A Visual AGV-Urban Car using Fuzzy Control

Miguel A. Olivares-Mendez *Student Member, IEEE*, Ignacio Mellado,
Pascual Campoy *Member, IEEE*, Ivan Mondragon, Carol Martinez *Student Member, IEEE*
Computer Vision Group
Universidad Politénica de Madrid
José Gutierrez Abascal 2, 28006, Madrid, Spain
www.vision4uav.eu/?q=researchline/agv1
miguelangel.olivares@upm.es
www.vision4uav.eu

Abstract— The goal of the work described in this paper is to develop a visual line guided system for being used on-board an Autonomous Guided Vehicle (AGV) commercial car, controlling the steering and using just the visual information of a line painted below the car. In order to implement the control of the vehicle, a Fuzzy Logic controller has been implemented, that has to be robust against curvature changes and velocity changes. The only input information for the controller is the visual distance from the image center captured by a camera pointing downwards to the guiding line on the road, at a commercial frequency of 30Hz. The good performance of the controller has successfully been demonstrated in a real environment at urban velocities. The presented results demonstrate the capability of the Fuzzy controller to follow a circuit in urban environments without previous information about the path or any other information from additional sensors.

I. INTRODUCTION

Autonomous mobility is a central problem in Robotics, and more precisely, the control of autonomous guided vehicles (AGV) is a very ambitious non-linear control problem. In addition, if it is focussed on car-like vehicles, the difficulty is increased by the complexity of all the dynamic components, being very difficult to obtain the actual vehicle dynamic model. Despite this complexity, some works have used linear control methods to control an AGV using a PID, like [1]. Nowadays there are more people applying non-linear control system to this kind of vehicles, being Fuzzy Logic the most used technique, as it is seen in the reported literature [2] and [3]. Furthermore, this technique has an important role in the research with car-like mobile robots, like in [4] and [5], in which a golf car is used. The use of real urban cars in robotics has a major sponsor: the US Department of Defence through its various agencies. Notable examples include the DEMO I, II and III projects [6] and the DARPA Grand Challenge [7]. Two of the most important researchers in this area are Sebastian Thrun and M. Montemerlo, who won the DARPA Grand Challenge in 2005 [8], and developed another model to win the second place at the 2008 DARPA Urban-Challenge [9]. The hard requirements of these competitions forced to researchers to use many sensors to measure the environment with high accuracy and, this is unattainable for a big part of the research community. To reduce costs, some works have been performed with just a visual sensor, for instance, surface classification like [10] or object detection like [11].

Computer vision has been also used for guidance tasks with mobile robots like [2], with a camera pointing forwards and [1] with a camera pointing downwards and a field of view of 1 meter high.

In this paper is presented an AGV -urban car (Citron C3)that is guided by a painted line on the ground, with one camera capturing the forthcoming 30cm in front of the vehicle. To control the steering of the car, a Fuzzy controller has been developed. Excellent results were obtained by testing the system in a real circuit for more than 3 km. This paper is organized as follows. Section II explains the system of the urban vehicle. Section III shows the visual hardware and the visual algorithm. The Fuzzy controller is described in detail in Section IV. Some of the successful experiments are presented in Section V with detailed plots. To finish with the conclusions in Section VI.

II. CAR SYSTEM

The car that was used for this research is a commercial vehicle "Citron C3 Pluriel" (Figure 1). To move the steering wheel a manipulation of the power-assisted steering motor was used. This assistance system consists of an electric DC motor attached to the steering rack trough a gear. This motor drives the steering to the action of the driver on the steering wheel. This action is measured through a torque sensor located in the steering bar. The signal from the sensor is received by a control/power until that sends a PWM signal to the motor, to assist the steering movement. This device allowed a fast automation since the mechanical and electrical elements were already installed in the car. For our purpose, the connections of the motor were cut, and it was attached to a a MAXON ADS 50/10 servo amplifier, with 240 Watts of peak power at 12 V. This card is able to generate a PWM signal whose duty cycle is proportional to an analog ± 10 V input signal. This input signal is generated by an Advantech USB-4711A acquisition card that is connected to an onboard computer. The necesary information to feedback the control system is provided by an absolute. The encoder gives the angular positions at a rate of 100Hz.

During the tests, a human driver controlled the speed of the vehicle manually. In order to measure the vehicle speed, a non-contact speed sensor L-CE Correvit was installed. It

Fig. 1.   Automated Citron C3 Pluriel

sends the measured speed at a 100Hz rate to the onboard computer.

The guiding line paint was produces with special pigments that appear blue-coloured when they are lit whit a bright ultraviolet light, while staying uncoloured under the normal light. In order to keep the illumination under control a special structure was designed and installed in front of the car. This structure is made up of a black metal box (Figure 2), and it contains the camera the camera and the ultraviolet lamp. The restricted height (47 cm) of the box forced us to use a wide angle low distortion lens for the camera, in order to capture the whole scene at the bottom of the box, which is 60x40 cm. The box is isolated at its base from the daylight by rubber tabs and brushes. Despite this benefit, this isolation reduces the visual field to an area of 50x30 cm.

III. Visual System

The visual system is composed of all the hardware and software to extract the relative position between the car and the line.

A. Visual Hardware

For these tests, we use a laptop with a Core-Duo Centrino processor, running at 2.6GHz, and 4 Gbytes of RAM. The

Fig. 2.   Black metal box of the visual system

operating system is Ubuntu 10.4. The camera is a Firewire Unibrain 520C, with a resolution of 320x240 pixels at 30 fps and a wide angle, low distortion lens attached. The whole set gives a field of view of 125 degrees and a working distance of 10 mm. For the lighting, a special UV lamp with a wavelength of 365 nm (black light) is needed to excite the pigments of the line. To avoid black frames because of flickering, the operation frequency of the lamp is 25 kHz.

B. Visual Algorithm

For the detection of the line, a custom real-time computer vision algorithm was designed. The algorithm is able to detect the lines centroid and orientation under harsh conditions, such like a partially occluded and poorly painted line on a rough terrain, coping with non-compact line shapes. The line detection has been successfully tested at up to 30 kph.

On the front-end of the visual system, the camera captures the scene which is lit with UV light at 30 fps. First, a colour-based segmentation is performed on YUV space. Despite some other colour spaces were tested, YUV was found to be the best performer under different light conditions. A rectangular prism inside the YUV colour space is defined, so that only the pixel values inside this volume are considered to be part of the line. The result is a binary image where only the line pixels are set. This method proved to be robust detecting lines of different blue tones and brightness.

In the binary image, every 8-connected pixel group is marked as a blob. At the first step, to reduce the noise, blobs having an area outside a defined range are discarded. Then, for every survivor, centroid, dominant direction and maximal length are calculated, and those being too short are ignored. The remaining blobs are clustered according to proximity and parallelism, so each cluster becomes a candidate line. The centroid and dominant direction of each candidate line are calculated from the weighted sum of the features of its component blobs, being the weight of each blob proportional to its relative area. In this way, the algorithm is able to accurately detect lines that are fragmented because of ageing.

Finally, from the whole set of candidate lines, a detected line must be selected for the current frame. In order to do that, the distance between the centroids of every candidate line in the current frame and the detected line in the previous frame is measured. If the smallest distance is higher than a certain threshold, the detected line will be the leftmost or rightmost candidate line, depending on the user-defined criterion. Otherwise, the closest candidate line is taken as detected line. This mechanism avoids switching to fake lines when there are traces of old paintings along the circuit, even when it is deteriorated.

The algorithm outputs whether the line is detected or not and, if it is, it also outputs the error of the line in the x-axis from to the centre of the image and the direction of the line, expressed as an angle.

IV. Fuzzy Controller

The steering control of the car includes two components. The first one is the Fuzzy controller and the other one is

the integral of the error. The latter is added at the end of the control loop to the output of the controller, making a structure of $Fuzzy + I$, as it is shown in Figure 3.
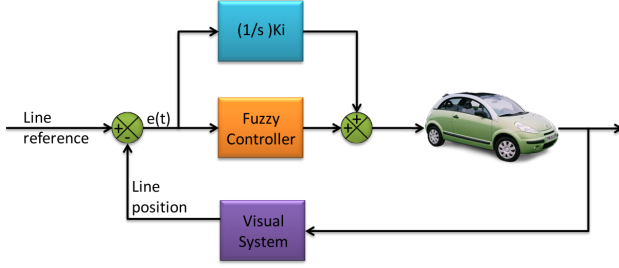

Fig. 3.   Control loop of the visual servoing system.

The Fuzzy controller was implemented using the MOFS (*Miguel Olivares' Fuzzy Software*). This software was used previously to implement Fuzzy Controllers in other different platforms like a wheelchair [12] or in an unmanned helicopter, where it was applied to control a pan and tilt visual platform onboard the UAV [13] and for the autonomous landing of the aircraft [14]. With this software, it is possible to easily define a fuzzy controller with the required number of inputs and to select the type of membership functions, the defuzzification model and the inference operator. A more detailed explanation of this software can be found in [15].

The controller has two inputs and one output. All are fuzzyfied using triangular membership functions. The first input is defined as the error between the centre of the image and the centre of the line to follow (Figure 4). The second input is the difference between current and previous error (Figure 5). The output of the controller is the absolute turn of the steering wheel to correct this error, in degrees (Figure 6). To obtain this output, 49 if-then rules were defined. The developed fuzzy system is a Mamdani type that use a height weight defuzzification model with the product inference model in Equation 1.

$$y = \frac{\sum_{l=1}^{M} \overline{y}^l \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) \right)}{\sum_{l=1}^{M} \prod_{i=1}^{N} \left(\mu_{x_i^l}(x_i) \right)} \qquad (1)$$

Where N and M represent the number of inputs variables and total number of rules respectively. $\mu_{x_i^l}$ denote the membership function of the lth rule for the ith input variable. $\overline{y}^l$ represent the output of the lth rule.
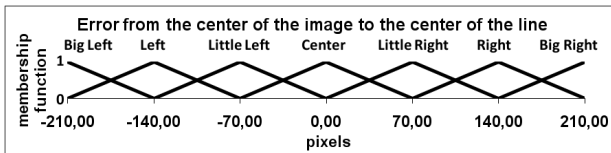

Fig. 4.   First input variable of the Fuzzy controller: the error between the centre of the line and the centre of the image, in pixels.

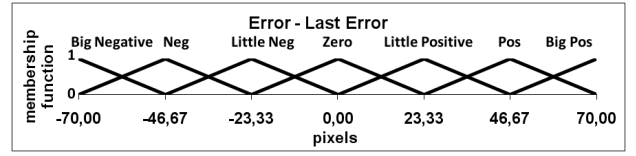The calculation of the integrator value is shown in Equation 2.

Fig. 5.   Second input variable of the Fuzzy controller: the difference between the last error and the actual, in pixels.
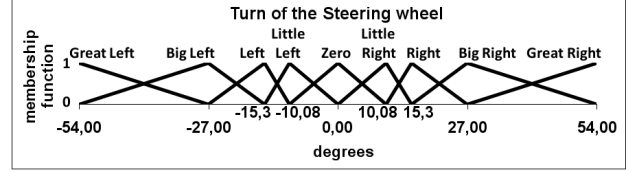

Fig. 6.   Output variable of the Fuzzy controller: the steering wheel angle, in degrees.

$$I_t = I_{t-1} + e \times \frac{1}{t} \times Ki \qquad (2)$$

Where e is the current error between the centre of the line and the centre of the image, t is the framerate, and Ki is a constant that appropriately weights the effect of the integrator, and for this case is equal to 0.6.

The initial idea of this work was to develop a controller for a circuit with short radius curves. In such conditions, the speed of the car can not be very high. Thus the actual velocity of the car is was not included in the Fuzzy controller, but it is taken into account multiplying the fuzzy output by $\frac{10}{v}$, being v the current velocity of the vehicle. The definition of the numerator value of this factor is based on the velocity, in kph, during a skilled human driving session, in which data was acquired to tune the rule base of the fuzzy controller. It is practically impossible for a human to drive faster than 10kph while keeping the line in following error low enough to meet the requirements of the application. This is because the driver only sees 30cm forward, and, at that speed, the contents of this area change completely every 0.108 seconds

The driving session performed by the human at 10kph output the necessary training data to modify the initial base of rules of the controller and the size of the fuzzy sets of its variables. For the definition of the fuzzy sets, a heuristic method was used based on the extraction of statistical measures from the training data. For the initial base of rules, we used a supervised learning algorithm, implemented in MOFS. This algorithm evaluates the situation (value of input variables) and looks for the rules that are involved in it (active rules). Then, according to the steering command given by the human driver, the weights of these rules are changed. Each time that the output of an active rule coincides with the human command, its weight will be increased. Otherwise, when the output differs from the human command, its weight will be decreased by a constant. Anytime the weight of a rule becomes negative the system sets the output of the rule to the one given by the human driver. Further details of the software are given at [15].

V. Experiments

To test the fuzzy controller, a closed loop line was painted with an oval shape, as shown in Figure 7. The two curves are 20 and 11 meters of radius and 72 and 34 meters long, respectively. The stretches are 40 and 44 meters long. The total length of the circuit is 190 meters. First, we present system behaviour results after two different step perturbations were applied at different velocities and circuit curvatures. Subsequently, results for a continuous 18 laps test are presented. The total distance driven during the second test is 3.5km.

Fig. 7.   Representation of the circuit on a Google Earth image.

A. Step perturbation test series

In order to measure how good the fuzzy controller is, a set of step tests was made. The step value is 50 pixels, equivalent to more than 6 cm. This step was applied to the car at different velocities in straight lines and curves. Some of the results of these tests are shown after these lines.

Figure 8 shows the error measured when a $+50$ and -50 pixels step perturbation is applied to the system at 10 kph with a resulting RSME value of 7.166 cm. At it is shown, the system corrects the error in just 27 frames, which is about 1 second for an average rate of 28 rfames per second during the test. The angle of the steering wheel versus the controller commands is shown in Figure 9, in which a delay of $7 - 8$ frames in the steering wheel action may be noticed. Ignoring this delay, the effective settling time would stay around 20 frames or 0.7 seconds.

Figures 10 and 11 represent the results for a step perturbation test at 15 kph in a curve. For this test the value of the RMSE is 6.8574 and the settling time is less than a second (25 frames).

B. Continuous driving tests

In this tests, the car covered 18 laps of the circuit. In Figure 12 the measured error during the whole test is shown. In this case, the RMSE was 5.0068 cm.
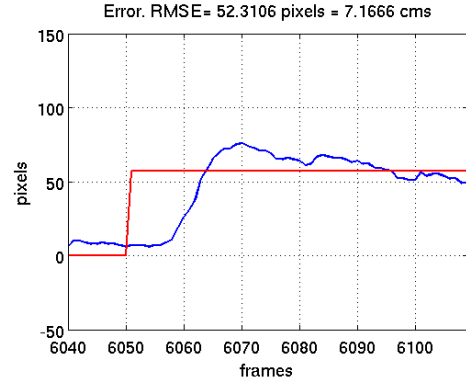

Fig. 8.   Representation of the error, in pixels, during the 50 pixels step test at 10 kph in a straight line. The measured RMSE is also shown at the top.
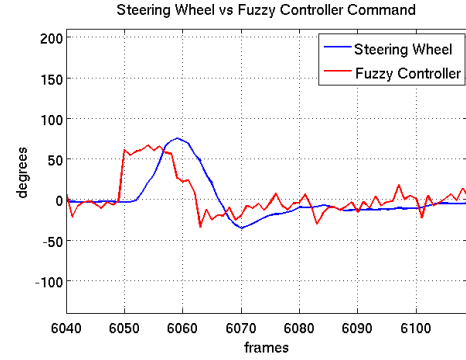

Fig. 9.   Evolution of the steering wheel angle versus the controller commands during the 50 pixels step test at 10 kph in a straight line.
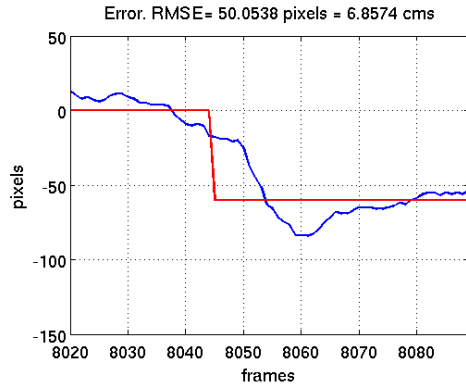

Fig. 10.   Representation of the error in pixels during the 50 pixels step test in straight at 15 kph. The value of the RMSE of the test in this part of the circuit is 6.8574 cm.

Figure 13 shows the comparison between the controller commands and the measured angle of the steering wheel. In the Figure, the changes between straight lines and curves may be noticed. In the straight lines, the steering wheel stays around zero degrees, while it turns between -100 and -150 degrees in the first curve, and between -150 and -300 in the second one. It is more easyly see in Figure 14, in which the plot is scaled to show only one lap.

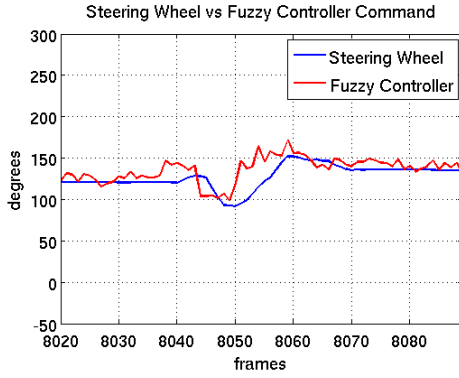In Figure 12 large error peak of even 170 pixels appear

Fig. 11. Reperesentation of the movements of the steering wheel versus the value of the commands sent by the controller during the 50 pixels step test in straight at 15 kph.
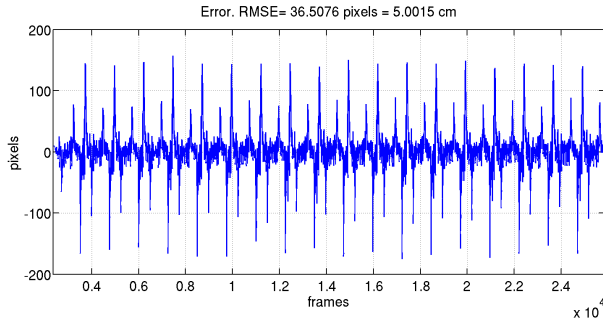

Fig. 12. Representation of the error in pixels during the 18 laps to the circuit. The value of RMSE for this test is 5.0015 cm.
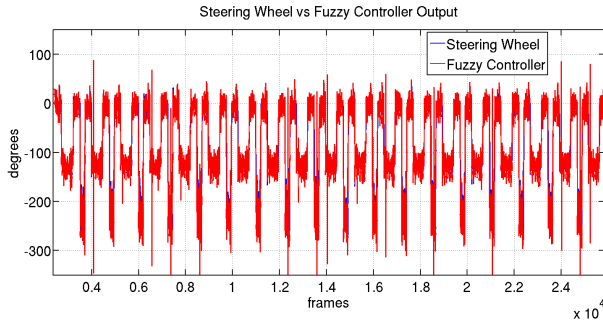

Fig. 13. Reperesentation of the movements of the steering wheel versus the value of the commands sent by the controller during the test of 18 laps to the circuit.
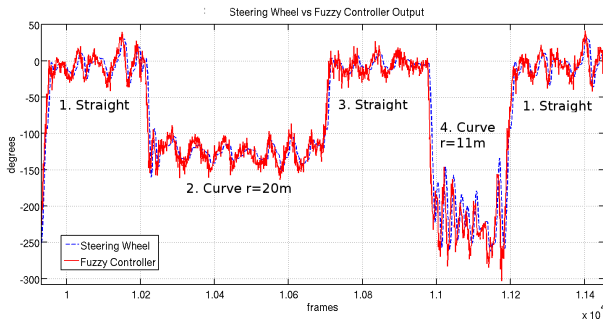

Fig. 14. Zoom to one lap of the circuit.

at every curvature change. However, they are decreased in a few frames by the controller. This errors appear because the circuit was not designed with clothoids. Therefore, curvature discontinuities happen when changing from straight line to curve and vice-versa. Figure 15 shows a zoom of one of this instants in which a peak of -171 pixels occurs. The evolution of the error is plotted in Figure 15(a), while the output of the controller and the steering wheel angle are in Figure 15(b).
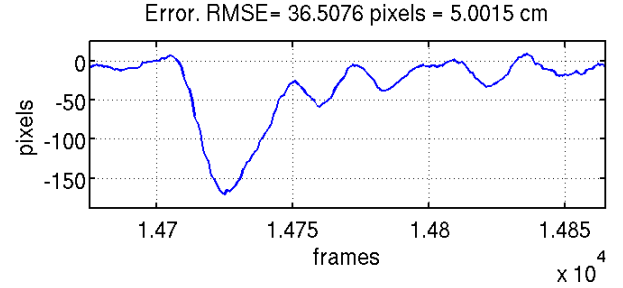
(a) Zoom of the error
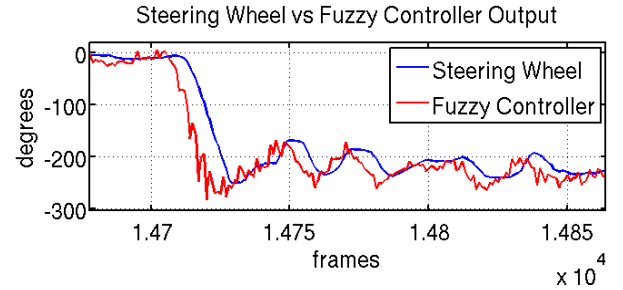
(b) Zoom of the steering wheel angle and controller commands

Fig. 15. Zoom of 170 pixels step at the beginning of the second curve.

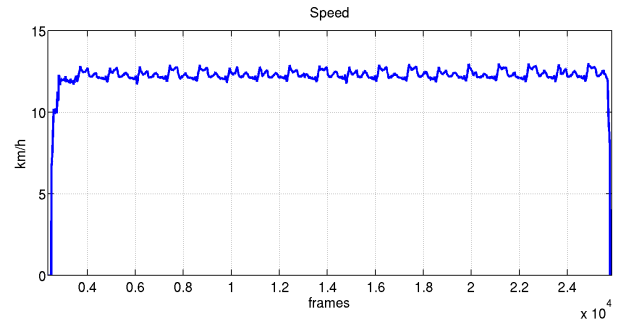The evolution of the vehicle speed is depicted in Figure 16, which covers speeds between 12 and 13 kph.

Fig. 16. Measure of the vehicle speed during the 18 laps test.

In [16] is possible to see a video of some of these tests.

VI. CONCLUSION

This work presents a low-cost visual line-guided system for an urban-car controlled by a Fuzzy Logic controller. Strong results on real-world tests are presented in order to check the behavior of the controller. The quick response of the vehicle with step command tests and the execelent line-following behavior during long distance tests support the

quality of the development control system. The controller reacts in about one second to a step perturbation of more than 6cm in the visual system of the car. It must be taken into account that the steering wheel of the car has a reaction delay of 0.7 seconds. Another long test of more than 3km (18 laps of the circuit) was run with a resulting RMSE of just 5.01 cm. The successful results show the excellent behaviour of the controller, despite the reduced perception span given by the camera, which puts some limitations on the AGV performance, being the maximum attainable velocity the most important.

Currently, we are working on a custom visual signaling system to acquire information about the position of the car and the forthcoming features of the path. This update will give some anticipation capabilities that will enable higher velocities and a better controller behaviour. On the other hand, we intend to join all the four inputs inside the fuzzy controller, including the error, its derivative, its integral and the car velocity.

REFERENCES

[1] Jin-Woo Lee, Sung-Uk Choi, Chang-Hoon Lee, Young-Jin Lee, and Kwon-Soon Lee. A study for agv steering control and identification using vision system. In *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, 2001.

[2] W.S. Wijesoma and K.R.S. Kodagoda. Design of stable fuzzy controllers for an agv. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, 2000.

[3] Yu Dianyong and Xue Hui. Application of fuzzy control method to agv. In *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, volume 2, pages 768 – 772 vol.2, 2003.

[4] Quan Yuan, Yunzhou Zhang, Hao Wu, and He Wang. Fuzzy control research in the courses of smart car. In *Machine Vision and Human-Machine Interface (MVHI), 2010 International Conference on*, pages 764 –767, 2010.

[5] K.R.S. Kodagoda, W.S. Wijesoma, and E.K. Teoh. Fuzzy speed and steering control of an agv. *Control Systems Technology, IEEE Transactions on*, 10(1):112 –120, January 2002.

[6] C.M. Shoemaker and J.A. Bornstein. The demo iii ugv program: a testbed for autonomous navigation research. In *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings*, pages 644 – 651, September 1998.

[7] DARPA. Darpa Gran Challenge http://www.darpagrandchallenge.com, 2011.

[8] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Winning the darpa grand challenge. *Journal of Field Robotics*, 2006. accepted for publication.

[9] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 2008.

[10] Liang Lu, C. Ordonez, E.G. Collins, and E.M. DuPont. Terrain surface classification for autonomous ground vehicles using a 2d laser stripe-based structured light sensor. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2174 –2181, 2009.

[11] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 18:81–102, 2004.

[12] Miguel Olivares and J.A.F. Madrigal. Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments. *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6, Oct. 2007.

[13] Miguel. Olivares, Pascual Campoy, Carol Martinez, and Iván Mondragon. A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. October 2009.

[14] M.A. Olivares-Mendez, I.F. Mondragon, P. Campoy, and C. Martinez. Fuzzy controller for uav-landing task using 3d-position visual estimation. In *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, pages 1 –8, 2010.

[15] Iván Mondragn, Miguel Olivares-Mndez, Pascual Campoy, Carol Martnez, and Lus Mejias. Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. *Autonomous Robots*, 29:17–34, 2010. 10.1007/s10514-010-9183-2.

[16] Computer Vision Group. Polytechnic University of Madrid. Agv test video http://www.vision4uav.eu/?q=researchline/agv1, 2010.