

# Depth Estimation for Autonomous Robot Navigation: A Comparative Approach

Sotirios Ch. Diamantas  
Intelligence, Agents, Multimedia Group  
School of Electronics and  
Computer Science  
University of Southampton  
Southampton, SO17 1BJ, UK  
Email: sd04r@ecs.soton.ac.uk

Anastasios Oikonomidis  
Centre for Risk Research  
School of Management  
University of Southampton  
Southampton, SO17 1BJ, UK  
Email: ao1@soton.ac.uk

Richard M. Crowder  
Intelligence, Agents, Multimedia Group  
School of Electronics and  
Computer Science  
University of Southampton  
Southampton, SO17 1BJ, UK  
Email: rmc@ecs.soton.ac.uk

**Abstract**—Depth estimation has long been a fundamental problem both in robotics science and in computer vision. Various methods have been developed and implemented in a large number of applications. Despite the rapid progress in the field the last few years, computation remains a significant issue of the methods employed. In this work, we have implemented two different strategies for inferring depth, both of which are computationally efficient. The first one is inspired by biology, that is optical flow, while the second one is based on a least squares method. In the first strategy, we observe the length variation of the optic flow vectors of a landmark at varying distances and velocities. In the second strategy, we take snapshots of a landmark from different positions and use a least squares approach to estimate the distance between the robot and a landmark. An evaluation of the two different strategies for various depth estimations has been deployed and the results are presented in this paper.

**Index Terms**—optical flow; least squares; depth estimation; visual navigation; mobile robotics

## I. INTRODUCTION

Depth estimation is an important task for a large number of robot navigation techniques. In this work, we estimate the distance between a robot and a landmark with the view to support the localisation and mapping problem, a problem which is at the heart of today's mobile robotic systems. This type of work has been developed in order to be integrated into a self-localisation system as is in [1] or similar type of systems as described in [2], [3]. These type of systems make no use of dead reckoning or odometry information and the position of the robot is estimated by means of a laser range finder. Thus, for a robot to be able to estimate its position in an environment, corners and other related features are extracted from the laser range scans. There are, however, cases where a robot navigates in an environment that does not contain these type of features. Such type of environments are the corridors or environments without geometrical objects within the range of the laser scanner. In this paper, the distance of the robot to a landmark is used as a means to calculate the translational distance the robot has covered in that environment while there is no presence of corner features. We, therefore, have adopted two different strategies for inferring the distance between the robot and a landmark and compare the results

from both approaches. The first one, that is optical flow, has been inspired by biology and the way insects, in particular honeybees, navigate. The second one is a least squares method that calculates the position of a landmark based on different snapshots taken during the navigation process of the robot.

Optical flow, that is the rate of change of image motion in the retina or a visual sensor, is extracted from the motion of an agent. Optical flow has extensively been used thus far for obstacle avoidance and centring a robot in corridor-like environments [4], [5], [6], [7]. Optical flow for depth estimation has been used in [8], [9] and a least squares approach for inferring depth appears in [10], [11]. Furthermore, a large amount of effort has been focussed on using total least squares [12], [13], least squares [14], or constrained total least squares methods [15] for calculating the optical flow field.

This paper is comprised of five sections. The following section, Section II, presents the background work while Section III presents the methodology of the two different strategies for depth estimation. Section IV presents and discusses the results from the two strategies. Finally, Section V epitomises the paper with a discussion on the conclusions drawn from this work.

## II. BACKGROUND WORK

This section describes the mathematics that underlie the Lucas-Kanade (LK) algorithm [16] which has been employed in this research work. In order for the optical flow algorithms to perform well, some suitable images need to be chosen. This suitability refers to images that have high texture and contain a multitude of corners. Such images have strong derivatives and, when two orthogonal derivatives are observed then this feature may be unique, and thus, good for tracking. Tracking a feature refers to the ability of finding a feature of interest from one frame to a subsequent one. Tracking the motion of an object can give the flow of the motion of the objects among different frames. In Lucas-Kanade algorithm corners are more suitable than edges for tracking as they contain more information. For the implementation of the LK algorithm the [17] library has been used.

The optical flow algorithm of Lucas-Kanade presupposes three main criteria to produce satisfactorily results. These are:

- 1) Brightness constancy. The brightness of a pixel does not change from frame to frame, that is  $I(x, y, t) = I(x + u, y + v, t + 1)$ .
- 2) Temporal persistence or small movements. The motion of the object that is tracked moves smoothly from frame to frame, that is  $I_x u + I_y v + I_t = 0$ , where  $v, u$  are the  $x, y$  components of the velocity  $\vec{u}$ .
- 3) Spatial coherence. Neighbouring points of a pixel that belong to the same surface have typically similar motion, and project to nearby points on the image plane.

The equation in the second criterion is an under constrained equation since it involves two unknowns for any given pixel and cannot be used to solve the motion of a pixel in the two dimensions. For this reason the third criterion is used as an assumption to solve the full motion of a pixel in the two dimensions. The third criterion assumes that the neighbouring pixels of any given pixel move coherently as they belong to the same object and project to the same image plane as the given pixel projects. Thus, for tackling the problem in case, the brightness values of the neighbouring pixels are considered and solve a system of linear equations [18]. Hence, if we take a window of  $5 \times 5$  pixels a system of 25 linear equations needs to be solved. However, if the window is too small the *aperture problem* may be encountered where only one dimension of the motion of a pixel can be detected and not the two-dimensional. On the other hand, if the window is too large then the spatial coherence criterion may not be met. Nevertheless, the system that needs to be solved following a window of  $5 \times 5$  pixels is expressed by (1)

$$\underbrace{\begin{bmatrix} I_x(p1) & I_y(p1) \\ I_x(p2) & I_y(p2) \\ \vdots & \vdots \\ I_x(p25) & I_y(p25) \end{bmatrix}}_{A = 25 \times 2} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\vec{u} = 2 \times 1} = - \underbrace{\begin{bmatrix} I_t(p1) \\ I_t(p2) \\ \vdots \\ I_t(p25) \end{bmatrix}}_{b = 25 \times 1} \quad (1)$$

The goal on the above system of linear equations is to minimise  $\|A\vec{u} - b\|^2$  where  $A\vec{u} = b$  is solved by employing least-squares minimisation as in (2),

$$(A^T A)\vec{u} = A^T b \quad (2)$$

where  $A^T A$ ,  $\vec{u}$ , and  $A^T b$  are equal to (3),

$$\underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{A^T A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{\vec{u}} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b} \quad (3)$$

and the solution to the equation is given by (4)

$$\vec{u} = \begin{bmatrix} u \\ v \end{bmatrix} = (A^T A)^{-1} A^T b. \quad (4)$$

If  $A^T A$  is invertible, i.e., no zero eigenvalues, it means it has full rank 2 and two large eigenvectors. This occurs in images

where there is high texture in at least two directions. If the area that is tracked is an edge, then  $A^T A$  becomes singular, that is (5),

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} -I_y \\ I_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5)$$

where  $-I_y, I_x$  is an eigenvector with eigenvalue 0. If the area of interest is homogeneous then  $A^T A \approx 0$  implying 0 eigenvalues. The pyramidal approach of the LK algorithm overcomes the local information problem at the top layer by tracking over large spatial scales and then as it proceeds downwards to the lower layers the velocity criteria are refined until it arrives at the raw image pixels.

### III. METHODOLOGY

This section describes the methodology followed for estimating depth using optical flow and least squares. For the optical flow strategy we used the 3D simulation environment in [19], while for the least squares strategy we employed the [20] 2D simulator. The programming environment of [21] was used for the analysis of the data. The purpose of both methods is to estimate the translational distance a robot has covered in a corridor-like environment due to the absence of ‘distinct’ features in laser scans. The translational distance a robot has travelled can be calculated using the trigonometric functions once the position of a landmark has been estimated.

#### A. Least Squares

For the estimation of depth using the least squares approach we track a visual landmark in the environment and infer the position of the landmark based on the orientation and position of the robot. A similar approach has been used by [11] to infer the starting position of the robot.

In this simulation experiment, we have taken various snapshots of a landmark at equally spaced time steps and measure the angle  $\theta$  between the robot position and the visual landmark. Every snapshot represents a linear equation and the solution of the linear system gives us the position of the landmark  $L$ . Thus, we form a set  $\Omega_i$  for every snapshot taken. Equations (6) and (7) show an example of two snapshots,

$$L \in \Omega_1 = \{h \in \mathbb{R}^2 \mid \underbrace{(v_1 - r_1)^T}_{\alpha_1} h = \underbrace{v_1^T \cdot r_1 - \|r_1\|^2}_{\beta_1}\} \quad (6)$$

$$L \in \Omega_2 = \{h \in \mathbb{R}^2 \mid \underbrace{(v_2 - r_2)^T}_{\alpha_2} h = \underbrace{v_2^T \cdot r_2 - \|r_2\|^2}_{\beta_2}\} \quad (7)$$

where  $r_1$  and  $r_2$  are the positions of the robot, and  $v_1$  and  $v_2$  are any two points on a line which is perpendicular to the line formed between the robot and the landmark, that is angle  $\theta$ . In essence, what we are trying to find is the  $\Omega_1 \cap \Omega_2 = \emptyset$ . Figure 1 shows a robot at three different positions,  $r^1$ ,  $r^2$ , and  $r^n$  at corresponding time intervals  $t_0$ ,  $t_1$ , and  $t_n$ , that also represent three different snapshots.

The following equations, (8) and (9), show the process for a  $n$  number of snapshots. Noise in the system is represented by variable  $\epsilon_i$ .

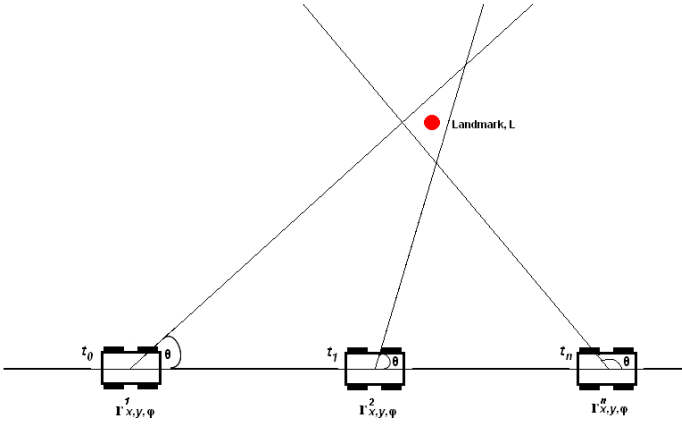


Fig. 1. Robot snapshots of the landmark  $L$  taken at different time steps and at different robot positions  $r$ .

$$\begin{aligned} h^T \alpha_1 + \epsilon_1 &= \beta_1 \\ h^T \alpha_2 + \epsilon_2 &= \beta_2 \\ &\vdots \\ h^T \alpha_n + \epsilon_n &= \beta_n \end{aligned} \quad (8)$$

$$h \in \operatorname{argmin} \sum_{i=1}^n (h^T \alpha_i - \beta_i + \epsilon_i)^2 \quad (9)$$

$$\underbrace{\left( \sum_{i=1}^n \alpha_i \alpha_i^T + \epsilon_i \right)}_C h = \underbrace{\left( \sum_{i=1}^n \alpha_i \beta_i \right)}_\gamma \quad (10)$$

$$h = C^{-1} \gamma \quad (11)$$

In (10)  $C$  is a  $2 \times 2$  matrix and  $\gamma \in \mathbb{R}^2$ . The position of the landmark is, thus, given by  $h$  where we are interested in its y-axis element, that is depth. In this experiment we have used  $n = 1000$  observations (also explained in the next subsection) with varying distances between the landmark and the robot. We have performed 10 trials for each observation in order to estimate the mean position of the landmark. In addition, the error,  $\epsilon_i$ , in  $\theta$  is uniformly distributed and it has been tested with three different deviations  $\pm 1^\circ$ ,  $\pm 3^\circ$ , and  $\pm 5^\circ$ .

### B. Optical Flow

In order for the robot to localise in an environment using optical flow vectors, a training data set of  $n = 1000$  observations has been implemented where a vector is observed at varying distances between the robot and the landmark, and at varying velocities. Moreover, the distances and velocities chosen to create the training set approximate the real distributions of distance and velocity when a robot navigates in an environment. Thus, a joint probability distribution has been created by two continuous and independent variables, that is distance,  $D$ , and velocity,  $E$ , and is expressed by (12),

$$f_{D,E}(d, e) = f_D(d) \cdot f_E(e) \quad \forall d, e. \quad (12)$$

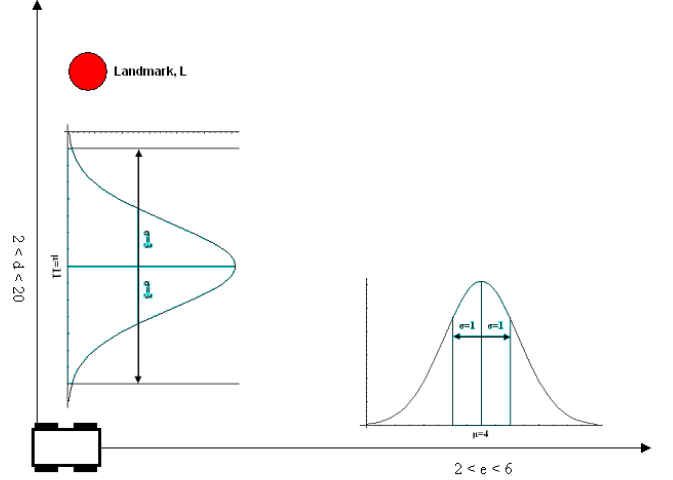


Fig. 2. Gaussian distributions used for modelling distance,  $D$ , and velocity,  $E$ , in the optical flow strategy. Their mean and standard deviation are  $\mu = 11$ ,  $\sigma = 3$  and  $\mu = 4$ ,  $\sigma = 1$ , respectively. The range of their values is  $2 < d < 20$  (metres) for distance and  $2 < e < 6$  (km/h) for velocity.

The velocity (in  $km/h$ ) and the distance (in *metres*) variables have been drawn from two Gaussian distributions with  $\mu = 4$ ,  $\sigma = 1$  and  $\mu = 11$ ,  $\sigma = 3$ , respectively. The  $n$  observations model the position of the optic flow vector in the plane under  $n$  varying distances and velocities. Figure 2 provides a pictorial representation of Gaussian distributions for velocity and distance.

One assumption that needs to be met in our method is that the majority of the vectors comprising a given landmark should have the same, or almost the same magnitude. In addition, the orientation of the camera on the robotic platform is perpendicular to the direction of motion so as a translational optical flow information is generated. During the testing of the optical flow algorithm the velocity of the robot is not known. The only information employed for inferring depth is the optical flow vectors of the landmark.

Regression analysis has been employed to estimate the distance between a landmark and the robot based only on the observed length of the optical flow vectors. The regression formula that expresses this distance is described by (13),

$$\ln D_i = a + b \cdot \ln len_i + \epsilon_i \quad (13)$$

where  $\ln D_i$  is the expected distance,  $a$  is the constant,  $b$  is the coefficient,  $\ln len_i$  is the length of the optical flow vectors, and  $\epsilon_i$  is the error. The following section presents the results from the two strategies.

## IV. RESULTS

In the least squares strategy, as it would be expected, the fit of the estimated distance to the actual distance increases as the number of snapshots increase and as the mean simulation error decreases. However, the larger the number of snapshots the greater the computational cost. Table I presents the  $R^2$  statistic

for various combinations of number of snapshots and mean simulation error. This  $R^2$  statistic for  $n$  snapshots and different error corresponds to the fit of the model  $D = 0 + 1 \cdot E(D)$  where  $D$  is the actual distance and  $E(D)$  is the expected distance estimated through the least squares method. The minimum number of snapshots is 4 as this is the number required for the system to become overdetermined. Assuming

TABLE I  
PERFORMANCE OF LEAST SQUARES METHOD -  $R^2$

Snapshots $n$	$R^2 (\epsilon_i \pm 1^\circ)$	$R^2 (\epsilon_i \pm 3^\circ)$	$R^2 (\epsilon_i \pm 5^\circ)$
4	0.971	0.137	0.000
6	0.991	0.722	0.000
8	0.996	0.883	0.336
10	0.998	0.940	0.649
12	0.999	0.967	0.801
14	0.999	0.979	0.881
16	1.000	0.987	0.915
18	1.000	0.991	0.944
20	1.000	0.993	0.959

that estimated distance is an unbiased estimate of the actual distance, if linear regression is used to estimate (for each combination of number of snapshots and mean simulation error)  $a$  and  $b$  in the function  $D = a + b \cdot E(D)$ , using the experiment's data, it should turn out that the best estimates are (approximately)  $a = 0$  and  $b = 1$ . However, this is not the case as there seems to be a tendency for this method to systematically underestimate distance, especially when the mean simulation error is high and the number of snapshots low. The level of underestimation is more obvious if in the previous function the constant is constrained to be 0. Interestingly, even though the bias decreases as the number of snapshots increases or the mean simulation error decreases, it is always significant as the standard error of the coefficient decreases too. In the following Table II the estimated coefficient of expected distance is presented for each combination of number of snapshots and mean simulation error. The confidence interval is 95%.

TABLE II  
PERFORMANCE OF LEAST SQUARES METHOD - COEFFICIENT

Snapshots $n$	Coeff. ( $\epsilon_i \pm 1^\circ$ )	Coeff. ( $\epsilon_i \pm 3^\circ$ )	Coeff. ( $\epsilon_i \pm 5^\circ$ )
4	1.017	1.166	1.726
6	1.013	1.130	1.386
8	1.009	1.084	1.239
10	1.006	1.057	1.162
12	1.004	1.042	1.118
14	1.004	1.033	1.089
16	1.003	1.026	1.073
18	1.002	1.021	1.059
20	1.002	1.018	1.049

In the optical flow strategy, after experimenting with various functional forms, we concluded that modelling the natural logarithm of distance as a linear function of the observed vectors' length is the optimal method to estimate distance. Thus, linear regression is utilised to estimate the combination of  $a$  and  $b$  that minimises the squared error in (13). Equation (13) has also been employed to compute  $R^2$  at different

velocity intervals in order to adapt to realistic implementations. The outputs of these regression models are presented in Table III.

TABLE III  
PERFORMANCE OF OPTICAL FLOW METHOD

Velocity $e$ (km/h)	Constant $a$	Coefficient $b$	$R^2$
$2 < e < 6$	2.96	-0.075	0.6029
$2 < e < 5$	3.03	-0.09	0.6483
$2 < e < 4$	3.08	-0.11	0.7569
$2 < e < 3$	3.1	-0.143	0.8052

Obviously, velocity influences the length of the optic flow vectors and the smaller the range used the better the estimation of the expected distance. However, we avoided using velocity as a variable in the tests as this work is intended to be used in self-localisation systems where the position of the robot is provided only by means of a laser range finder. Hence, velocity cannot be estimated with a laser finder.

The results show that optical flow can perform better in cases where the mean error is high and a small number of snapshots have been taken in the least squares strategy. Nevertheless, when velocity range is large enough, optical flow does not perform satisfactorily enough when compared to least squares.

## V. CONCLUSION

In this work we have addressed the problem of depth estimation using two different strategies, namely optical flow and least squares. The idea is to use a depth estimation method in a self-localisation system where the only sensor used for providing odometry data is a laser range finder. Such type of systems cannot operate in environments where there are no 'distinct' features like corners. Thus, the addition of a visual sensor can support the localisation process of the robot.

The results in this paper reveal that least squares is conditional on the number of snapshots and the mean simulation error. Optical flow, on the other hand, is conditional on the velocity of the robot. The results presented in this paper show that there is a trade-off between accuracy and computational cost.

## ACKNOWLEDGMENT

The authors would like to thank Dr Renato Cavalcante for his helpful comments on this work.

## REFERENCES

- [1] S. C. Diamantas and R. M. Crowder, "Localisation and mapping using a laser range finder: A goal-seeking approach," in *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems*, Valencia, Spain, 2009, pp. 270–276.
- [2] F. Amigoni, S. Gasparini, and M. Gini, "Map building without odometry information," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 2004, pp. 3753–3758.
- [3] T. Einsele, "Real-time self-localization in unknown indoor environments using a panorama laser range finder," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1997, pp. 697–702.

- [4] T. Camus, D. Coombs, M. Herman, and T.-S. Hong, "Real-time single-workstation obstacle avoidance using only wide-field flow divergence," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, 1996.
- [5] W. Warren and B. R. Fajen, "From optic flow to laws of control," in *Optic Flow and Beyond*, L. M. Vaina, S. A. Beardsley, and S. K. Rushton, Eds. Kluwer Academic Publishers, 2004, pp. 307–337.
- [6] P. C. Merrell, D.-J. Lee, and R. Beard, "Obstacle avoidance for unmanned air vehicles using optical flow probability distributions," *Sensing and Perception*, vol. 5609, pp. 13–22, 2004.
- [7] S. Hrabar, G. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 2005, pp. 302–309.
- [8] N. Slesareva, A. Bruhn, and J. Weickert, "Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps," in *27th DAGM Symposium*, vol. 3663, 2005, pp. 33–40.
- [9] I. Satoru, Y. Masaaki, H. Hiroyuki, and S. Mitsuru, "Depth estimation by optical flow method from images including moving objects," *Journal of the Institute of Image Electronics Engineers of Japan*, vol. 28, no. 1, pp. 48–55, 1999.
- [10] A. S. Malik and T.-S. Choi, "Depth estimation by finding best focused points using line fitting," in *Proceedings of the 3rd International Conference on Image and Signal Processing*, vol. 5099, 2009, pp. 120–127.
- [11] D. Boley, E. Steinmetz, and K. T. Sutherland, "Recursive total least squares: An alternative to using the discrete kalman filter in robot navigation," in *Lecture Notes in Artificial Intelligence*, vol. 25, 1995, pp. 221–234.
- [12] A. Bab-Hadiashar and D. Suter, "Robust total least squares based optic flow computation," in *Proceedings of the 3rd Asian Conference on Computer Vision*, vol. 1351, 1997, pp. 566–573.
- [13] C. jen Tsai, N. P. Galatsanos, and A. K. Katsaggelos, "Total least squares estimation of stereo optical flow," in *Proceedings of the IEEE International Conference on Image Processing*, 1998, pp. 622–626.
- [14] S. J. Maybank, "Algorithm for analysing optical flow based on the least-squares method," *Image Vision Computing*, vol. 4, no. 1, pp. 38–42, 1986.
- [15] C. jen Tsai, N. P. Galatsanos, and A. K. Katsaggelos, "Optical flow estimation from noisy data using differential techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999, pp. 3393–3396.
- [16] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, August 24–28, 1981, pp. 674–679.
- [17] OpenCV, "<http://opencv.willowgarage.com/wiki/>," 2008.
- [18] G. Bradski and A. Kaehler, *Learning OpenCV, computer vision with the opencv library*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2008.
- [19] J. Klein, "Breve: a 3D simulation environment for the simulation of decentralized systems and artificial life," in *Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems*, 2002.
- [20] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The player/stage project: tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, June 30 - July 3 2003, pp. 317–323.
- [21] MATLAB, "<http://www.mathworks.com/>," 2007.