Depth Computation Using Optical Flow and Least Squares

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 computation in robotics is an important step towards providing robust and accurate navigation capabilities to a mobile robot. In this paper we examine the problem of depth estimation with the view to be used in parsimonious systems where fast and accurate measurements are critical. For this purpose we have combined two methods, namely optical flow and least squares in order to infer depth estimates between a robot and a landmark. In the optical flow method the variation of the optical flow vector at varying distances and velocities is observed. In the least squares method snapshots of a landmark are taken from different robot positions. The results of the two methods show that there is a significant increase in depth estimation accuracy by combining optical flow and least squares.**

Keywords: *depth computation, optical flow, least squares, robot navigation*

I. Introduction

Depth computation has long been an essential task in mobile robots that use camera images for navigating in an environment. In this research, we compute the distance between a robot and a landmark with the view to support the localization and mapping problem, a problem which is at the heart of today's mobile robotic systems. Localization and mapping as well as depth computation are vital components of a navigating robot. Such type of robots have a plethora of application ranging from search and rescue to planet exploration. This type of work has been developed in order to be integrated into a self-localization system as in [1] or similar type of systems as 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 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 methods for calculating the distance between the robot and a landmark and combine the results from both approaches to infer a more robust depth estimate. 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 be it natural or artificial. One of the first works that studied the relation of scene geometry and the motion of the observer was by [4]. Optic flow has extensively been used thus far for obstacle avoidance and centering a robot in corridor-like environments [5], [6], [7], [8]. Optic flow for depth estimation has been used in [9], [10] and a least squares approach for inferring depth appears in [11], [12]. Furthermore, a large amount of effort has been focused on using total least squares [13], [14], least squares [15], or constrained total least squares methods [16] for calculating optical flow field. A comparative approach between optical flow and least squares appears in [17].

This paper consists of five sections. Section II discusses the background work on optical flow and least squares as well as the criteria needed for extracting optical flow. Section III presents the methodology followed for the two different methods, namely optical flow and least squares, for computing depth. Section IV discusses the results of the two methods and it presents a mixed approach where the outcomes of the two methods are combined. Finally, Section V provides a discussion on the conclusions drawn from this research.

II. Background Work

This section describes the mathematics that underlie the optical flow algorithms, and in particular, the Lucas-Kanade (LK) algorithm [18] which has been employed in this research. In order for the optical flow algorithms to perform well, some suitable images need to be chosen. This suitability refers to

*Currently a postdoctoral research associate within Intelligent Robot Lab, Pusan National University, Republic of Korea.

978-1-4244-9315-9/10/$26.00 ©2010 IEEE

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 [19] 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. Neighboring 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 neighboring 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 neighboring pixels are considered and solve a system of linear equations [20]. Hence, if we take a window of 5×5 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 5×5 pixels is expressed by (1)

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

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

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

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

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

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

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

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),

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

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 computing depth using optical flow and least squares. The 3D simulation environment in [21] is used for the optical flow method, while the 2D simulator in [22] is employed for the least squares method. Data analysis has been implemented in the programming environment of [23]. 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 computing depth using the least squares method, a visual landmark in the environment is tracked and the position of the landmark is inferred based on the orientation and position of the robot. A similar approach has been used by [12] to infer the starting position of the robot.

In this simulation, various snapshots of a landmark have been taken at equally spaced time steps and, the angle θ between the robot position and the visual landmark is calculated. Each snapshot represents a linear equation and the solution of the linear system gives us the position of the landmark L. Thus, a set Ω_i is formed for every snapshot taken. Equations (6) and (7) show an example of two snapshots,

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

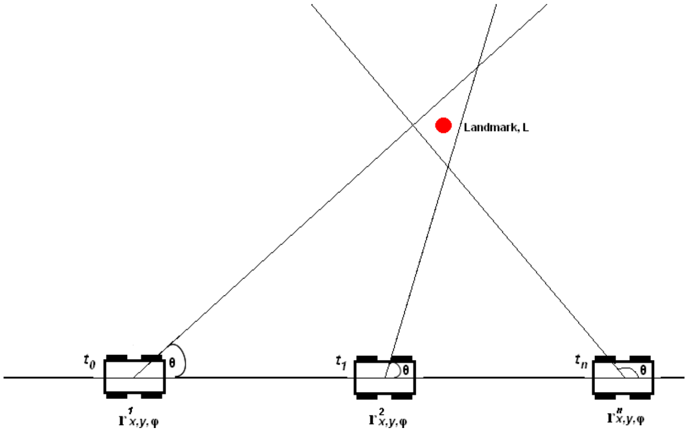


Fig. 1. Robot snapshots of the landmark L taken at different time steps and at different positions r.

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

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 that is perpendicular to the line formed between the robot and the landmark, that is angle θ. 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 ϵ_i.

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

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

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

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

In (10) C is a 2×2 matrix and $\gamma \in \Re^2$. The position of the landmark, L, 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 drawn from a Gaussian distribution (explained in detail in the next section) for different 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, ϵ_i, in θ is uniformly distributed and it has been tested with three different deviations ±1°, ±3°, and ±5°.

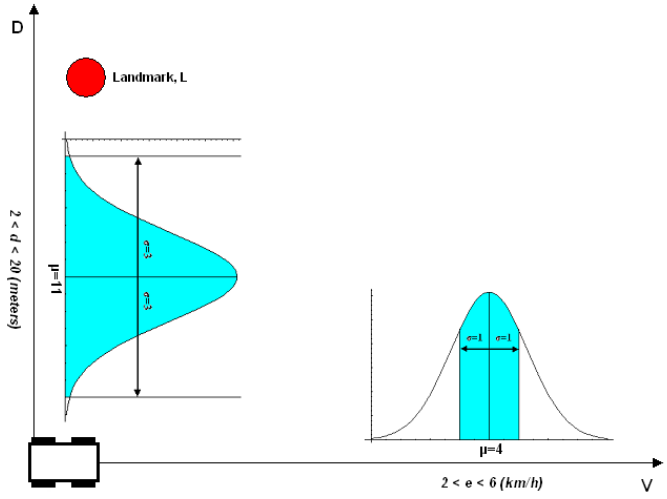


Fig. 2. Gaussian distributions used for modeling distance, D, and velocity, E, in the optical flow method. Their mean and standard deviation are $\mu = 11, \sigma = 3$ and $\mu = 4, \sigma = 1$, respectively. The range of their values is $2 < d < 20$ $(meters)$ for distance and $2 < e < 6$ (km/h) for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

B. Optical Flow

For the computation of depth using the optical flow method, 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 data set are similar to the ones used by the robot when it navigates during test runs. Thus, a joint probability distribution has been created with two continuous and independent variables, that is, distance and velocity, and is expressed by (12),

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

The velocity, E (in km/h), and the distance, D (in $meters$), 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 different distances and velocities. Figure 2 provides a pictorial representation of the Gaussian distributions used to model distance, D, between the robot and a landmark, L, and velocity, E, of the robot. 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 used 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),

$$E(D) = e^{a+b\cdot len} \tag{13}$$

where $E(D)$ is the expected distance, a is the constant, b is the coefficient, and *len* is the length of the optical flow vectors (in pixels). The following section presents the results from the two methods as well as those from the combined approach.

IV. Results

Following the least squares method for inferring depth, 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 variable error in θ corresponds to the fit of the model $D = 0 + 1 \cdot E(D)$ where D is the 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.

TABLE I
Performance of Least Squares method - R^2

Snapshots n	R^2 ($\epsilon \pm 1°$)	R^2 ($\epsilon \pm 3°$)	R^2 ($\epsilon \pm 5°$)
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

The graphs, Figs. 3(a)-3(d), depict the convergence of the least squares method as the number of snapshots increase. In this simulation, an object was placed at various distances from the robot [24]. The accuracy of the estimation drops as the distance between the robot and the object increases. Moreover, it can be seen how the increasing number of snapshots improve the estimated distance, that is small deviation, in all three mean errors. Deviation represents the difference between the actual and the estimated distance of the landmark. In each of the four graphs three different mean error levels have been used. In addition, the behavior of the method is shown when the system is still underdetermined, that is when only two snapshots have been taken. Furthermore, as it can be seen from the graphs, a sensible estimate can be inferred with a small number of steps, that is snapshots.

Assuming 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

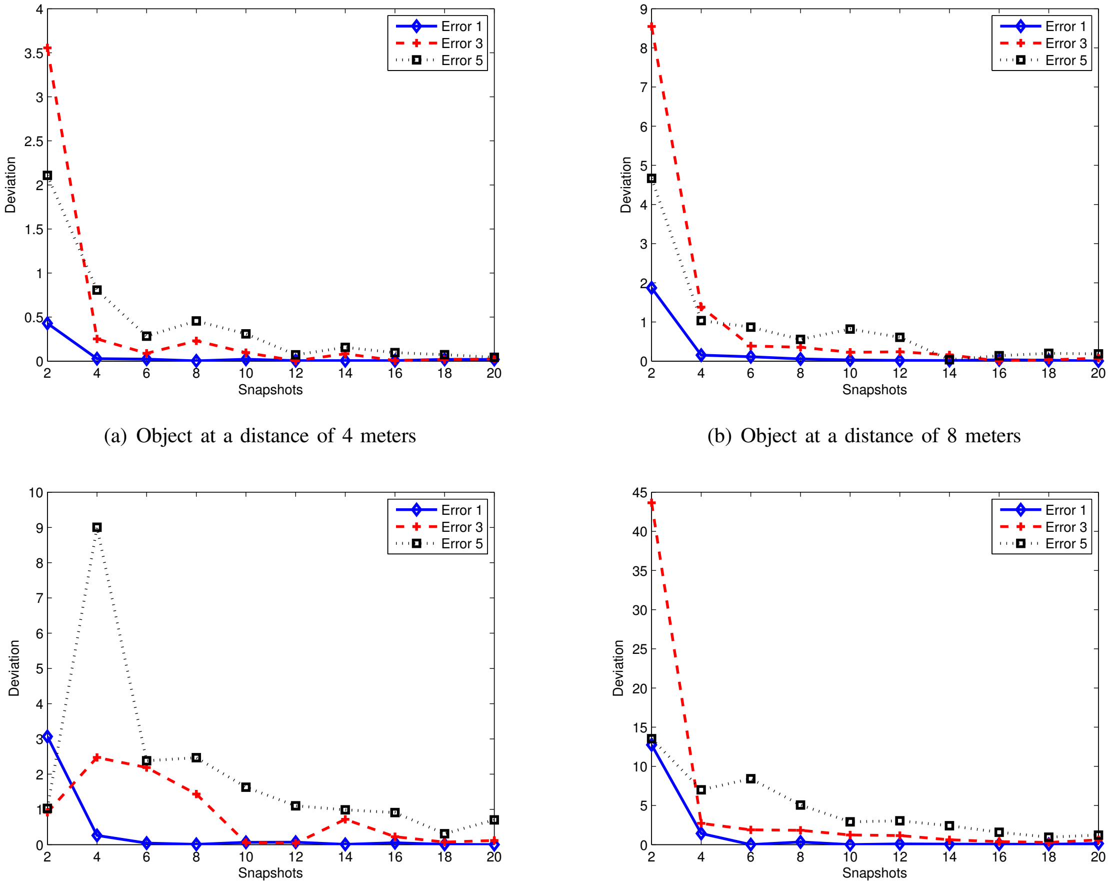
Snapshots n	Coeff. ($\epsilon \pm 1°$)	Coeff. ($\epsilon \pm 3°$)	Coeff. ($\epsilon \pm 5°$)
4	1.017	1.166	1.726
6	1.013	1.130	1.386
8	1.009	1.084	1.239
10	1.006	1.057	1.162
12	1.004	1.042	1.118
14	1.004	1.033	1.089
16	1.003	1.026	1.073
18	1.002	1.021	1.059
20	1.002	1.018	1.049

In the optical flow method, after experimenting with various functional forms, we concluded that modeling 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 utilized to estimate the combination of a and b that minimizes the squared error in (13). Equation (13) has been employed to compute R^2. The outcome of the linear regression is $R^2 = 0.6029$, constant $a = 2.96$, and coefficient $b = -0.075$. A better R^2 would have been achieved had a smaller velocity range been considered.

As these two methods, that is optical flow and least squares, offer two independent estimations of the distance between the robot and the landmark, combining them should offer a more accurate estimate. Thus, ordinary least squares (OLS) regression has been employed to estimate a function that gives the most accurate measurement of the actual distance. Obviously, as the number of snapshots increases, the estimation becomes very accurate, but cost inefficient, too. Therefore, measuring distance based on the length of the observed landmark offers little additional information, an estimation derived after a large number of snapshots was taken. However, it can significantly improve estimations that arise from a lower number of snapshots and therefore, provide higher accuracy, without sacrificing cost efficiency.

By way of example, we model the natural logarithm of distance as a function of the estimated distance derived by four snapshots, $(n = 4)$, mean error equal to three, $(\epsilon \pm 3°)$, and the length of the observed vector. After experimenting with various functional forms of the two variables, we concluded on the equations described in Table III. The dependent variable is *ln distance* and $R^2 = 0.8366$.

Obviously, increasing the number of snapshots decreases the additional value of the optical flow method. Thus, if



(a) Object at a distance of 4 meters

(b) Object at a distance of 8 meters

(c) Object at a distance of 12 meters

(d) Object at a distance of 16 meters

Fig. 3. Deviation ($meters$) between actual and computed depth using least squares approach under different object distances.

TABLE III
PERFORMANCE OF LEAST SQUARES AND OPTICAL FLOW METHODS

Variable	Coefficient	T-Stat	P-Value
$ln(e3_sn4)^2$	0.642	23.020	0.000
$ln(e3_sn4)^3$	-0.141	-18.040	0.000
$ln(len)$	0.821	2.300	0.022
$(ln(len))^2$	-0.533	-3.010	0.003
$(ln(len))^3$	0.086	3.010	0.003
constant	0.587	2.410	0.016

a high number of snapshots is taken there is no need to consider information derived by optical flow. As a consequence, whether to estimate the distance based on optical flow information, through observed snapshots, or via a mixed approach, is a trade-off between desired accuracy and cost efficiency. Similarly, as the number of snapshots increase, the optical flow related variables' significance statistics will be reducing as the additional value of this information becomes insignificant, if a high number of snapshots is taken.

V. CONCLUSION

This paper has presented the results of computing the distance between a robot and a landmark by combining two different methods, namely optical flow and least squares. The idea is to use a depth estimation method in a self-localization 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 localization process of the robot. In this paper, both methods employ a camera sensor in order to make estimates of depth.

The results derived from this research show that combining optical flow and least squares can improve significantly the estimation of distance between the robot and a landmark. In addition, as it was shown in the tables, least squares is conditional on the number of snapshots and the mean simulation error entailing a trade-off between accuracy and computational cost. Finally, the smaller the distance between a robot and an object, the greater is the accuracy of the method.

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] J. J. Gibson, *The Perception of the Visual World*. Santa Barbara, CA, USA: Greenwood Publishing Group, 1974.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] S. C. Diamantas, A. Oikonomidis, and R. M. Crowder, "Depth estimation for autonomous robot navigation: A comparative approach," in *International Conference on Imaging Systems and Techniques*, Thessaloniki, Greece, 2010, pp. 426–430.

[18] 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.

[19] OpenCV, "http://opencv.willowgarage.com/wiki/," 2008.

[20] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. Sebastopol, CA, USA: O'Reilly Media, Inc., 2008.

[21] 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.

[22] 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.

[23] MATLAB, "http://www.mathworks.com/," 2007.

[24] S. C. Diamantas, "Biological and metric maps applied to robot homing," Ph.D. dissertation, School of Electronics and Computer Science, University of Southampton, 2010.