A RECONFIGURABLE AGV WITH OMNI-DIRECTIONAL SENSING

B. KOTZE, G.D. JORDAAN AND H. VERMAAK

ABSTRACT

Automatic guided vehicles are being used increasingly in a manufacturing environment. Developing a platform that could be easily reconfigured is perhaps a desirable option for a user with low capacity outputs. The research described in this article concentrates on such a vehicle and the development of the actuators and sensors for navigation and proper functionality.

Keywords: AGV, omni-directional sensing, polar plot, NI6009 and MATLAB®.

1. INTRODUCTION

The operation of automatic guided vehicles (AGV) involves several aspects, including its power source, environmental detection and its drive system to name a few. This forms part of a motorised platform to be utilised in a certain environment being used for a specific application.

Object observation and/or recognition in a changing environment is a crucial part for navigation purposes. Infrared sensing, ultrasonic and whisker sensors are but a few sensing techniques used for detecting objects in the path of an AGV, as well as to determine the distance to the object (Fend, Bovet & Hafner, 2007: 2). Single cameras as well as 2-D and 3-D images are also typically used to obtain the distance to and information about an object in the way of a functioning AGV (Fernandes & Neves, 2006: 157-160). Cameras, with associated image processing techniques, can improve the quality of information provided to the AGV due to the unique versatility of vision. However, it presents particular challenges, as it requires acquisitioning techniques dependent on a changing environment and a tremendous amount of image processing. Depending on the application, thermal images and the like can also be utilised in meeting specialised sensing requirements (Treptow, Cielniak & Duckett, 2007).

Figure 1 depicts a block diagrammatical representation of such a platform – incorporating a single camera facing a round dome mirror. The image is mathematically transformed into a panoramic view and if necessary transmitted through a wireless medium such as a wireless local area network (WLAN) – to a remote computer that analyse the information received and sends appropriate control instructions back to the AGV. The system makes provision for a human machine interface (HMI) that communicates instructions to and from the AGV. This information comprises of an omni-directional image that can be viewed as a 360°, panoramic image (Swanepoel, Kotze & Vermaak, 2008: 41-44).

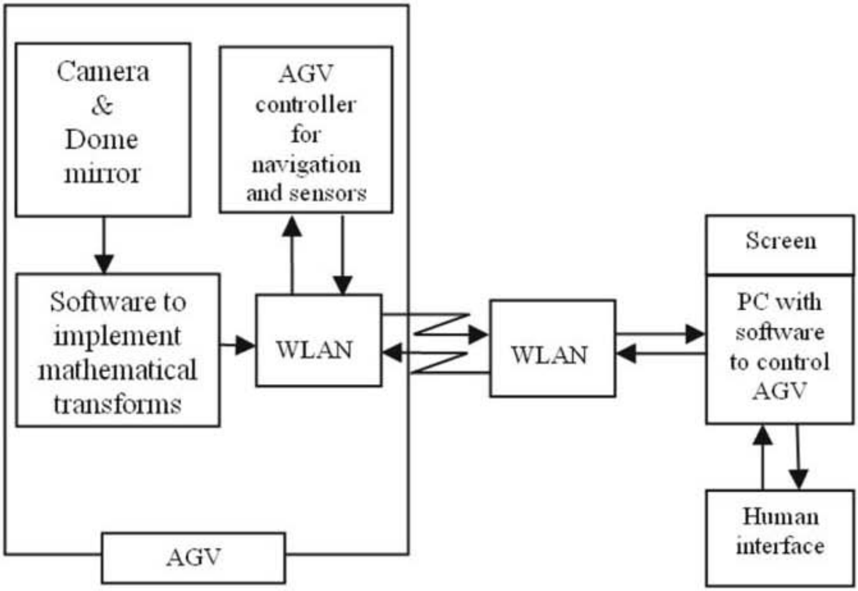

Figure 1 : Block diagram of AGV platform

Some of the building blocks of the AGV platform are discussed in the following paragraphs.

2. OMNI-DIRECTIONAL SENSING

The omni-directional sensor consists of a dome mirror and camera connected to a Central Processing Unit (CPU) via Universal Synchronous Bus (USB) or fire wire. The configuration shown in Figure 2 is mounted on top of the AGV as the omni-directional vision sensor (Swanepoel, 2009: 58).

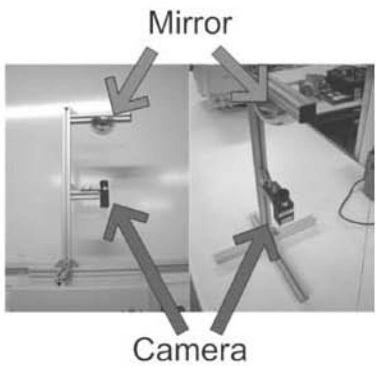

Figure 2: Configuration of the omni-directional vision sensor

Pictures taken and the associated video streaming are in a circular shape shown in Figure 3 (Swanepoel, 2009: 80).

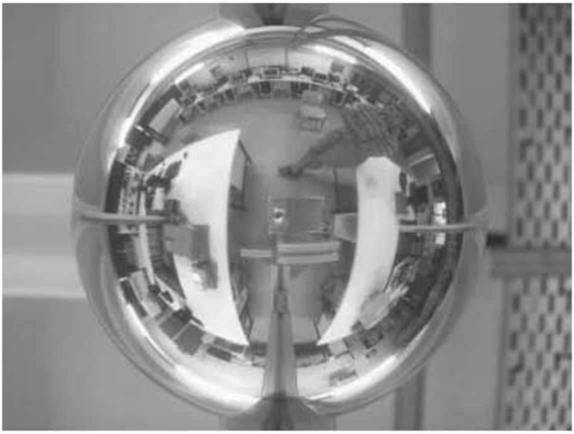

Figure 3 : Dome mirror picture before conversion into panoramic format

The image is then converted by means of a polar plot into a panoramic picture as shown in Figure 4.

Figure 4 : Omni-directional, converted panoramic picture

The polar transfer function was developed in MATLAB® and then implemented in Microsoft® Visual Studio® 2008 C# compiler. The reasons for this were to minimise transfer or calculation errors. Such errors are possible to be overlooked and to test for when writing the code directly into C#. It is also considered to be good practice to verify the accuracy and functionality of mathematical functions on a mathematical platform.

A test pattern generated for testing the accuracy of these transfers can be seen in Figure 5 . Such test patterns were used to measure the accuracy of the transfer of pictures from a round shape – hence, a conceptual simulated omni-directional image – into a panoramic view.

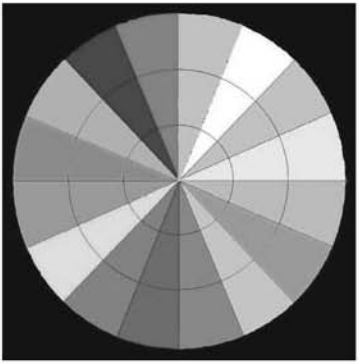

Figure 5 : Test pattern generated to assess polar transforms for accuracy

The result obtained by the MATLAB® generated executable file utilised in the C# graphical user interface (GUI) of the polar transfer function with the test pattern (Figure 5) as input can be seen in Figure 6.

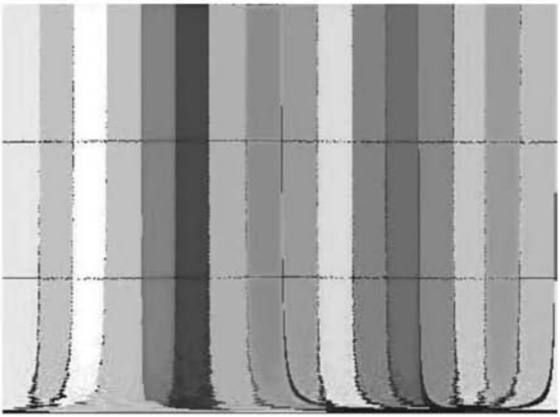

Figure 6 : Results generated by polar transfer

Figure 6 indicates a slight sinusoidal distortion of the transferred, converted image. This effect is due to selecting the wrong image centre point or selecting an incorrect centre point in the mirror and camera setup when running the software. This distortion, if any, is not visible and negligible in normal environmental images – as can be seen in Figure 7 and the transformed image, Figure 8.

Figure 7: Environmental picture in circular form (680 X 670 pixels).

Figure 8 : Polar transferred image of Figure 7

Initially a panoramic transfer was generated with the MATLAB® function with a step resolution of 0.4° (Users Guide, Image Acquisitioning Toolbox, 2003), (Users Guide, Image Processing Toolbox, 2003). This function does the transform on pixel level and is very time consuming. It took one second for the image of 3.6 megabytes to be transformed with this function, on an Intel® Core™ 2 Duo 3 GHz CPU with 1.99 GB of RAM. This was improved to approximately 0.38 seconds using a more relevant resolution of 1° as it did not make much difference in the observation of objects. On a more moderately sized picture of 640 by 480 pixels, this time could be reduced even more.

3. CONTROL AND SENSING OF THE AGV

The AGV with one jockey wheel is driven by the two back wheels. The platform, without the vision system, is shown in Figure 9. The wheels are driven by two 24V DC-motors. The driven wheels are controlled through a joystick and PIC micro-controller circuit board receiving and sending commands in series to and from a HMI (Data Sheet, Microchip, 1999).

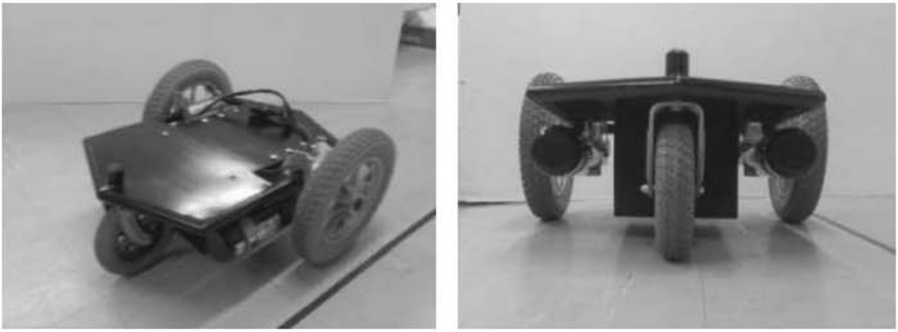

Figure 9: AGV-base platform with the vision configuration removed (Swanepoel, 2009: 29)

The serial data is being transmitted and received telemetrically through a WLAN connection. The WLAN was used for its radio frequency (RF) bandwidth and range (QUATECH, 2008). The delay between the images being received and the AGV commands transmitted was not of any significance in this initial test and trial runs.

The HMI is depicted in Figure 10 where the distances received from the ultrasonic sensors are being displayed. The route travelled as well as the controller commands and the video feed can be monitored on the same HMI.

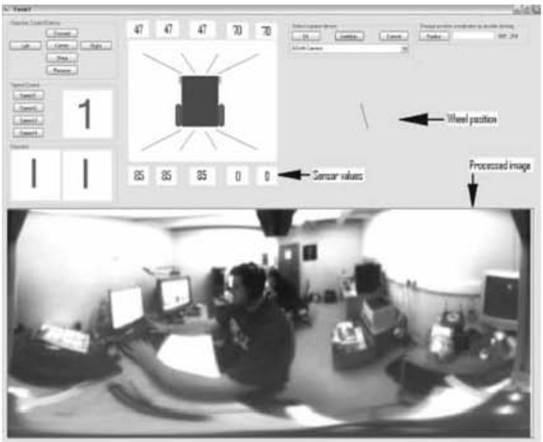

Figure 10: HMI screen capture (Swanepoel, 2009: 67)

Due to the lack of knowledge on the dynamics of the AGV, no control laws were inplemented over the delayed remote line. This limits the movement of the AGV in fast and constrained environments.

The platform utilises ultrasonic sensors to provide a proximity picture of the environment around the AGV (see Figure 11), (Swanepoel, 2009: 18). A total of eight sensors, based on a parking distance sensor unit, are used. This gives a virtual 2-D picture around the AGV, up to a distance of 1.8 metres.

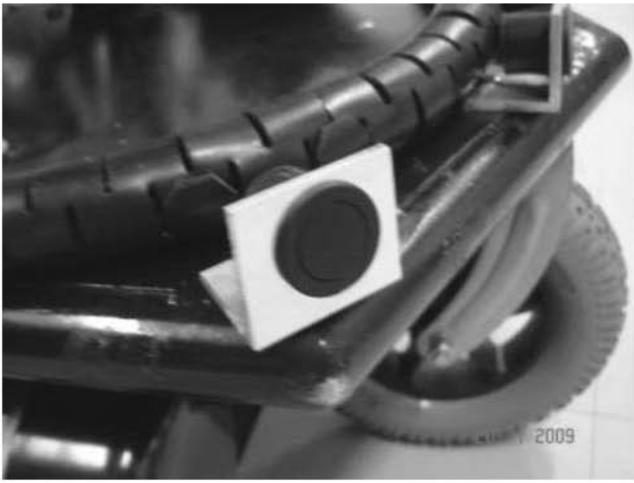

Figure 11: Ultrasonic sensors measuring the distance of the AGV from obstacles

A system of dead reckoning is implemented in determining the AGVs position in an environment. To enable the system to determine the distances travelled, wheel sensors are utilised. Figure 12 shows the position of these sensors and how it is mounted on the AGV (Swanepoel, 2009: 41).

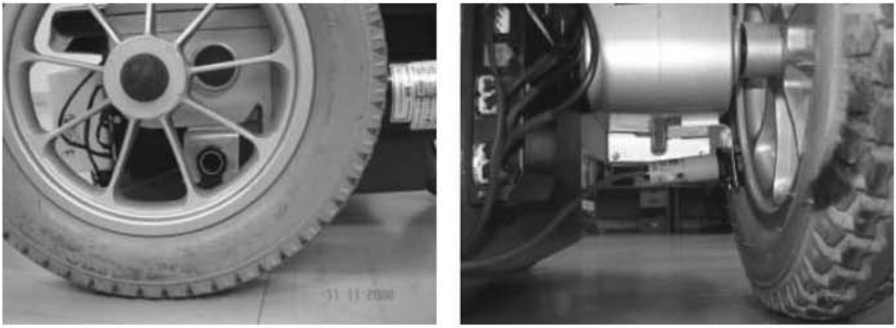

Figure 12: Wheel sensors for distance-travelled measurements

At this stage the AGV can navigate from point A to B without human intervention and can detect excessive movement, as well as stationary obstacles, in its region of interest. The amount of movement allowed and region of interest can be set by the user, depending on the current application.

4. MATHEMATICAL MODELLING

Throughout the research process, appropriate mathematical models were developed and tested in MATLAB®. The functions were then transferred to a C# compiler environment to create an .exe file for implementation on the hardware. This process of changing MATLAB® code in a m-file format to C# is not without conversion problems and/or human error. The system is currently utilising a laptop on the AGV as controller and communication hub for implementation of these algorithms.

MATLAB® Version 7.8 proved able to facilitate this whole process on the same software development platform. The result was an .exe file compiled with MATLAB® to be used by the C# GUI getting the user inputs and parsing the variable inputs for conversion to the .exe.

The AGV controller for navigation and the different sensors – as shown in Figure 1 – is controlled by a PIC16 series microcontroller from Microchip. The drive motors are controlled by receiving a value from the HMI via the WLAN converting it to an analogue signal for the motor controllers. The National Instruments USB-6009 multifunction I/O, already a manufacturing standard, could replace this microcontroller unit. This USB-6009 was not implemented but tested – with satisfactory results – for possible conversion later in the research if deemed necessary. The USB-6009 consists of eight, 14-bit multifunctional inputs and outputs as can be seen in Figure 13 (National Instruments, 2007). The device was initialised in MATLAB® as Dev1. Port 0 and 1 was used as outputs and inputs individually controlling the AGVs movement by only using MATLAB®. This was a test to evaluate MATLAB® to be the sole development and control platform for the AGV.

Figure 13: NI USB-6009 inside and outside an enclosure

This provided and proved that a similar AGV hardware platform could be configured, tested and implemented using a single software development platform like MATLAB®.

5. CONCLUSION AND FUTURE WORK

The AGV platform is still in a basic form and different applications and functionalities can be addressed by configuring it as add-ons in terms of hardware and software programming. However, the unit was successfully used in several research applications, proving its versatility and ease of reconfiguration. Continuous improvements are being made on the platform to suit different applications.

Future work could include:

- Improved calibration of omni-directional mirror, camera setup and conversion; since this was not optimised up to this stage of the research (Scaramuzza, 2008).
- The use of a hyperbolic mirror instead of a dome, as the results of Scaramuzza proved that a polar transfer function is not good enough for creating a good panoramic image (Scaramuzza, 2008).
- Localisation within a set environment to be configured by the user, utilising laser scanners and other RF techniques.
- Navigation and object recognition through a range of sensors and image acquisitioning and processing techniques (Bittencourt & Osório, 2007).
- Compression of data over the wireless feed.
- Object linking and embedding (OLE) for process control (OPC) compliant drivers and end sockets.
- Very high speed integrated circuit hardware description language (VHDL) transfer unit for implementation on the camera data and video stream.
- Standardising on a hardware control platform that could suit different applications in a reconfigurable environment.

The implementation of the PC104 platform could be considered in terms of the controllable hardware for real-world applications instead of the self-developed PIC controller.

6. REFERENCES

Bittencourt J.R. and Osório F.S. 2007. Adaptive Filters for Image Processing based on Artificial Neural Networks, UNISINOS – Universidade do Vale do Rio dos Sinos, Centro de Ciências Exatas e Tecnológicas Curso de Informática, Brasil,http://www2.computer.org/portal/web/csdl/doi/10.1109/SIBGRA.2000.895824 Retrieved on 22 September 2009.

Data Sheet, PIC16F87X, 28/40-Pin 8-bit CMOS FLASH Microcontrollers, Microchip, U.S.A., 1999.

Fend M., Bovet S. and Hafner V.V. 2007. The Artificial Mouse – A Robot with Whiskers and Vision, Artificial Intelligence Laboratory University of Zurich, http://www.amouse.de/ Retrieved on 22 September 2009.

Fernandes J.C.A. and Neves J.A.B. 2006. Using Conical and Spherical Mirrors with Conventional Cameras for 360° Panorama Views in a Single Image, ICM 2006 Proceedings, IEEE 3rd International Conference on Mechatronics, July 3-5, ISBN: 1-4244-9713-4/06.

MATLAB® Version 7.8 (R2009a), Help Demos, [Software Accessed].

National Instruments. 2007. Getting Started Guide, NI-DAQ™ mx for USB Devices, December.

Scaramuzza D., Omni-directional Vision: From Calibration to Robot Motion Estimation, A dissertation submitted to ETH ZURICH for the degree of Doctor of Science, pp. 27 – 50 and 145, 2008.

Swanepoel P., Kotze B. and Vermaak H. 2008. Omni-directional image sensing for automated guided vehicle, 2nd Robotics & Mechatronics Symposium Proceedings, 11 November, Bloemfontein, South Africa, ISBN: 978-0-620-42463-9.

Swanepoel P.J. 2009. Omni-directional Image Sensing for Automated Guided Vehicle, Dissertation M.Tech., School of Electrical and Computer Systems Engineering, CUT, Free State, Bloemfontein, April.

Treptow A., Cielniak G. and Duckett T. 2007. Comparing Measurement Models for Tracking People in Thermal Images on a Mobile Robot, Department of Computer Science University of Tuebingen, Germany and Department of Technology University of Oerebro, Sweden, http://robots.lincoln.ac.uk/users/gcielniak/papers/treptow05comparing.pdf Retrieved on 22 September 2009.

Users Guide, Image Acquisitioning Toolbox, Version 1, The Mathworks, March 2003.

Users Guide, Image Processing Toolbox, Version 4, The Mathworks, May 2003.

QUATECH. 2008. Products - 802.11b/g serial and Ethernet wireless solutions [online], Available at http://www.quatech.com/catalog/airbornedirect_80211bg.php Retrieved on 6 April 2009.