Mobile Robot Localization Using Optical Flow Sensors

Sooyong Lee and Jae-Bok Song

Abstract: Open-loop position estimation methods are commonly used in mobile robot applications. Their strength lies in the speed and simplicity with which an estimated position is determined. However, these methods can lead to inaccurate or unreliable estimates. Two position estimation methods are developed in this paper, one using a single optical flow sensor and a second using two optical sensors. The first method can accurately estimate position under ideal conditions and also when wheel slip perpendicular to the axis of the wheel occurs. The second method can accurately estimate position even when wheel slip parallel to the axis of the wheel occurs. Location of the sensors is investigated in order to minimize errors caused by inaccurate sensor readings. Finally, a method is implemented and tested using a potential field based navigation scheme. Estimates of position were found to be as accurate as dead-reckoning in ideal conditions and much more accurate in cases where wheel slip occurs.

Keywords: Mobile robot localization, mobile robot navigation, optical flow, redundant sensing.

1. INTRODUCTION

Accurate position estimation is a key component to the successful operation of most autonomous mobile robots. In general, there are three phases that comprise the movement sequence of a mobile robot: localization, path planning, and path execution. During localization, the position and orientation in the reference coordinate system is determined using external sensors. A path is then updated from the current position to the goal if the current robot position is deviated from the pre-planned path. The final phase is the execution of the planned path. The movement sequence is repeated so that the robot will remain on course towards the goal.

Localization can be further decomposed into two types, absolute and relative [1]. Absolute localization relies on landmarks, maps, beacons, or satellite signals to determine the *global* position and orientation of the robot. Relative localization (or intermediate estimation) is usually used during movement, because absolute localization methods are more time consuming.

Manuscript received June 24, 2004; revised November 8, 2004; accepted November 8, 2004. Recommended by Editor Keum-Shik Hong. This research was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Science and Technology of Korea.

Sooyong Lee is with the Department of Mechanical and System Design Engineering, Hongik University, 72-1 Sangsu-Dong, Mapo-Gu, Seoul 121-791, Korea (e-mail: sooyong@hongik.ac.kr).

Jae-Bok Song is with the Department of Mechanical Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea (e-mail: jbsong@korea.ac.kr).

Commonly, dead-reckoning (open-loop estimation) is used for intermediate estimation of position during path execution. Dead-reckoning is often used when wheel encoders are available for drive wheel position measurement. However, due to errors in kinematic model parameters, wheel slip, or an uneven surface, poor position estimates may occur. Poor estimates in position during path execution require more frequent localization, incurring extra overhead and possibly slowing the movement of the robot. A worse scenario is one where poor estimates would cause a collision, impeding the operation of the robot. It is therefore important to minimize errors in estimated position during the path execution phase.

Dead-reckoning usually fails (causes poor estimates) in the presence of wheel slip. However, using the methods described in this paper, accurate estimates of position can be maintained even when wheel slip occurs. Indeed, when either systematic errors (errors related to robot properties or parameters) or non-systematic errors (random errors caused by the environment) occur, dead-reckoning usually fails to accurately estimate position. Dead-reckoning only produces accurate estimates when all kinematic constraints are upheld. Two methods are developed, implemented and tested using inexpensive optical flow sensors. The placement of the optical sensors affects estimation errors. An optimal placement scheme (in the sense of minimizing estimation errors) is proposed. Additionally, path planning is discussed and implemented. Optical flow sensors can be used to accurately estimate the position of the robot by utilizing the methods described.

In most movement schemes, dead-reckoning errors are an accepted part of the movement sequence. These

errors are usually counteracted by making frequent localization "stops". This is unfortunate, because in many cases dead-reckoning proves to be inaccurate. If a more accurate method were available, less intermediate localization would be necessary. This would in turn free computational resources that could be used to accomplish higher level tasks.

Other researchers have implemented similar dead-reckoning correction techniques. A [1] towed robot (called a trailer), which has accurate wheel encoders and a rotary encoder on the connection link, is used to determine the relative movement and direction of the trailer with respect to the robot. This information allows an accurate estimated position to be maintained. This method reduced dead-reckoning errors by an order of magnitude or more. However, the added bulk of the trailer can complicate the movement of the robot making it difficult to navigate in close quarters, especially when moving backwards.

Another method of correcting dead-reckoning errors in navigation uses optical flow. In [2], optical flow was used to aid in the navigation of an omni-directional robot. A CCD camera was positioned at a 45^{o} downward angle to the ground in front of the robot. The optical flow obtained was combined with the results of dead-reckoning via maximum likelihood technique. The method used to calculate optical flow is quite complex, requiring a large number of computations to obtain good results.

GPS has many applications to mobile robot navigation [3]. GPS has been used to correct position estimations by adjusting kinematic parameters. In [4] GPS was used to correct for heading and step size in a pedestrian navigation system. When GPS is available the parameters are adjusted such that when GPS service is unavailable, a good estimate of position is maintained. This method is easily portable to mobile robots. However, GPS accuracy is limited, so when fine positional control is necessary it can prove ineffective. GPS is further limited by the fact that it will only work in outdoor environments where line-of-sight to at least 3 satellites is possible.

Barshan and Durant-Whyte developed inertial navigation methods for mobile robots in [5]. A series of solid state gyros, accelerometers and tilt sensors were employed in conjunction with an extended Kalman filtering method to estimate position and orientation. Using accelerometers to determine position and orientation, however, has several draw-backs such as 1-8 cm/s drift rate. Also, the minimum detectable acceleration can sometimes be too large to detect small motion. Inertial methods can be quite computationally intensive, expensive and complex.

2. SENSOR INTERPRETATION

In this paper, commercial optical mice were used for optical flow measurement. These sensors are inexpensive, reliable, accurate, and very fast. Two methods of estimating the location and orientation of the robot are investigated in this section. The first uses a single sensor and a constraint on the kinematics of the robot. The second method uses two sensors and no kinematic constraints.

2.1. Single optical sensor

Unless the optical flow sensor is placed at the point of interest on the robot, measurements directly from the optical sensor will not be useful. Additionally, because the optical sensor only provides displacements in the x and y directions, information about the angular displacement of the robot is lost. As a result, another method must be used to determine the angular velocity of the robot. Fig. 1 shows the coordinates used for the rigid body method.

The robot can be viewed as a rigid-body where the velocity at the sensor ($\Delta x/\Delta t$ and $\Delta y/\Delta t$) is known. The kinematic constraints of a differential drive robot allow the calculation of movement, given this velocity. Specifically, the center of the robot is assumed to move only in a direction perpendicular to the wheel axis. The following equation relates the velocity of a point on the axis of rotation of a rigid body (robot center) with a point not on that axis of rotation (the sensor location):

$$V_r = V_s + \omega_r \times r_{s/r}, \tag{1}$$

where V_r is the velocity of the center of the robot, V_s is the sensor velocity, ω_r is the angular velocity of the robot, and $r_{s/r}$ is the vector from the location of the sensor to the robot center (Fig. 1). Rewriting in matrix form:

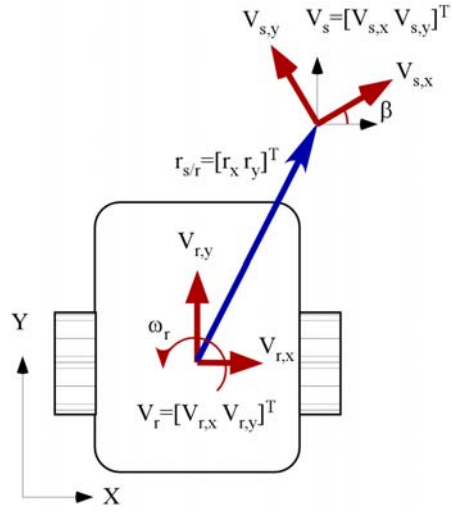

Fig. 1. Rigid-body model for sensor interpretation.

$$\begin{bmatrix} V_{r,x} \\ V_{r,y} \end{bmatrix} = \begin{bmatrix} V_{s,x} \\ V_{s,y} \end{bmatrix} + \begin{bmatrix} -\omega_r r_y \\ \omega_r r_x \end{bmatrix}. \quad (2)$$

It should be noted that the sensor may not be oriented along the x and y robot coordinates. Denoting the orientation of the sensor as β, and the sensor coordinates as $\tilde{x}$ and $\tilde{y}$, the following transformation is made to robot coordinates:

$$\begin{bmatrix} V_{s,x} \\ V_{s,y} \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{bmatrix} \begin{bmatrix} V_{s,\tilde{x}} \\ V_{s,\tilde{y}} \end{bmatrix}. \quad (3)$$

With the assumption that $V_{r,x} = 0$, ω_r and $V_{r,y}$ are obtained from the following two equations:

$$\omega_r = -\frac{V_{s,x}}{r_y}, \quad (4)$$

$$V_{r,y} = V_{s,y} - \omega_r r_x = V_{s,y} + \frac{r_x}{r_y} V_{s,x}. \quad (5)$$

The robot's velocity can now be easily translated to the *global* coordinate system as follows:

$$\theta_r = \int (\omega_r) dt, \quad (6)$$

$$X_r = \int (-V_{r,y} \sin(\theta_r)) dt, \quad (7)$$

$$Y_r = \int (V_{r,y} \cos(\theta_r)) dt. \quad (8)$$

A similar approach to using the kinematic constraint $V_{r,x} = 0$ is to use the weighted pseudo-inverse. This method gives one possible solution to the underdetermined case, where three variables are determined from only two equations. Rewriting the rigid body equation in matrix form:

$$\begin{bmatrix} V_{s,x} \\ V_{s,y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & r_y \\ 0 & 1 & -r_x \end{bmatrix} \begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix}. \quad (9)$$

The underdetermined pseudo-inverse takes the form:

$$\begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix} = W^{-1} A^T \left(A W^{-1} A^T \right)^{-1} \begin{bmatrix} V_{s,x} \\ V_{s,y} \end{bmatrix}, \quad (10)$$

where W is a weighting matrix and

$$A = \begin{bmatrix} 1 & 0 & r_y \\ 0 & 1 & -r_x \end{bmatrix}. \quad (11)$$

If we heavily weight the velocity in the x direction to be small, nearly an identical solution to the constrained case emerges. A suitable weighting matrix follows:

$$W = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

Fig. 11 in Section 4 shows how varying the weighting matrix changes the apparent movement of the robot using experimental data. The single sensor method will give better results when wheel slip perpendicular to the wheel axis occurs rather than dead-reckoning. In fact, as long as the kinematic constraint $V_{r,x} = 0$ is not violated, the method will supply accurate position estimates. If multiple sensors are used, the constraint can be removed, allowing precise positional estimates in all kinematic conditions. This is the topic of the following subsection.

2.2. Multiple optical sensors

Given a second sensor, there are at least two approaches that could be used to interpret the data. First, the additional sensor could be viewed as a redundant sensor, which would lead to data fusion methods. Data fusion should give better results than a single sensor given data error. However, it would still provide inaccurate results if the kinematic constraint, $V_{r,x} = 0$ were violated.

Alternatively, the velocity information from the second sensor can be used to determine the motion of the rigid-body without the constraint, $V_{r,x} = 0$, that was necessary using only a single sensor.

Looking again at the rigid-body model, two sensors at different locations give ample information to determine the motion of the rigid-body without any kinematic constraints. In this case, the rigid-body model leads to the following four equations:

$$V_{r,x} = V_{s1,x} - \omega_r \cdot r_{s1,y}, \quad (13)$$

$$V_{r,y} = V_{s1,y} + \omega_r \cdot r_{s1,x}, \quad (14)$$

$$V_{r,x} = V_{s2,x} - \omega_r \cdot r_{s2,y}, \quad (15)$$

$$V_{r,y} = V_{s2,y} + \omega_r \cdot r_{s2,x}, \quad (16)$$

where $V_{r,x}$ and $V_{r,y}$ are the velocities of the center of the robot, ω_r is the angular velocity of the robot, $V_{si,x}$ and $V_{si,y}$ are the sensor velocities and $r_{si,x}$ and $r_{si,y}$ are the x and y distances from the i^{th} sensor position to the robot center.

To solve this system of equations for the angular

and linear velocities of the robot, the system is written as

$$\begin{bmatrix} 1 & 0 & -r_{s1,y} \\ 0 & 1 & r_{s1,x} \\ 1 & 0 & -r_{s2,y} \\ 0 & 1 & r_{s2,x} \end{bmatrix} \cdot \begin{bmatrix} V_{r,x} \\ V_{r,y} \\ \omega_r \end{bmatrix} = \begin{bmatrix} V_{s1,x} \\ V_{s1,y} \\ V_{s2,x} \\ V_{s2,y} \end{bmatrix}. \quad (17)$$

This overdetermined system can be solved using the pseudo-inverse [6]. Similar to one sensor case, the robot's velocities are translated to the global coordinates as

$$\theta_r = \int (\omega_r)\, dt\,, \quad (18)$$

$$X_r = \int \left(V_{r,x} \cos\theta_r - V_{r,y} \sin\theta_r \right) dt\,, \quad (19)$$

$$Y_r = \int \left(V_{r,x} \sin\theta_r + V_{r,y} \cos\theta_r \right) dt\,. \quad (20)$$

The least squares overdetermined solution can also be weighted similarly to the underdetermined case. However, the weightings would be of the sensed velocities $\left(V_{s1,x}, V_{s1,y}, V_{s2,x}, V_{s2,y}\right)$. It is not readily apparent which of these velocities should receive a higher importance than the other. This avenue of research was not investigated.

Although multiple sensors do not increase accuracy when compared to a single sensor, the kinematic constraint can now be removed. This yields a powerful method to determine intermediate estimates of the robot position and orientation.

3. SENSOR PLACEMENT

3.1. Optimal sensor location

Another important issue to consider is minimizing an error. Using the method described in the preceding section, several possible sources of an error are present. These include errors in the position and orientation of the sensors and errors in the sensed velocities. Minimizing these errors will lead to better estimates of robot position.

In order to determine the best position for a single sensor, the error in the measurement of the velocity must be minimized. The maximum absolute deviation of a function $F\left(x_o, x_1, \cdots, x_n\right)$ is defined as:

$$dF = \left|\frac{\partial F}{\partial x_0}\right| \cdot dx_0 + \cdots + \left|\frac{\partial F}{\partial x_n}\right| \cdot dx_n\,. \quad (21)$$

Using this definition and the previously given single sensor rigid-body model, the maximum absolute deviations are as follows:

$$d\omega_r = \left|-\frac{1}{r_y}\right| \cdot dV_x + \left|-\frac{V_x}{r_y^2}\right| \cdot dr_y\,, \quad (22)$$

$$dV_r = \left|\frac{r_x}{r_y}\right| \cdot dV_x + |1| \cdot dV_y + \left|\frac{V_x}{r_y}\right| \cdot dr_x + \left|\frac{V_x \cdot r_x}{r_y^2}\right| \cdot dr_y\,. \quad (23)$$

Errors in sensor orientation are ignored because calibration of the sensors can easily be carried out using a straight line path. Sensor orientation can be tuned such that the sensed path closely matches the actual path. Additionally, the position of the sensor is known within a few millimeters and is therefore assumed to be exactly known. (In experiments, the sensors were placed at 20cm from the center of the robot, and measurement errors are on the order of 0.1cm.) This leads to a further simplified equation:

$$d\omega_r = \left|-\frac{1}{r_y}\right| \cdot dV_x\,, \quad (24)$$

$$dV_r = \left|\frac{r_x}{r_y}\right| \cdot dV_x + |1| \cdot dV_y\,. \quad (25)$$

Plots of first order deviations of these equations are shown in Figs. 2 and 3.

As can be seen, the optical sensor should be located as far away as possible from the center of the robot along the *y*-axis, because the global minima is located at $r_x = 0$. Ideally, the error in the determination of the robot's position will be zero for $r_x = 0$ and $r_y = \infty$. However, this is impractical both physically and mathematically. Looking at (4) if $r_y = \infty$ then ω

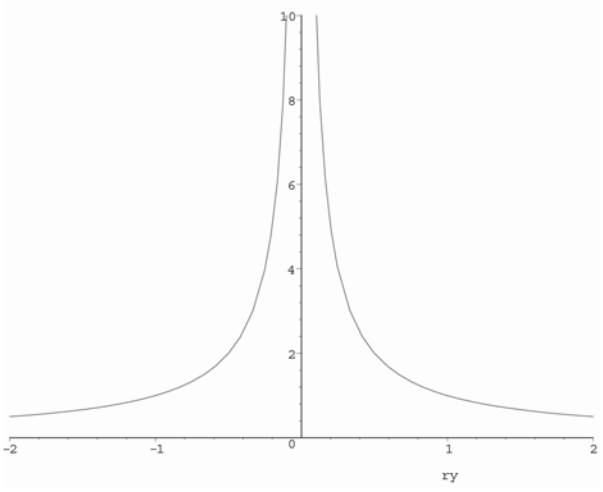

Fig. 2. Error of ω_r versus r_y.

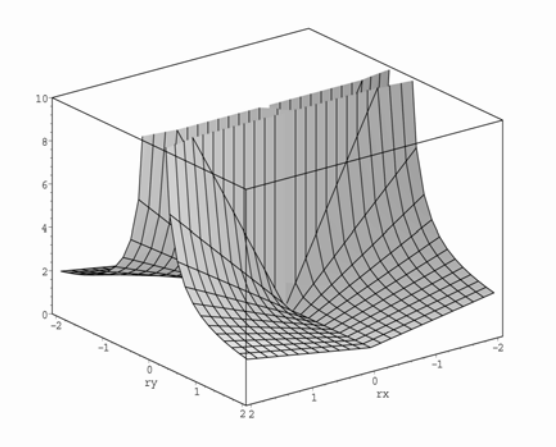

Fig. 3. Error of V_r versus r_x, r_y.

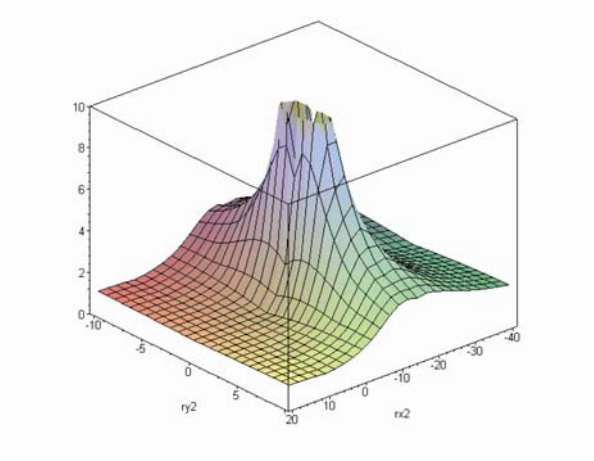

Fig. 5. Error of $V_{r,y}$ versus sensor 2 position.

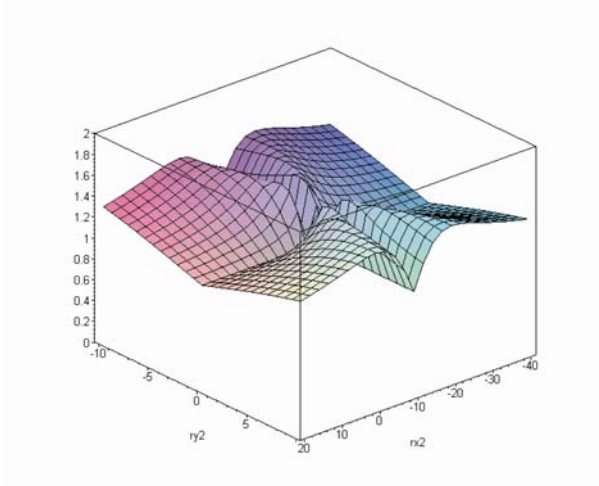

Fig. 4. Error of $V_{r,x}$ versus sensor 2 position.

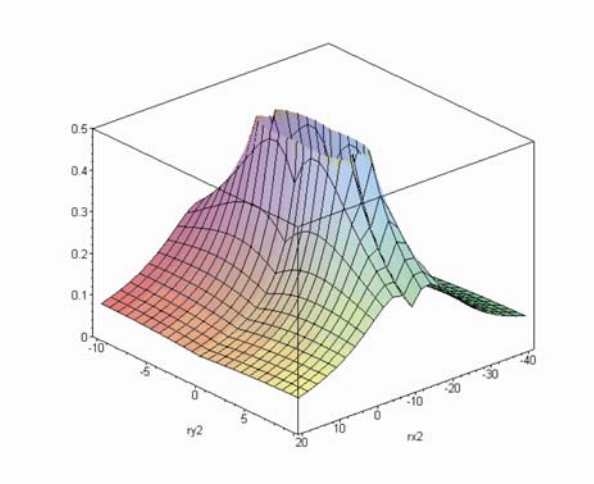

Fig. 6. Error of ω_r versus sensor 2 position

always equals zero, which is unacceptable.

To find the optimal position using two sensors, again the maximum absolute deviations of the functions are used. Because of the large number of parameters in each equation, visualizing the deviations is difficult. Assuming again that the positions of the sensors relative to the robot's center can be measured exactly, the only error present is in the sensor velocities.

The first observation is that there is a range of positions where the errors become minimal. To minimize the error in the measurement of the angular velocity (ω_r) the sensors should be located as far from the center of the robot as possible. Looking at Fig. 5, if the error in *y*-direction velocity $V_{r,y}$ is minimal, the *x*-positions of the sensors have the same value with different signs. The same is observed for the error of the velocity in *x*-direction $V_{r,x}$. The first order deviations of V_r and ω_r when sensor 1 is placed at $r_{1,x} = 10$ and $r_{1,y} = 0$ are shown in Figs. 4, 5, and 6.

The location of the sensors is important because the sensors provide only the linear velocities *dx* and *dy*, but not the angular velocity, $d\theta$. The change in orientation is clearly important when updating the position. Therefore, any errors in *dx* or *dy* will greatly affect the accuracy of either rigid-body method. As indicated in Fig. 3, the larger the r_y, the smaller the error in ω_r. However, it is impractical to have a long arm with a sensor mounted because the arm may be bent and also it may cause collision with nearby obstacles. The *x*-position of a sensor is not as sensitive as the *y*-position either in error of V_r nor ω_r. Therefore, the front sensor is attached at $r_{s1,x} = 40\text{mm}$ and $r_{s1,y} = 60\text{mm}$. The rear sensor is at the symmetric location at $r_{s1,x} = -40\text{mm}$ and $r_{s2,y} = -60\text{mm}$.

3.2. Ground clearance

Another important aspect regarding the placement of the optical sensors is focal length. Commercial optical flow sensors have a narrow band of operation

corresponding to the focal length.

This is because in order to properly determine the movement of the sensor, the correspondence between the number of pixels in the image and a real measurable length must be known. Changing the focal length will vary this relationship, causing errors. If this constant focal length is not maintained, errors in distance will accumulate quickly.

The sensors used in the experimental section of this paper have a nominal focal length of 2.4mm. Accurate data are obtained only if the focal length is properly maintained. If the focal length is increased even a few tenths of a millimeter, inaccurate *dx* and *dy* data results. This was overcome by forcing the sensors to the ground, which tended to significantly increase friction. However, an increase in friction can cause the wheels to slip, making it difficult to move. It is desirable to reduce friction and at the same time maintain the 2.4mm focal length recommended by the chip manufacturer.

4. EXPERIMENTAL VALIDATION

4.1. Sensor specifications

The optical flow sensors used were the Agilent ADNS-2051. This sensor allows fast, accurate, optical sensing of microscopic images. The sensor is capable of 800 counts per inch while moving at up to 14 inches per second. Successive images are used to calculate the Δx and Δy values at up to 2300 frames per second.

A microcontroller is used that communicates directly with the ADNS-2051. This chip communicates with the ADNS-2051, setting modes and communicating with the PC or other devices using the PS-2 standard.

The optical flow sensor uses successive images to interpret the movement between images. Images are taken of a point on the ground directly beneath the sensor. The images are 16×16 pixels and represent a microscopic area.

4.2. Robot specifications

In order to test the algorithms presented in this paper, a robot system was designed and built. A modular design approach was implemented. Each module is responsible for one part of the overall operation of the robot. This type of design permits testing of various configurations, as well as allows easy replacement of any single system.

4.2.1 Hardware

The mobile robot includes a 12 inch, round plastic base, two 1 amp 12 Volt DC geared motors, two castor wheels, two 6 inch diameter rubber wheels, and two encoders attached to the drive wheel axles.

The differential driven wheels allow turns to be made in place. This symmetric design offers high maneuverability. The maximum straight line speed is approximately 78 feet per minute (50 revolutions per minute of the drive wheels). The robot is shown in Fig. 7. Two optical flow sensors are used but only the front one is shown in the figure.

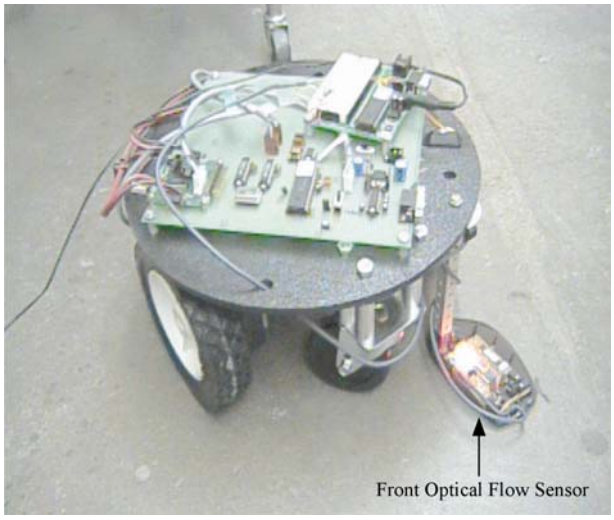

Fig. 7. Mobile robot with optical sensors.

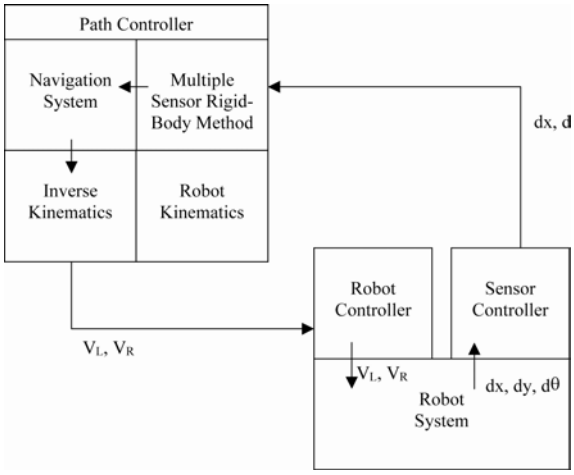

Fig. 8. Interaction between objects in Java.

The robotic base consists of a microcontroller circuit with various communication and electronic chips, two motors, two 1 amp motor controllers, two HEDS-5500 encoders and two Agilent HCTL-2020 encoder-decoder chips.

The sensor system consists of a controller and two optical flow sensors. The controller includes a microcontroller, serial communication module. The microcontroller interfaces with the two sensors and transmits the data to the PC using serial communication.

4.2.2 Software

The software consists of three independent systems that must operate congruously in order to function.

These systems are the robot system, the sensor system and a PC based controller. The robot and sensor systems are based on PIC 16F877 microcontrollers running at 10 MHz. The PC controller is implemented using Java. The operation of the system is outlined in Fig. 8.

Software design requirements for the robotic base were to control the speed of two motors to desired velocities and output the position (or velocity) of both the left and right wheels. The robotic base receives wheel commands from the PC and controls the difference in wheel positions between successive interrupts.

4.3. Intermediate localization

Experiments were performed comparing intermediate localization (positional estimates) using dead-reckoning, the single sensor and the multiple sensor rigid-body method. Experiments were performed where no kinematic violations occurred to test the accuracy of localization. Experiments were also performed where kinematic violations were forced.

In experiments where kinematic constraints were upheld ($V_{r,x} = 0$ and no wheel slip) as in Fig. 9, little performance difference can be seen between the methods. There are three different estimated robot paths in this figure. The first one is from the encoder data, (so called *dead-reckoning*). The robot configuration (position/orientation of the robot) is estimated from the robot forward kinematics with the measured encoder data at each sampling time. The second one is estimated with an optical flow sensor that is fixed in front of the robot, from (5)-(9). The third one is with two optical flow sensors using (17)-(20). When the kinematic constraints were intentionally violated with an outside disturbance as in Fig. 10, however, only the multiple sensor method gave good results.

The results of the single sensor method are interesting. It would seem that when kinematic disturbances are encountered, the method gives unreliable results. However, this should be expected because the kinematic constraint is required for a solution to exist in the first place.

Additionally, if the underdetermined pseudo-inverse solution is used (see Section 2), there is little accuracy gained. This solution does not use the constraint, but simply weights the solution so that velocity in the *x* direction is reduced. This method was investigated to determine if an accurate estimate would result if the constraint $V_{r,x} = 0$ is removed. Fig. 11 shows how changing the weighting of $V_{r,x}$ effects the estimated path compared to the dual-sensor method.

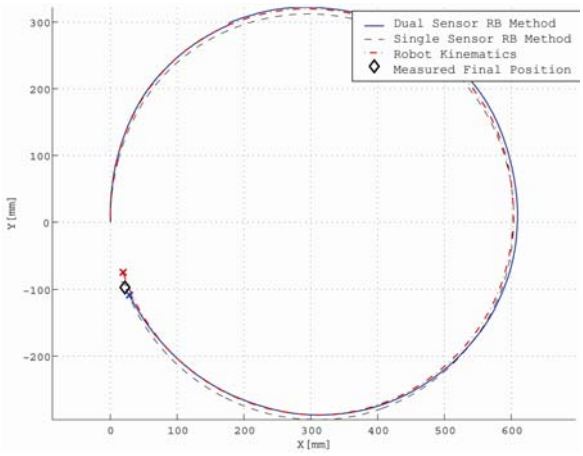

Fig. 9. Localization methods compared with kinematics upheld.

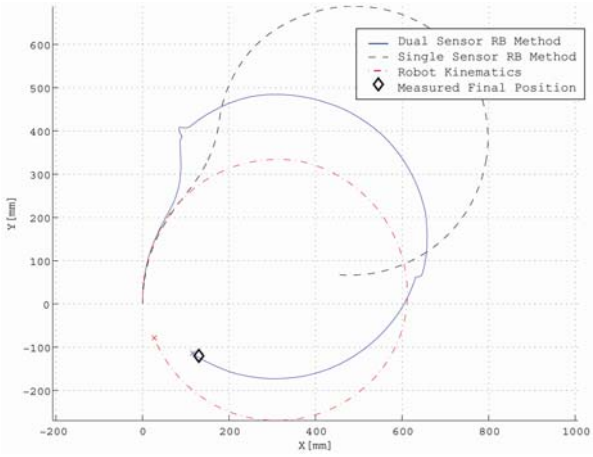

Fig. 10. Localization methods compared with kinematic constraints.

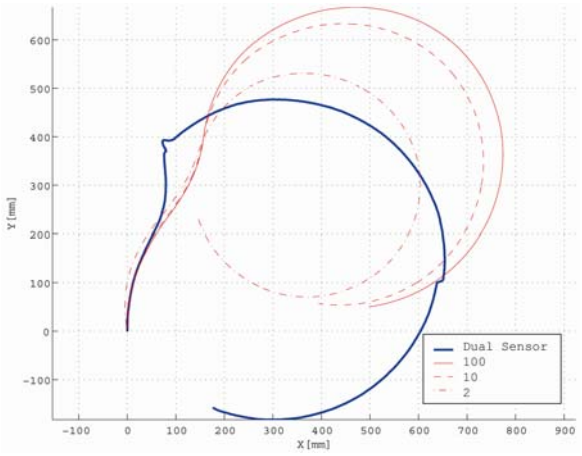

Fig. 11. Weighted pseudo-inverse for the underdetermined single sensor case.

4.4. Potential field based navigation

In order to test the effectiveness of the intermediate

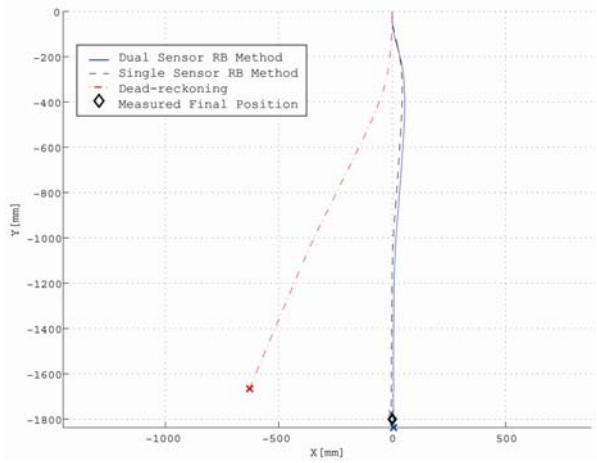

Fig. 12. Navigation with wheel slip.

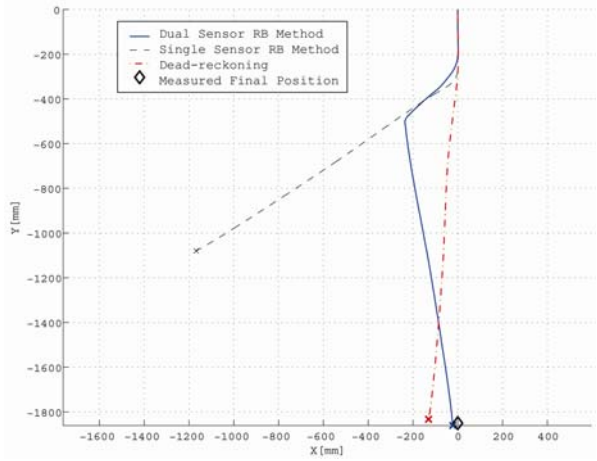

Fig. 13. Navigation with kinematic violation.

estimation method described, navigation tests were performed. The single and multiple sensor rigid-body models and dead-reckoning were compared under various conditions.

A potential field based approach was used to generate velocity commands for the mobile robot to follow. An online method was used where the desired V_r and ω_r were generated depending upon the current robot location/orientation and the location of the goal point.

A slight modification to the conventional potential field based method was used. This is because of the non-holomonic constraints that are present with a differential drive mobile robot. The method was to make the robot turn towards the goal point before approaching it. The following relationships were used:

$$V_{r,desired} = k_v V_{potential} \cos(\alpha), \quad (26)$$

$$\omega_{r,desired} = k_\omega V_{potential} \alpha, \quad (27)$$

where k_v and k_ω are proportional constants and $V_{potential}$ is the potential velocity generated by the potential field method. Using this approach, results were satisfactory. Also, obstacles were assumed not to be present, but could easily be added.

In tests where no wheel slip or kinematic violations occurred, all three methods were similarly accurate. The largest performance difference of the multiple-sensor method can be seen in the presence of kinematic violations, such as wheel slip. Fig. 12 indicates an actual test run where wheel slip was forced. As one can see, the robot has a sudden change in orientation at the beginning of movement. The multiple-sensor method is able to detect this movement, which allows the potential field based navigation method to adjust its course accordingly. Even when kinematic constraints are violated, as in Fig. 13 where the robot was pushed off course, the multiple sensor method still detects the proper change in position and orientation so that the robot successfully reaches its goal.

5. CONCLUSIONS

Using optical flow sensors, we have derived a method that can accurately provide intermediate position estimates. The location of the sensors has been investigated in order to minimize potential errors introduced by incorrect sensor readings. Experimentally, the multiple sensor rigid-body technique has proven to be more accurate and more robust than dead-reckoning. Both systematic and non-systematic errors can be detected and a good estimate of location and orientation can be maintained.

REFERENCES

[1] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 6, pp. 869-880, 1996.

[2] K. Nagatani, S. Tachibana, M. Sofne, and Y. Tanaka, "Improvement of odometry for omnidirectional vehicle using optical flow information," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 468-473, October 2000.

[3] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory And Application*, vol. 2, American Institute of Aeronautics and Astronautics, 1996.

[4] V. Garaj, F. Cecelja, and W. Balachandran, "A method for dead reckoning parameter correction in pedestrian navigation system," *Proc. of the 18th IEEE Instrumentation and Measurement Technology Conference*, vol. 3, pp. 1554-1558, May 2001.

[5] B. Barshan and H. F. Durant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 11, no. 3, pp. 328-342, June 1995.
[6] W. L. Brogan, *Modern Control Theory*, Prentice Hall, Inc., Upper Saddle River, New Jersey, 1991.
[7] G. Ramirez and S. Zeghoul, "A new local path planner for nonholonomic mobile robot navigation in cluttered environments," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 2058-2063, April 2000.
[8] B. Horn, *Robot Vision*, MIT Press, 1986.
[9] L. G. Shapiro and G. C. Stockman, *Computer Vision*, Prentice-Hall Inc., 2001.
[10] N. Molton, S. Se, J. M. Brady, D. Lee, and P. Probert, "A stereo vision-based aid for the visually impaired," *Image and Vision Computing*, vol. 16, no. 4, pp. 251-263, April 1998.

Sooyong Lee received the B.S. and M.S. degrees in Mechanical Engineering from Seoul National University, Seoul, Korea in 1989, and 1991, respectively, and the Ph.D degree from MIT, Cambridge, MA, in 1996. He worked as a Senior Research Scientist at KIST and then as an Assistant Professor in the Department of Mechanical Engineering at Texas A&M University. He joined Hongik University, Seoul, Korea in 2003 and is currently an Assistant Professor in the Mechanical and System Design Engineering Department. His current research includes mobile robot localization and navigation, and active sensing.

Jae-Bok Song received the B.S. and M.S. degrees in Mechanical Engineering from Seoul National University, Seoul, Korea, in 1983 and 1985, respectively, and the Ph.D. degree in Mechanical Engineering from MIT, Cambridge, MA, in 1992. From 1992 to 1993 he worked as a Research Associate at MIT. He joined the faculty of the Department of Mechanical Engineering, Korea University, Seoul, Korea in 1993, where he has been a Full Professor since 2002. His current research interests are robot navigation, and design and control of the robotic systems including haptic devices and field robots.