinate frame, designated by the Cartesian coordinates X_A and Y_A. The second is the robot frame of reference, specified by X_R and Y_R. The final reference frame is aligned with each sensor's measurement axes and is designated x_i and y_i. Note that although only one mouse sensor is shown in Figure 4, there can be an arbitrary number of mice, and their reference frames will be labeled x_1, x_2, x_3 and so forth. We can also reference these points using a polar coordinate system, which will be useful in later derivations. The robot's position in absolute coordinates is given by r_R and θ_R, while each mouse's position is given by r_i and θ_i. Each mouse's rotation relative to the robot is labeled ϕ_i.

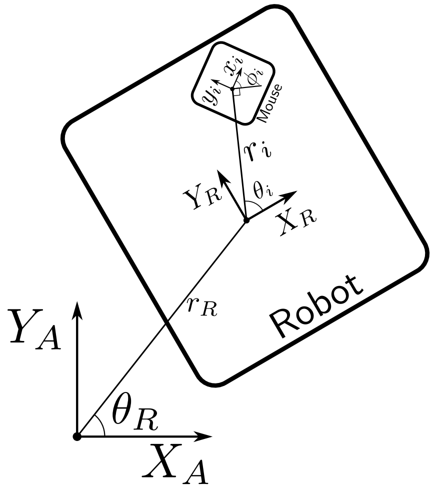
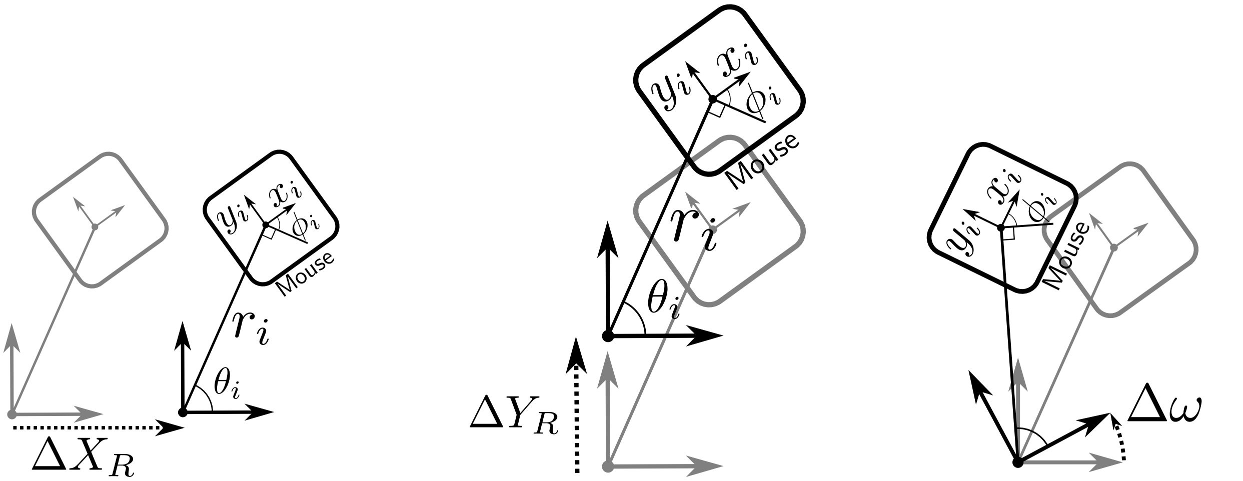


Figure 4: Important frames of reference: absolute coordinates, robot coordinates, and mouse coordinates.

Using these definitions, we can relate the 3-dimensional motion of the robot to the motion of each mouse.



(a) Translation in the X-dimension, ΔX

(b) Translation in the Y-dimension, ΔY

(c) Rotation about the robot center, $\Delta\omega$

Figure 5: Three possible movements for the robot base and the resulting motion measured by the optical mouse sensor.

If the robot base translates by ΔX_R as shown in Figure 5a, then the mouse will also translate by ΔX_R. Because the mouse has a different orientation than the robot, the values measured by the mouse axes x_i and y_i will be:

$$x_i = \Delta X_R \cdot \sin(\theta_i + \phi_i) \tag{1}$$

$$y_i = \Delta X_R \cdot \cos(\theta_i + \phi_i) \tag{2}$$

Likewise, if the robot translates by ΔY_R, the sensors will measure

$$x_i = -\Delta Y_R \cdot \cos(\theta_i + \phi_i) \tag{3}$$

$$y_i = \Delta Y_R \cdot \sin(\theta_i + \phi_i) \tag{4}$$

If the robot rotates around its center by $\Delta\omega$ (where ω is in radians), the mouse will travel a distance of $\Delta\omega \cdot r_i$. The measurements along the mouse axes will be

$$x_i = -\Delta\omega \cdot r_i \cdot \cos(\phi_i) \tag{5}$$

$$y_i = \Delta\omega \cdot r_i \cdot \sin(\phi_i) \tag{6}$$

We can add equations 1, 3, and 5 to get the total motion measured by the sensor x-axis:

$$x_i = \Delta X_R \cdot \sin(\theta_i + \phi_i) - \Delta Y_R \cdot \cos(\theta_i + \phi_i) - \Delta\omega \cdot r_i \cdot \cos(\phi_i) \tag{7}$$

And we can do the same for the y-axis with equations 2, 4, and 6:

$$y_i = \Delta X_R \cdot \cos(\theta_i + \phi_i) + \Delta Y_R \cdot \sin(\theta_i + \phi_i) + \Delta\omega \cdot r_i \cdot \sin(\phi_i) \tag{8}$$

Since we know the motion of each sensor, we can put those values into equations 7 and 8 and solve for ΔX, ΔY, and $\Delta\omega$. With only one sensor, we have 2 equations and three unknowns, but each additional sensor provides two more solutions to these equations. With six sensors, we have a total of 12 equations and three unknowns. This is an overconstrained system, and because of errors there will be no solution which can satisfy all 12 equations. Instead we use a linear least squares algorithm to find the "best fit" for the data. Mathematically, this is equivalent to finding the linear trendline for a set of data points.

First, we write the system based on Equations 7 and 8 in matrix form:

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} = \Delta X_R \begin{bmatrix} \sin\alpha_1 \\ \cos\alpha_1 \\ \sin\alpha_2 \\ \cos\alpha_2 \\ \vdots \end{bmatrix} + \Delta Y_R \begin{bmatrix} -\cos\alpha_1 \\ \sin\alpha_1 \\ -\cos\alpha_2 \\ \sin\alpha_2 \\ \vdots \end{bmatrix} + \Delta\omega \begin{bmatrix} r_1\cos\phi_1 \\ r_1\sin\phi_1 \\ r_2\cos\phi_2 \\ r_2\sin\phi_2 \\ \vdots \end{bmatrix} \tag{9}$$

We can rearrange this equation to write

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} = A \begin{bmatrix} \Delta X_R \\ \Delta Y_R \\ \Delta\omega \end{bmatrix} \tag{10}$$

where A is defined to be a matrix with all of the coefficients grouped together.

$$A = \begin{bmatrix} \sin\alpha_1 & -\cos\alpha_1 & r_1\cos\phi_1 \\ \cos\alpha_1 & \sin\alpha_1 & r_1\sin\phi_1 \\ \sin\alpha_2 & -\cos\alpha_2 & r_2\cos\phi_2 \\ \cos\alpha_2 & \sin\alpha_2 & r_2\sin\phi_2 \\ \vdots & \vdots & \vdots \end{bmatrix} \tag{11}$$

To solve for ΔX_R, ΔY_R, and $\Delta\omega$, we multiply both sides of Equation 10 by the inverse of A. However, because there is no single solution to this system of equations, we must use the Moore-Penrose pseudoinverse rather than the true inverse. The pseudoinverse (A^+) is defined as

$$A^+ = A^* \left(AA^*\right)^{-1} \tag{12}$$

Because all of the values in this problem are real numbers, we can use the transpose of the matrix (A^T) rather than the Hermitian transpose (A^*):

$$A^+ = A^T \left(AA^T\right)^{-1} \tag{13}$$

Once the pseudoinverse has been calculated, we can easily find the motion as a function of the sensor measurements [2]:

$$\begin{bmatrix} \Delta X_R \\ \Delta Y_R \\ \Delta \omega \end{bmatrix} = A^+ \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \end{bmatrix} \tag{14}$$

Once the robot's relative motion is determined, translating this value back to absolute coordinates is trivial. This algorithm was modeled with MATLAB, and example results are shown in Appendix C.

IV. Error reduction

Although the individual sensors are very accurate and averaging them together provides even greater accuracy, they are still subject to errors. Ideally, the system should detect which sensor readings are incorrect and discard them in favor of measurements known to be more correct. There are several techniques which we can use to detect these errors, which are described in the following sections.

After detecting the errors, we must account for them in the position estimation algorithm described in the previous section. To do this, we extend the linear-least-squares algorithm to be a *weighted* linear-least-squares algorithm. That is, rather than trying to make the solution fit all of the points equally, it favors some over others. Each measurement is given a weight: A high weighting factor causes the derived equation to fit a data point very closely (at the expense of a looser fit at other points), while a factor of zero forces a data point to be completely ignored.

A. Integration of multiple sensors

In [7], several assumptions are made about the characteristics of the mouse sensors which are used to detect and correct for errors. In general, mouse sensors do not measure a distance greater than the actual distance traveled. The optical flow algorithm may miss counts due to poor focus or insufficient frame overlap, but it will not overestimate the motion. Thus, when there is disagreement between one or more sensors, the sensor with the higher measurement is probably correct.

B. Use of encoder data

The proposed robot system also has a pair of rotary encoders placed on the wheels, which provide very precise angular measurement (8192 counts per revolution). In the same way that multiple mouse sensors can be compared against each other to determine which is more accurate, the mouse sensors can be compared against the encoder readings to detect errors.

Except for rare circumstances when quickly decelerating, the robot will not move further than indicated by the encoders. In almost all situations, wheel slip will cause the encoders to measure a distance equal to or higher than the actual. By contrast, the mouse sensors will measure a value less than or equal to the actual. Thus if the mouse sensors and encoders agree exactly, then the motion estimate is very good. If they disagree, there is an error - although the system cannot directly determine whether the encoder or mouse (or both) is incorrect.

C. Use of surface quality

Each mouse sensor also provides an estimate of its accuracy - the SQUAL reading. In general, lower surface qualities will result in lower measured distances for a given speed. At one extreme, an SQUAL reading of zero indicates that the sensor has no features to track, and thus it will fail to report any motion. The robot can average surface quality measurements from all of the sensors, and then lower the weights of the sensors with below-average surface qualities. Any sensor with a surface quality in the single digits should be completely ignored.

V. ADDITIONAL USES

Besides its obvious use as a motion tracking sensor, the Avago mouse chips have other unexploited applications on mobile robots. In [17], it is shown by reading the shutter value, the mouse sensor can be used to detect light and dark areas on the surface. The shutter readings can be interleaved with motion readings, allowing our motion tracking sensors to simultaneously double as line following sensors. Figure 6 shows a plot of the surface brightness as the sensor was moved over a dark line. When the sensor is over the line, the shutter time reaches its maximum, and the surface quality drops to about 15. By monitoring these readings, the robot can easily differentiate white and black surfaces.

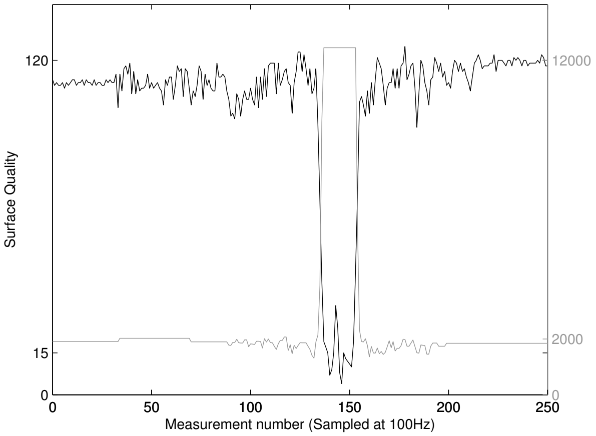


Figure 6: Plot of the shutter value as the mouse was moved across a black line. Higher brightness values indicate darker surfaces, i.e, longer shutter times.

The narrow focus range of the sensor can also be used advantageously. As described earlier, the surface quality is highest when the target surface is in sharp focus and drops to zero when no surface is beneath the sensor. By reading the SQUAL continuously, the robot can detect if one of the sensors has run off the edge of the playing field and take corrective action. Figure 7 shows a plot of the surface quality as the lens-based sensor moves off the edge of the surface. Although the sensor readings are similar to those shown above for crossing a line, here the surface quality drops to less than 5, which indicates that the tracking surface has disappeared altogether.

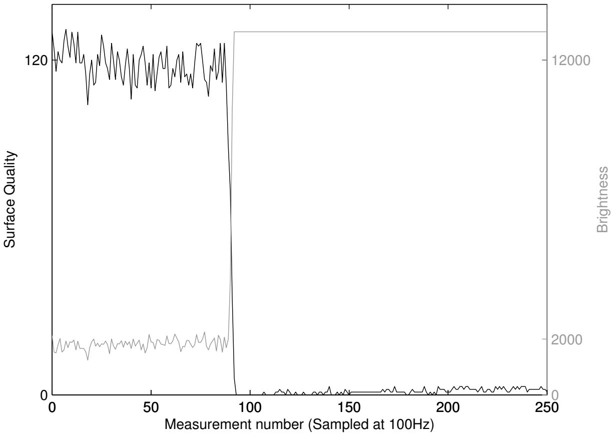


Figure 7: Plot of the surface quality (SQUAL) reading as the mouse was moved over the competition surface and off the edge.

VI. Conclusions and Future work

Optical mice provide a promising option for robot motion tracking, provided the environment is accommodating. The design we have presented in this paper allows small and inexpensive lenses to be used to give higher tracking speeds and reduce height sensitivity, and the algorithms presented can be used to detect and eliminate errors, taking full advantage of sensor redundancy.

To build upon this work, we have built a test jig to carefully characterize the lens-based sensor units. For a given set of focal lengths, we plan to determine the working height range of the sensor and maximum working speed of the sensor. Finally, we will determine whether there is a reliable mathematical relationship between surface quality and recorded distance, which will be used to refine the error-correction algorithm.

References

[1] J. A. Cooney, W. L. Xu, and G. Bright, “Visual dead-reckoning for motion control of a mecanum-wheeled mobile robot,” *Mechatronics*, vol. 14, no. 6, pp. 623–637, 2004.

[2] D. Sorensen, “On-line optical flow feedback for mobile robot localization/navigation,” Master’s thesis, Texas A&M, May 2003.

[3] J. Palacin, I. Valgañon, 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.

[4] U. Minoni and A. Signorini, "Low-cost optical motion sensors: An experimental characterization," *Sensors and Actuators A: Physical*, vol. 128, no. 2, pp. 402–408, 2006.

[5] *Avago ADNS-9500 Datasheet*, Avago Technologies. [Online]. Available: http://www.avagotech.com/docs/AV02-1726EN

[6] Company history. Avago Technologies. [Online]. Available: http://www.avagotech.com/pages/corporate/company_history/

[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*, Apr. 2005, pp. 1675–1680.

[8] J.-S. Hu, Y.-J. Chang, and Y.-L. Hsu, "Calibration and on-line data selection of multiple optical flow sensors for mobile robot localization," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sep. 2008, pp. 987 –992.

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

[10] S. Kim and S. Lee, "Robust mobile robot velocity estimation using redundant number of optical mice," in *Information and Automation, 2008. ICIA 2008. International Conference on*, Jun. 2008, pp. 107–112.

[11] D. Sekimori and F. Miyazaki, "Precise dead-reckoning for mobile robots using multiple optical mouse sensors," in *Informatics in Control, Automation and Robotics II*, J. Filipe, J.-L. Ferrier, J. A. Cetto, and M. Carvalho, Eds. Springer Netherlands, 2007, pp. 145–151.

[12] S. Singh and K. Waldron, "Design and evaluation of an integrated planar localization method for desktop robotics," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 2, May 2004, pp. 1109–1114.

[13] N. Tunwattana, A. Roskilly, and R. Norman, "Investigations into the effects of illumination and acceleration on optical mouse sensors as contact-free 2d measurement devices," *Sensors and Actuators A: Physical*, vol. 149, no. 1, pp. 87–92, 2009.

[14] J. Bradshaw, C. Lollini, and B. Bishop, "On the development of an enhanced optical mouse sensor for odometry and mobile robotics education," in *System Theory, 2007. SSST '07. Thirty-Ninth Southeastern Symposium on*, Mar. 2007, pp. 6–10.

[15] S. Thakoor, J. Morookian, J. Chahl, D. Soccol, B. Hine, and S. Zornetzer, "Insect-inspired optical-flow navigation sensors," NASA Jet Propulsion Laboratory, Tech. Rep. NPO-40173, Oct. 2005.

[16] M. Cimino and P. Pagilla, "Location of optical mouse sensors on mobile robots for odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 5429–5434.

[17] M. Tresanchez, T. Pallejà, M. Teixidó, and J. Palacín, "The optical mouse sensor as an incremental rotary encoder," *Sensors and Actuators A: Physical*, vol. 155, no. 1, pp. 73–81, 2009.

[18] *Avago ADNS-2610 Datasheet*, Avago Technologies. [Online]. Available: http://www.avagotech.com/docs/AV02-1184EN

Appendix

A. *Optical mouse theory of operation*

An optical mouse chip consists of a tiny camera, a digital signal processing (DSP) engine, a serial interface, and appropriate electronics to regulate power and drive the illumination LED. The camera runs at a frame rate of over one thousand frames per second, capturing a grayscale image of a tiny patch of the tracking surface. The DSP engine continuously compares each frame to the one before it, using optical flow algorithms to estimate the relative motion of the sensor. Figure 8 shows two images of the letter ‘e’ in size 6 font with a small amount of movement between the two frames. The DSP hardware within the chip compares these two images and determines that the mouse has moved about two pixels to the right.

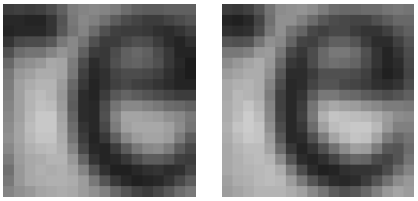


Figure 8: Two images captured from a modified mouse with a tiny amount of movement between them.

A cross section of a typical mouse is shown in Figure 9. The sensor, clip, LED, and lens/light pipe are all provided by Avago Technologies; the mouse manufacturer supplies the baseplate, plastic shell, buttons, scroll wheel, and typically the PC interface hardware.

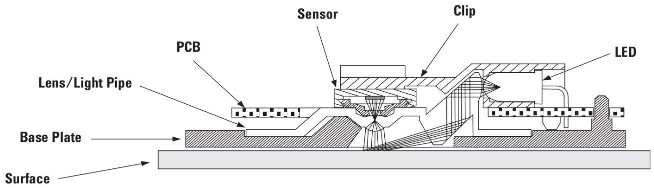


Figure 9: Cross section of a typical optical mouse. Image taken from the Avago ADNS-2610 datasheet [18].

Laser mice work on identical principles, but use a small infrared laser for illumination rather than a visible-light LED. The laser produces a much sharper image and brings out surface features more clearly, which is why laser mice typically have better tracking performance on smooth surfaces than traditional optical mice. Optical mice were used in this work because they are more commonly available and it is easier to provide visible-light illumination. A comparison of several Avago sensors is shown in Table I.

B. *Lens comparison and selection*

An important part of the sensor package design is the selection and placement of the lens. The primary criterion is that the lens provide a magnification that increases the maximum tracking speed of the sensor. A second criterion

Table I: Comparison of several Avago sensors

Name	Type	Rated resolution (counts/inch)	Rated speed (inches/sec)	Imager resolution
ADNS-2610	Optical	400	12	18×18 pixels
ADNS-5060	Optical	1050	30	19×19 pixels
ADNS-7050	Laser	800	20	22×22 pixels
ADNS-9500	Laser	5000	150	30×30 pixels

is the object-to-image distance, the distance between the tracking surface and the imaging sensor inside the mouse chip. For our application, we desire a tracking speed of at least 45 inches per second. The vertical placement of the sensor is negotiable, but should be between 2 and 4 cm above the surface. A diagram of the setup is shown in Figure 10.

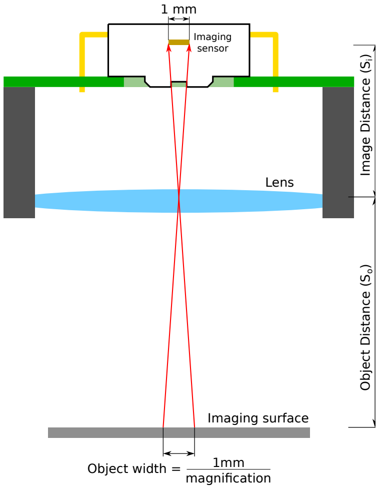


Figure 10: Cutaway view of the idealized lens system. Note that although the lens used in the system is a compound lens rather than a thin lens, for the purposes of this analysis it can be treated the same way based on its effective focal length (EFL).

From the Avago sensor and lens datasheets, we can determine that the standard lens provides a magnification very close to 1. That is, the area of the surface that is recorded is the same size as the camera itself. Based on this information, the desired magnification is determined by dividing the rated tracking speed by the desired speed:

$$\frac{30\,\frac{\text{inches}}{\text{second}}}{45\,\frac{\text{inches}}{\text{second}}} = \frac{2}{3} \approx 0.67 \tag{15}$$

This means that a 1.5 mm wide patch of the surface will be focused onto the 1 mm wide image plane within the sensor.

For a particular focal length f , we can relate the object distance S_o, and the image distance S_i using the lens equation:

$$\frac{1}{f} = \frac{1}{S_o} + \frac{1}{S_i} \tag{16}$$

We also know that the magnification, m, is given by

$$m = \frac{S_i}{S_o} \tag{17}$$

Combining equations 16 and 17 we get

$$\frac{1}{f} = \frac{1}{S_o} + \frac{1}{m \cdot S_o} \tag{18}$$

and solving for S_o gives us

$$S_o = f + \frac{f}{m} \tag{19}$$

We can find S_i using Equation 17 and add S_o and S_i to find the total object-to-image distance. This set of equations is plotted in Figure 11 for a series of commonly available focal lengths.

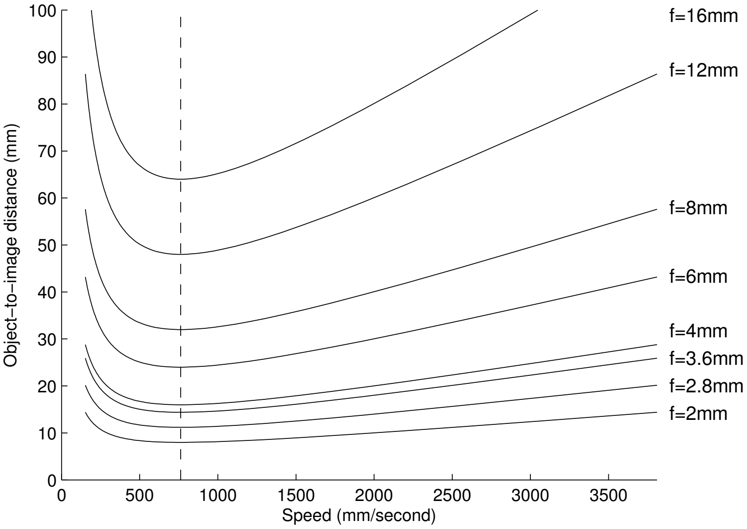


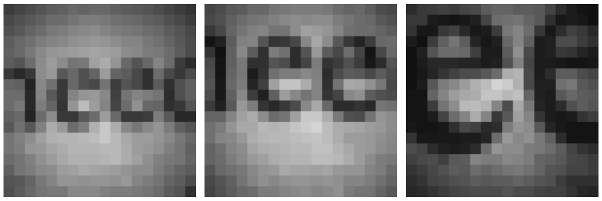
Figure 11: Plot of the object-to-image distance as a function of focal length and tracking speed. Each line represents a particular fixed focal length. The vertical dashed line shows a magnification of 1, which gives the rated tracking speed of 30 inches per second (762 $\frac{\text{mm}}{\text{second}}$).

Figure 12 compares the images captured from the sensor using a constant height above the surface and varying focal lengths. Lower focal lengths provide a wider view; high focal lengths provide a "zoomed in" view.

Based on Figure 11, a focal length of 6 mm is most appropriate for the ADNS-5060. The ADNS-2610 used in the experiments has a tracking speed of only 12 inches per second (304.80 $\frac{\text{mm}}{\text{second}}$), so a larger magnification is necessary. A focal length of 3.6 mm was selected for testing.

C. Mouse motion simulations

This section describes some example motion simulations developed in MATLAB. In each diagram, the large rectangle represents the robot base, and each trapezoid represents a mouse sensor where the "front" of the sensor

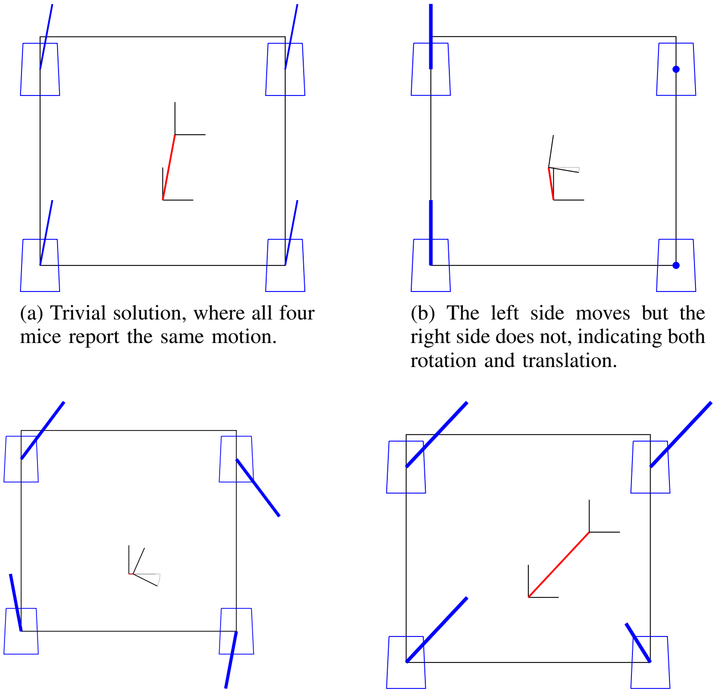


(a) 3.6 mm (b) 4.0 mm (c) 6.0 mm

Figure 12: Comparison of focal lengths. The sensor PCB was placed 1.0 inches above a piece of paper with 6-point type, and each lens was focused appropriately.

faces the narrow end of the trapezoid. The lines indicate the motion sensed by each mouse, and the line from the center of the robot shows the calculated motion vector. X and Y axis reference marks are drawn on the base to indicate rotation.

The simulations in Figure 13 show only four mice for simplicity, but the code can handle any number of mice.



(a) Trivial solution, where all four mice report the same motion.

(b) The left side moves but the right side does not, indicating both rotation and translation.

(c) Each sensor reports a different value, indicating rotation with very little translation.

(d) Example of sensor weighting: the lower-right sensor was given a weight of zero, so its measurement is ignored.

Figure 13: Example motion simulations calculated and drawn with MATLAB.