15th Triennial World Congress, Barcelona, Spain

WIRELESS VISUAL SERVOING FOR ODIS – AN UNDER CAR INSPECTION MOBILE ROBOT [1]

Lili Ma *, **Matthew Berkemeier** *, **Yangquan Chen** *, **Morgan Davidson** *, **Vikas Bahl** * **and Kevin L. Moore** *

* *Center for Self-Organizing and Intelligent Systems (CSOIS)*
Dept. of Electrical and Computer Engineering
UMC 4160, College of Engineering, 4160 Old Main Hill
Utah State University, Logan, UT 84322-4160, USA

Abstract: This paper presents a simple robot localization technique using the wireless visual servoing technique involved in an autonomous ground vehicle ODIS (omnidirectional inspection system) for under-car inspection tasks in standard parking lot environment. Based on the current architecture of ODIS and the dedicated scripting language, an iterative visual servoing scheme is proposed to align the yellow line of the parking lot. The iterative scheme tolerates the uncertain time delay due to the wireless connections without introducing stability problem due to time-varying delay in real-time visual servoing. Experimental results are presented to show that for our specific application, the wireless visual servoing technique presented in this paper is an efficient way for robot localization.

Keywords: Omni directional vehicle, Under-car inspection, Mobile robot, Autonomous ground vehicle, Visual servoing, localization, Iterative, Wireless.

1. BACKGROUND

The Utah State University Omni-Directional Inspection System) (USU ODIS) is a small, manportable mobile robotic system that can be used for autonomous or semi-autonomous inspection under vehicles in a parking area (Moore *et al.*, 2001; Flann *et al.*, 2001*a*,*b*). Customers for such a system include military police (MP) and other law enforcement and security entities (Nguyen and Bott, 2000). A unique feature in ODIS is the notion of the "smart wheel" (Poulson *et al.*, 1998) developed by the Center for Self-Organizing and Intelligent Systems (CSOIS) at USU which has resulted in the so-called T-series of omni-directional (ODV) robots (Moore and Flann, 2000).

Fig. 1 shows the mechanical layout of the ODIS robot. The robot is 9.8 cm tall and weighs approximately 20 kgs. The mechanical subsystem of the ODIS vehicle is based on three omni-directional wheel assemblies mounted within a stressed-skin chassis, which also contains the vetronics subsystem, the battery and power distribution subsystem, sensors, and a camera gimble subsystem. The ODIS vetronics architecture includes seven different processing nodes: the Master Node, three Sensor Nodes, the Camera Node, and three Wheel Nodes. A user interface is maintained through two independent RF communications links, one connecting the master node to an external joystick and a second that connects the master node to the Planner Node, Planner GUI (graphical user interface), and Vehicle GUI, all of which reside off-vehicle. There are two types of sensing functions

[1] This work is supported in part by U.S. Army Automotive and Armaments Command (TACOM) Intelligent Mobility Program (Agreement no. DAAE07-95-3-0023). Corresponding author: Dr YangQuan Chen. E-mail: yqchen@ieee.org; Tel. 01-435-7970148; Fax: 01-435-7972003. URL: http://www.crosswinds.net/~yqchen.

on ODIS. For navigation ODIS uses a combination of GPS (global positioning system), a fiber optic gyro (FOG), and odometry based on wheel encoder measurements. For environmental sensing ODIS uses three types of sensors. Ten ultrasonic sensors are used together with thirty-two infrared (IR) sensors and one laser distance measurement sensor. The system also has a camera node to control the camera pan/tilt mechanism that is used to point the camera relative to the vehicle. However, the camera is primarily designed for the under car inspection video transmission to the base-station computer not intending for decision-making or control. For more detailed description, see (Moore *et al.*, 2001; Flann *et al.*, 2001*a*,*b*).

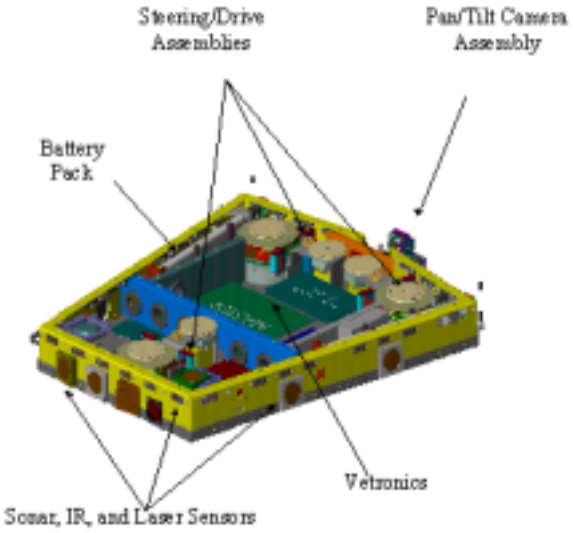

Fig. 1. The mechanical and vetronics layout of ODIS

ODIS is designed to perform the under car inspection by sending wirelessly the captured video frames to GUI of the base station. The working period can be hours long for many parking lots. Therefore, it is important for ODIS to be able to localize and estimate its pose with respect to an internal world model (map). Although some techniques can be used to improve the FOG accuracy (Zhu *et al.*, 2000), the accumulative errors due to FOG and ODIS odometry can be progressively dominant and may significantly deflect the pose of ODIS (position and orientation in world coordinate system). Therefore, from time to time, localization is essential for a reliable task execution of ODIS. Mobile robot localization is a sustaining research topic with no unique solution (Mouaddib and Marhic, 2000; Shimshoni, 2001) since the localization depends on the structure of the environment and the sensing capacities of the specific mobile robot. In general, the localization method is environment and robot specific.

In our case, the environment is the car parking area. We can use the laser or sonar to do the localization based on landmarks such as lamp posts, curbs etc. However, the most commonly seen in the parking area are the yellow lines painted on the ground. In this paper, we use visual servoing to a yellow line in a parking lot to perform the localization task. As explained in the above, the challenge here is that the camera is designed for the under car inspection video transmission and within ODIS there is *no* video capture card. We consider to capture the video frames, wirelessly sent from the ODIS camera transmitter, on the base station computer (ODIS laptop computer) and perform the visual servoing to yellow line to achieve ODIS localization by wirelessly sending the scripts to ODIS from the base station computer. Therefore, the research efforts reported in this paper can be regarded as a *value-added block* for ODIS functionality. Based on the current architecture of ODIS and the dedicated scripting language (Flann *et al.*, 2001*b*), an iterative visual servoing scheme is proposed to align to the yellow line painted on the ground of the parking lot. The iterative scheme tolerates the uncertain time delay due to the wireless connections without introducing stability problem due to time-varying delay in real-time visual servoing. Experimental results are presented to show that for our specific application, the wireless visual servoing technique presented in this paper is an efficient way for robot localization.

The rest of this paper is organized as follows. Sec. 2 presents the basic idea and implementation architecture for ODIS wireless visual servoing in some detail. Sec. 3 describes the iterative visual servoing controller scheme. Experimental results are presented in Sec. 4. Some concluding remarks together with some of our planned further efforts are given in Sec. 5.

2. ODIS LOCALIZATION VIA WIRELESS VISUAL SERVOING

This section describes the basic idea and the architecture for ODIS wireless visual servoing together with some implementation details including the low level yellow line extraction and fitting.

2.1 *Basic Idea*

The basic idea for ODIS localization is simply the visual servoing to the yellow line painted on the ground of the parking lot, as illustrated in Fig. 2. We can regard the yellow line of the parking lot as a landmark for localization. It is actually implied that from the map the position and orientation of each yellow line is known. The command for ODIS to perform yellow line alignment is from the high-level block, e.g., from either planner or just GUI click. The major objective is to reset the FOG after inspection of several parking lots so that the

drifting can be kept small. Obviously, this can be done via a dedicated utility script sent wirelessly from the base station to ODIS. Moreover, unlike the image-based path tracking (Jiang *et al.*, 1996; Shen *et al.*, 2001; Mezouar and Chaumette, 2001), the real-time requirement is not critical here.

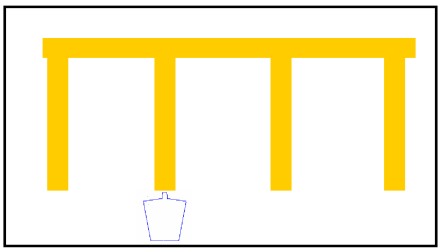

Fig. 2. Basic idea of ODIS localization using yellow line alignment via visual servoing

2.2 *Hardware Architecture*

Figure 3 shows the hardware architecture for the wireless visual servoing for ODIS localization. ODIS laptop is a Pentium-3 notebook computer with Windows 2000 installed and a PCMCIA image capture card. The GANZ CM3000 camera unit is a compact module including a small color camera and a video transmitter installed on ODIS. The video signal is sent over a 900 MHz transmitter (CVR-1000) to a receiver (CVT-M) located at the base station. Both transmitter and receiver are made by Coherent Communications operating in several bands within the 902 - 928 MHz, which allows us to operate without interfering with other onboard modems. By connecting the receiver to the video capture card inside ODIS laptop, live images can be grabbed continuously via Microsoft Vision SDK (software development kit). On the other hand, through wireless LAN, scripts or commands can be sent to ODIS. Meanwhile, ODIS status and position/orientation can be queried from the base station computer. Therefore, a closed loop is formed via wireless connections.

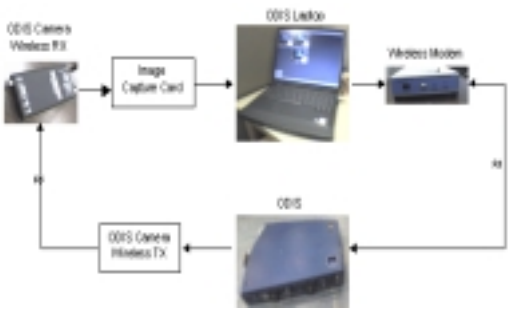

Fig. 3. Hardware architecture of the wireless visual servoing system

The signal flow in the above closed-loop system is shown in Fig. 4. ODIS continuously sends live image frames to ODIS laptop (Host). For each image frame, if there is a yellow line, the Yellow Line Detection Function calculates its angle and the starting point in image frame which serve as the output measurement of the visual servoing system. Control Strategy block compares the measured output with the desired one and constructs the corresponding translation and/or rotation commands which are to be sent wirelessly down to ODIS. ODIS low level control subsystems execute these commands (actuation).

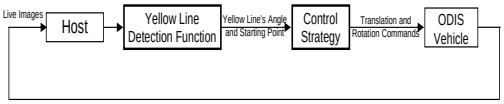

Fig. 4. Signal flow of the wireless visual servoing system

2.3 *Software Architecture*

The Host is based on the WINDOWS 2000 platform. Therefore, we apply Microsoft Vision SDK to acquire the live images. Microsoft Vision SDK includes an MFC AppWizard which is easily used to create MFC programs with selected functionalities of the Vision SDK. The resulting programs will attach a sequence to an image source and support a second thread capturing images in the background. One major feature of Microsoft Vision SDK is that it provides a basic image object that supports a diversity of pixel formats such as, for color image, RGBA pixel types (Red, Green, Blue, and alpha).

Using socket programming, the visual servoing module can be executed independently which is capable of talking to ODIS bidirectionally. This independent module can be easily integrated into the base station GUI or planner as an additional function or utility script. Basically, visual servoing module is triggered by base station GUI or planner and returns an event completion flag and some possible exceptional flags. Central in the module are the Yellow Line Detection Function and the visual servo algorithm which will be described in the following in some details.

2.4 *Yellow Line Detection Function*

The five steps in Yellow Line Detection Function are (1) Yellow Color Separation; (2) Connected Component Labeling; (3) Maximal Region Determination - Line Region Mask; (4) Points for Line Determination and (5) Line Fitting. In Step (1), we use RGB component to create a difference

image by the formula $Red + Green - 2Blue$. For each row in the difference image, a threshold value is calculated using an average of the minimum and maximum difference values in the row. Finally each row is thresholded and the result is stored in what we call an *Enhanced Mask*. Step (2) is applied to find all equivalence classes of connected pixels in a binary image using the 8-connectivity principle. Taking the *Enhanced Mask* as input, the output via the Connected Component Labeling is another image called *Region Map* in which every pixel in one connected region is labeled e.g. "1" and every pixel in another connected region is labeled "2", etc. In Step (3), with a *Region Map* obtained in Step (2), simply choosing the region with the maximal number of pixels as what we call the *Line Region Mask*. Step (4) determines the points to form a line for line fitting in Step (5). This is done by scanning the *Line Region Mask* from both left and right till in the middle. The middle points are used to estimate the yellow line parameters such as the angle and the starting point position. Finally, with the data set obtained in Step (4), an isotropic line fitting algorithm to determine the yellow line parameters: orientation angle and the starting point position. Figure 5 shows the images at each step of the Yellow Line Detection Function described above.

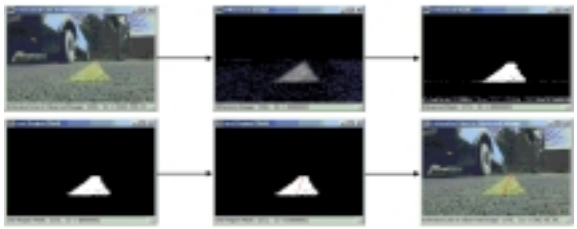

Fig. 5. Image processing results in various steps of the Yellow Line Detection Function

3. AN ITERATIVE VISUAL SERVOING CONTROL ALGORITHM

3.1 *Simple Proportional Controller*

Due to the hardware architecture (Sec. 2.2), currently, the wireless visual servoing control scheme is implemented in a start-stop manner or as an iterative controller as described by Fig. 6.

Figure 7 shows the image frame and camera frame. The iterative visual servoing controller is a simple proportional one given by

$$\Delta x = -C_1(u_d - u) = -C_1\delta u \quad (1)$$

$$\Delta z = C_2(v_d - v) = C_2\delta v \quad (2)$$

$$\varphi = -C_3(\theta_d - \theta) = -C_3\delta\theta \quad (3)$$

where Δx and Δz are incremental translation distances in x and z direction in the camera frame respectively; φ is the yaw angle increment with respect to the current ODIS orientation; the yellow line angle θ is positive if clockwise with respect to v axis in the image frame; u_d and v_d are the desired position in the image plane which correspond to the yellow line aligned vertically in the middle of the image plane (i.e., $\theta_d = 0$); C_1, C_2, C_3 are all positive gains which were roughly determined during our tests.

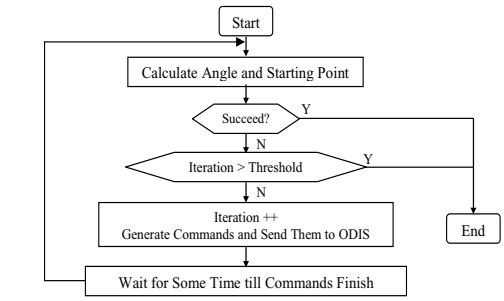

Fig. 6. Iterative visual servoing control scheme

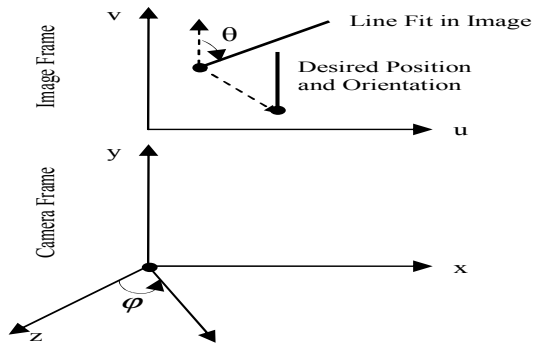

Fig. 7. Image frame and camera frame

3.2 *Coordination Transformation*

It should be clear at this point that the movements due to the above controller are with respect to the camera frame. In actual operations, the corresponding movements with respect to ODIS body fixed frame or inertial frame are desirable. It is clear from Fig. 8 that a pure translation with respect to the camera frame corresponds to a pure translation in the inertial frame. Therefore, from Fig. 8, we can easily get

$$\Delta x_B = \Delta z_C \quad (4)$$

$$\Delta y_B = -\Delta x_C \quad (5)$$

where Δx_B and Δy_B are incremental translation distances in x and z direction in the body fixed frame of ODIS respectively; Δx_C and Δz_C are the same as x and z defined in (1) and (2).

However, the rotation case is quite different with the translation case explained in the above. For a pure rotation with respect to the camera frame, it

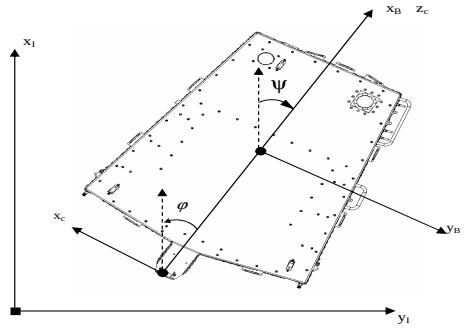

Fig. 8. ODIS coordinate systems: initial, body fixed and camera frames

corresponds to a translation in inertial frame plus a rotation in the body fixed frame as illustrated in Fig. 9. We can see from Fig. 9 that the rotation angle φ in the camera frame corresponds to rotating ODIS with an angle -φ in the body fixed frame and to translating to a new position (Point C) in the inertial plane such that the camera position (Point A) is kept unchanged. So, a combination of rotation and translation is required for this case.

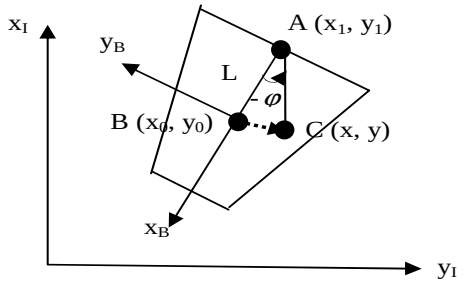

Fig. 9. ODIS movements corresponding to a pure rotation in the camera frame

Since at any time ODIS knows its position (x_0, y_0) and orientation ψ and so does the base station computer, the position of Point C in Fig. 9 can be calculated as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(-\varphi) & -\sin(-\varphi) \\ \sin(-\varphi) & \cos(-\varphi) \end{bmatrix} \begin{bmatrix} L\sin(\frac{\pi}{2} - \psi) \\ L\cos(\frac{\pi}{2} - \psi) \end{bmatrix} + \begin{bmatrix} x_0 - L\sin(\frac{\pi}{2} - \psi) \\ y_0 - L\cos(\frac{\pi}{2} - \psi) \end{bmatrix} \quad (6)$$

where L is the distance from the origin of the body fixed frame of ODIS to the origin of the camera frame.

3.3 *Some Implementation Issues*

During our implementation of the visual servoing control algorithm described above, there are numerous technical details need to be addressed. The following listed are some of the important issues.

- *Yellow line searching.* Upon receipt of the localization command of visual servoing, it may happen that the yellow line does not lie in the field of view (FOV) of ODIS camera. A routine is designed to automatically locate the yellow line within the FOV of camera such that the iterative visual servoing routine can start to work. If there is no yellow line found after 360° trial, report an exception to the planner or base station GUI.
- *Upper half-image plane clipping.* It may happen that a yellow colored car appears in the captured image. This will greatly affect the yellow line detection. Fortunately, since the car image is usually in the upper half of the image plane, simply clipping the upper half and using only the lower half plane for yellow line detection will solve the "yellow car disturbance problem".
- *Image frame skipping.* Due to the wireless connections, not all video frames are with good quality. Occasionaly, a frame can be very bad and not usable due to e.g. electromagnetic interference. We set a lower threshhold and check the number of yellow pixels in the lower half image plane. Discard those images containing few yellow pixels.
- *Median filtering.* A window buffer is built to store the yellow line parameters and a median filter is used to make sure that the "measurement" is not jumping due to the wireless connections.

4. EXPERIMENTAL RESULTS

In our present effort, we did not perform any camera calibration and the visual servoing controller gains $C_j (j = 1, 2, 3)$ are determined by experimental trials. Cautious or smaller gains can ensure the convergence of the iterative visual servoing process which suffers from the slow convergence rate. On the other hand, too aggressive or larger gains may oscillate the whole system although they may make the convergence faster. A suitable design of the controller gains largely depends on the modeling efforts. At present, our objective is just to make the system work. Systematic and optimal design of the control gains or the development of the other control laws for the visual servoing system are our possible future efforts.

Due to the space limitation, we report here a typical wireless visual servoing experimental result. The results are summarized in Fig. 10 where we can see that the visual servoing errors in the image frame converge to near zeros as the number of iterations increases. After the alignment to the yellow line satisfactorily with a number of visual servoing control iterations, the ODIS can be thought of being localized and then a reset FOG command can be sent to ODIS.

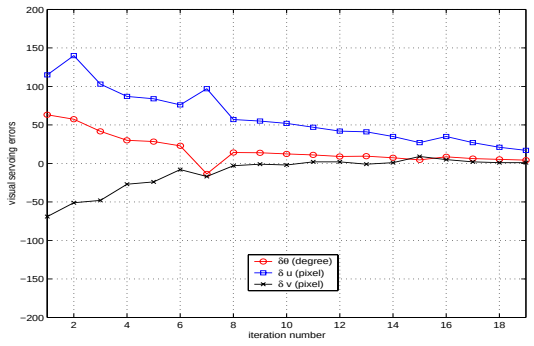

Fig. 10. Visual servoing errors vs. iteration in the image frame

5. CONCLUDING REMARKS

We have presented in this paper a simple robot localization technique using the wireless visual servoing technique involved in an autonomous ground vehicle ODIS for under-car inspection tasks in standard parking lot environment. Based on the current architecture of ODIS and the dedicated scripting language, an iterative visual servoing scheme is proposed to align to the yellow line of the parking lot. The iterative scheme tolerates the uncertain time delay due to the wireless connections without introducing stability problem due to time-varying delay in real-time visual servoing. Experimental results are presented to show that for our specific application, the wireless visual servoing technique presented in this paper is an efficient way for robot localization.

A near future effort is to use the "continuous" visual servoing instead of the "iterative scheme" presented in this paper. We need to change the interface between scripting command and the sensor motion scheduler (SMS) to receive in real time the time-varying reference signal for tracking purpose. Presently, SMS only accepts set-point commands for motion control. With this "reference tracking mode" in SMS, the "continuous" visual servoing can be done easily. However, in this case, the control period and delays will be important and stability issues will be dominant. The convergence analysis and simulation studies for this continuous case have been done and will be reported separately.

ACKNOWLEDGMENTS

The authors are grateful to S. Rich for his installation and integration of the camera module on ODIS, M. Frandsen for his help in the wireless visual servoing tests, R. V. Mierlo for his early investigation of the yellow line detection algorithm.

References

Flann, N. S., K. L. Moore and L. Ma (2001*a*). A small mobile robot for security and inspection operations. In 'Preprints of the 1st IFAC Conf. on Telematics Applic. in Autom. and Tobotics (TA2001)'. Weingarten, Germany. pp. 79–84.

Flann, N. S., M. Davidson, J. Martin and K. L. Moore (2001*b*). Intelligent behavior generation strategy for autonomous vehicles using a grammar-based approach. In 'Proc. of 3rd Int. Conf. on Field and Service Robotics'. Helsinki Univ. of Tech., Otaniemi, Espoo, Finland.

Jiang, P., H. Chen and Y. Wang (1996). Robot visual servoing for curve tracking. In 'Proc. of the 13th IFAC World Congress'. San Francisco, USA. pp. 337–342.

Mezouar, Y. and F. Chaumette (2001). Design and tracking of desirable trajectories in the image space by integrating mechanical and visibility constraints. In 'Proc. of the 2001 IEEE Conf. on Robot. and Autom.'. Seoul, Korea. pp. 731–736.

Moore, K. L. and N. S. Flann (2000). 'A six-wheeled omnidirectional autonomous mobile robot'. *IEEE Control Systems* **20**(6), 53–66.

Moore, K., N. Flann, S. Rich, M. Frandsen, Y. Chung, J. Martin, M. Davidson, R. Maxfield and C. Wood (2001). Implementation of an omni-directionsl robotic inspection system (ODIS). In 'Proc. of SPIE Conf. on Robotic and Semi-Robotic Ground Vehicle Tech.'. Orlando, FL, USA.

Mouaddib, E. M. and B. Marhic (2000). 'Geometrical matching for mobile robot localization'. *IEEE Trans. on Robot. and Autom.* **16**(5), 542–552.

Nguyen, H. G. and J. P. Bott (2000). Robotics for law enforcement: beyond explosive ordnance disposal. Tech. Rep. 1839. SPAWAR Systems Center, San Diego, CA 92152-5001.

Poulson, E., J. Jacob, R. Gunderson and B. Abbot (1998). Design of a robotic vehicle with self-contained intelligent wheels. In 'Proc. of SPIE Conf. on Robotic and Semi-Robotic Ground Vehicle Tech., vol. 3366'. Orlando, FL, USA. pp. 68–73.

Shen, Y., Y.-H. Liu, K. Li, J. Zhang and A. Knoll (2001). Aymptotic motion control of robot manipulators using uncalibrated visual feedback. In 'Proc. of the 2001 IEEE Conf. on Robot. and Autom.'. Seoul, Korea. pp. 241–246.

Shimshoni, I. (2001). On mobile robot localization from landmark bearings. In 'Proc. of the 2001 IEEE Conf. on Robot. and Autom.'. Seoul, Korea. pp. 3605–3610.

Zhu, R., Y. Zhang and Q. Bao (2000). 'A novel intelligent strategy for improving measurement precision of FOG'. *IEEE Trans. on Instru. and Measur.* **49**(6), 1183–1188.