# Monocular Vision: A Real-Time Perception Toolkit for Mobile Robots in Outdoor Environments

A. Miranda Neto, A. C. Victorino and J. V. Ferreira

***Abstract*—Many applications for control of autonomous platform are being developed and some important aspects are: (a) the estimation of drivable image area and (b) the excess of information, frequently redundant, that imposes a great computational cost in data processing. In this way, we have proposed (i) a robust algorithm for detecting the horizon line to generate (ii) the navigable area. It permits to investigate dynamically only a small portion of the image (road) ahead of the vehicle. Moreover, taking into account the temporal coherence between consecutive frames, we also have proposed a set of tools based on Pearson's Correlation Coefficient: (iii) a discarding criteria methodology applied as (iv) a dynamic power management solution; (v) an environment observer method which selects automatically only the regions-of-interest; and taking place in the obstacle avoidance context, (vi) a method for collision risk estimation for vehicles in dynamic and unknown environments. Applying the PCC to these tasks has not been done yet, making the concepts unique. All these solutions have been evaluated from real data obtained by experimental vehicles.**

## I. Introduction

The perception of the environment is a major issue in autonomous and (semi)-autonomous systems. In the last three decades, visual navigation for mobile robots has become a source of countless research contributions.

The primary interest in this work, the machine vision is a powerful means for sensing the environment and is widely employed to deal with a large number of tasks in the automotive field [1]. For example, determining the area of free road ahead (and correlated information) is a key component of several driving assistance modules [2]. This modules use monocular camera systems instead of stereo camera systems because monocular systems have advantages in terms of reduced costs and the ease with which they can be fitted to vehicles [3]. The objective in terms of cost to fill ADAS functions has to be very lower than the current Adaptive Cruise Control (500 Euros) [4]. Additionally, the monocular vision contribution to the DARPA Grand Challenge [5] shows that the reach of lasers was approximately 22 meters, whereas the monocular vision module often looks 70 meters ahead.

To better understand our proposal, after acquiring the image by a monocular camera, the system fulfills its role through the tasks in the following layers: Section II (a) Environment observer method; Section III (b) Navigable area detection; Section IV (c) Autonomous navigation: a reactive perception system. The results are presented in each respective section and the conclusions in Section V.

## II. Environment Observer Method

### A. Pearson's Correlation Coefficient

The Pearson's Correlation Coefficient (PCC) [6] in (1) is widely used in statistical analysis, pattern recognition and image processing:

$$r_1 = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2}\sqrt{\sum_i (y_i - y_m)^2}} \quad (1)$$

where $x_i$ is the intensity of the $i^{th}$ pixel in image 1, $y_i$ is the intensity of the $i^{th}$ pixel in image 2, $x_m$ is the mean intensity of image 1, and $y_m$ is the mean intensity of image 2.

### B. Discarding Criteria methodology

The discarding criteria was presented as a simple solution to improve the performance of a real-time navigation system by choosing, in an automatic way, which images should be discarded and which ones should be treated at the visual perception system [7]. Basically, if the PCC indicates that there is a high correlation between a reference frame and another new frame acquired, the new frame is discarded without being processed (for example, the system can repeat a last valid command). Otherwise, the frame is processed and it is set as the new reference frame for the subsequent frame. In our case, the inclusion of an automatic image discarding method leads in a reduction of the processing time. Although the system spends some milliseconds computing the PCC, it gains much more time, in some cases, discarding more than 90% of the images.

### C. Real-Time Dynamic Optimization

A robot can have many periodic tasks, such as motor and sensor control, sensing data reading, motion planning, and data processing. It may also have some aperiodic tasks, such as obstacle avoidance and communication. Moreover, for mobile robots, the tasks' deadlines are different at different traveling speeds. At a higher speed, the periodic tasks have shorter periods [8].

Based on the PCC variation and by exploiting the temporal coherence between consecutive frames, it is proposed a new environment observer method [9]. This monocular-vision

Arthur de Miranda Neto and Janito Vaqueiro Ferreira are with the Autonomous Mobility Laboratory (LMA) at FEM/UNICAMP, Brazil. Alessandro Correa Victorino is with the Heudiasyc laboratory UMR 7253 CNRS/UTC, France.

system observes if there are no significant changes in the environment, permitting that some logical components may be shut down to save processor energy consumption, and/or to make the CPU available for running concurrent processes.

The Fig. 1 (a) shows an autonomous displacement through the Mojave Desert [10]. In Fig. 1 (b), due to PCC nature, taking the first frame of the Fig. 1 (a) as reference frame, a lower value of correlation is achieved when it is closer to the vehicle. That is, when the derivative approaches its maximum point (black line), there is the obstacle detection.

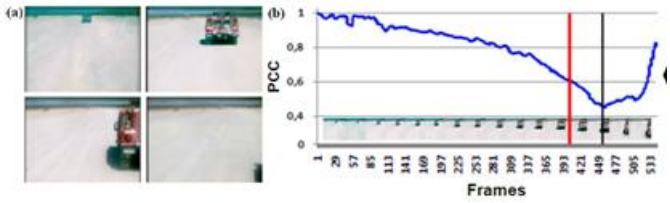


Figure 1. (a): The frames of the desert video [10]; (b) From a reference frame, its correlation with all others; Blue line: the Pearson's correlation in (1); the vertical black line: maximum point before collision; the vertical red line: Empirical Risk.

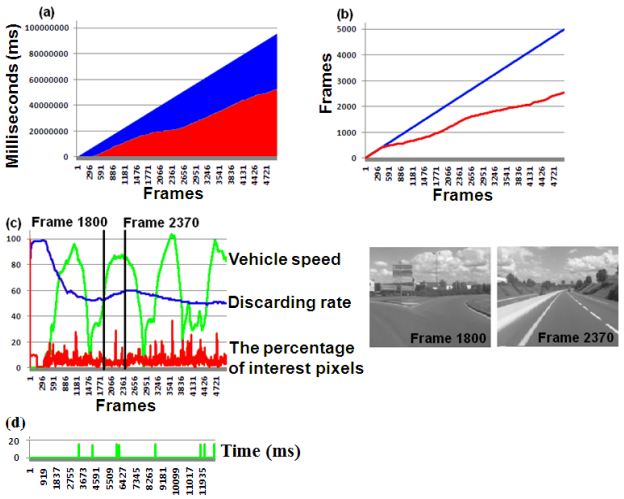


Figure 2. Real environment in France: (a) in blue: the cumulative impact computations (ms); in red: the cumulative computations (ms) by using the discarding criteria. (b) In blue: the number of frames; in red: the number of discarded frames by using the discarding criteria. (c) In blue: discarding rate; In red: the percentage of interest pixels; In green: The vehicle speed; In the analysis window, represented by two black vertical lines, the performance evaluation of the discarding criteria in acceleration from 37 Km/h to 86 Km/h; (d) Green line: computational time.

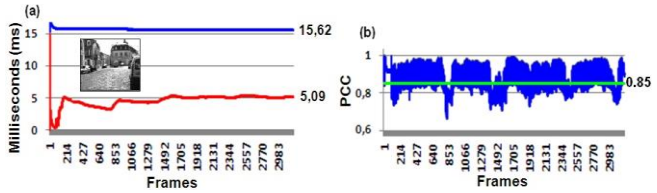


Figure 3. Real environment in France: The computational mean time of a horizon finding algorithm [11] in unknown and urban environment; (a) The red line: the computational mean time was 5.09 ms with the discarding criteria; (a) The blue line: the computational mean time was 15.62 ms without the discarding criteria; (b) The green line: an empirical PCC threshold equal to 0.85; (b) In blue: the performance based on discarding criteria: above the green line, it presents all discarded images.

The Figures 2 and 3 show the performance of this method in real, dynamic and unknown environments. For all these cases, the discarding rate remains over 65%. Fig. 3 (a) presents the computational mean time of a horizon finding algorithm [11] in unknown and urban environment. In this way, from an empirical PCC threshold equal to 0.85, the red line shows that the computational mean time was 5.09 ms, against 15.62 ms without the discarding criteria. In Fig. 3 (b), above the green line, it presents the discarded images.

### *D. Automatic Regions-of-Interest Selection based on PCC*

One important aspect is the excess of information, frequently redundant, that imposes a great computational cost in data processing. We have considered a robot equipped with a vision perception system, and we proposed [12] an automatic regions-of-interest selection based on PCC, which processes it in real time.

According to the Pearson's correlation, in a certain analysis window (pair of frames), if the obstacle/object occupies a big portion of the scene, the PCC threshold tends to be low. Conversely, if obstacle/object occupies a small portion of the frame, it means that it is away from the vehicle and the system will have time enough to react. However, in real-time obstacle avoidance, for example, where are these interest points/pixels? Or, in a sequence analyzed, which pixels of the pair of images contributed most to the Pearson's coefficient computed? Which of them really need to be reprocessed?

Right after the Pearson's correlation in (1), for each pair of pixels analyzed in (2), the only possible result is: [-1 or +1]. That is, all pixels with intensities below these means will be candidates for interest points (ROI).

$$r_2 = \frac{(x_i - r_{1Xm})(y_i - r_{1Ym})}{\sqrt{(x_i - r_{1Xm})^2}\sqrt{(y_i - r_{1Ym})^2}} = \begin{cases} -1 \\ or \\ +1 \end{cases} \qquad (2)$$

where $x_i$ is the intensity of the $i^{th}$ pixel in image 1, $y_i$ is the intensity of the $i^{th}$ pixel in image 2, $r_{1Xm}$ and $r_{1Ym}$ were obtained in (1): i.e.: $x_m$ and $y_m$ from (1).

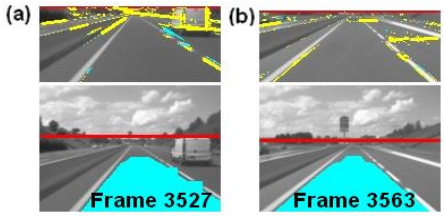


Figure 4. Real environment in France: After the horizon finding algorithm performance [11], red line: (a) Speed 97.01 Km/h, the interest pixels (yellow) represent 5% of the image; (b): Speed: 100.2 Km/h, the interest pixels (yellow) represent 2% of the image.

From these interest points, the Interactive Thresholding Algorithm (ITA) [12] reclassifies the background and foreground pixels based on Otsu Thresholding Method [13]. Fig. 4 *(b) present the results of this process, where the red pixels (interest points) represent $r_2 = -1$. Taking as base an image resolution equal to image 96x72 taking from a desert

video, by processing the images only when $r_2 = -1$, only about 205 thousand points were processed, instead of 3.7 million points. In off-road context the same result is observed, only about 10 million points were processed, instead of 48 million points [12], [14].

Results for different types of image texture (road surfaces) were selected and its results are presented in [14]. For obstacle avoidance task, the Fig. 4 presents results at high speed on real-time conditions.

### *E. Collision Risk Estimation based on PCC*

The collision warning algorithms typically issue a warning message when the current range to an object is less than the critical warning distance, where the safety can be measured in terms of the minimum time-to-collision (TTC) [15]. To calculate the TTC, several techniques are presented in the literature [16], [17]. Measuring distances is a non-native task for a monocular camera system [16]. However, TTC estimation is an approach to visual collision detection from an image sequence. In this way, we have presented a novel approach to obtain Collision Risk Estimation (CRE) based on PCC from a monocular camera [18].

To better understand, let's go back to Fig. 1 where the robot Stanley has moved at an average speed of 30.7 km/h [19]. In Fig. 1 (b), a lower value of correlation is achieved when it is closer to the vehicle (black line). For our proposal an Empirical Risk-of-Collision needs to be defined. This is represented in Fig. 1 (b) by the red line. Taking into account this $R_c$, the CRE is estimate in (3):

$$CRE_s = \frac{R_c}{(1 - r_1)} \tag{3}$$

where 1 (one) represents the reference frame and $r_1$ was obtained in (1) and $R_c = 1 - 0.6$.

Fig. 5 and Table I present the performance of the CRE in dynamic and unknown environment. These results were obtained in real conditions using our experimental vehicles. Since in real conditions this monocular-vision system has been designed to investigate only a small portion of the road ahead of the vehicle, where the absence of other vehicles has been assumed [1], the Fig. 4.a-(*a) presents the fix analysis region (yellow line). As shown in [18], the computational mean time of the CRE process was equal to 7.8 ms.

## III. NAVIGABLE AREA DETECTION

### *A. Horizon Finding Algorithm (sky removal)*

For land vehicle navigation, the monocular vision systems have been applied to investigate the road information, and in order to decrease the volume of data for processing, some systems have been designed to investigate only a small portion of the road ahead of the vehicle where it is unlikely the existence of other vehicles [1]. For example, some systems seek to identify the sky region because this is not a region of interest, and therefore the horizon line threshold is applied to subtract a road image [20]. Stanford Racing Team [5] implemented the horizon finding algorithm originally proposed by [21] to eliminate all pixels above that horizon.

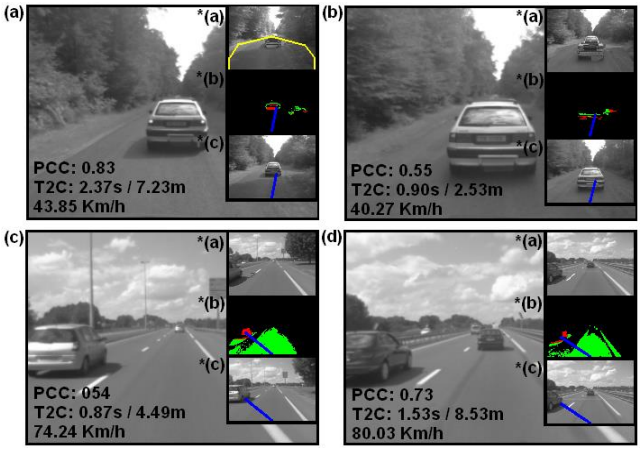


Figure 5. Real environment in France: *(a): the reference frame after the region-merging algorithm presented in [18]; *(b) ITA results [12]; *(c) Obstacle direction based on the center of area of the red points in *(b).

| Frames | (1- $r_1$) | Variation in the Range | Risk of Collision | CRE in Sec(s) | Distance in Meters |
|---|---|---|---|---|---|
| (a) 1001 | (1-0.8315) | 0.1685 | ( $R_c$ / 0.1685) | 2.37s | 7.23m |
| (b) 1024 | (1-0.5584) | 0.4416 | ( $R_c$ / 0.4416) | 0.90s | 2.53m |
| (c) 1139 | (1-0.5411) | 0.4589 | ( $R_c$ / 0.4589) | 0.87s | 4.49m |
| (d) 4654 | (1-0.7394) | 0.2606 | ( $R_c$ / 0.2606) | 1.53s | 8.53m |

TABLE I. Relationship between frames of the Fig. 7 and Collision Risk Estimation (CRE).

In this way, from a dynamic threshold search method, a robust horizon finding algorithm that finds the horizon line was proposed by us [11] and applied to generate the navigable area. It permits to investigate dynamically only a small portion of the image (road) ahead of the vehicle. This algorithm is robust to illumination changes and does not need any contrast adjustments. Although the Otsu method is an excellent method to choose an ideal threshold, it considers for all cases the information of the image as a whole (global information). After sky removal, it permits to investigate only a small portion of the image ahead of the vehicle, and the new result can be seen in Fig. 6.

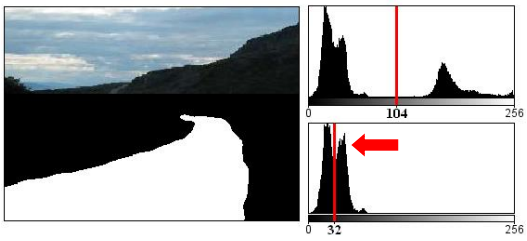


Figure 6. Left: Otsu segmentation after sky removal; Right: Global Otsu threshold (104); new Local Otsu threshold (32) after sky removal [11].

Figures 7 and 8 present the performance of this horizon line detection process in unknown environments: desert road videos available by DARPA [10], [19]; urban environment data obtained by our intelligent vehicles. For each image a real horizon line (black) was registered manually. In (a) the green line represents the Otsu horizon line detection; in (b) the magenta line represents the weighted average of the Hough

transformation lines; in (c) the blue line represents the weighted average between the Otsu horizon line detection and the Hough transformation [22] result; in (d), finally, the red line represents the robust horizon finding algorithm based on Otsu segmentation, Hough transformation and Kalman filtering. Fig. 9 presents the computational mean time of this algorithm in unknown environments. In the Fig. 9 (a) the mean time was as 6.27 ms without the discarding criteria [7]. In the Fig. 9 (b), from an empirical PCC threshold, the mean time was 1.20 ms with the discarding criteria [7].

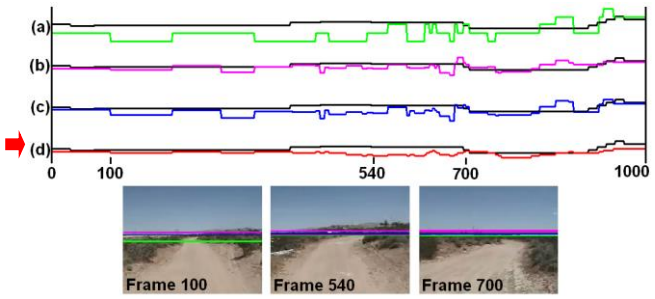


Figure 7. Desert road videos available by DARPA [10], [19].

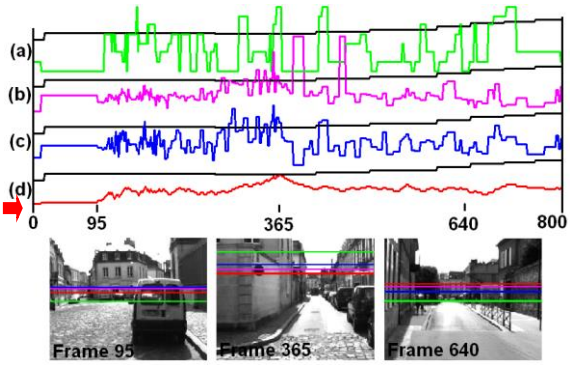


Figure 8. Real environment in France: data obtained by our intelligent vehicle.

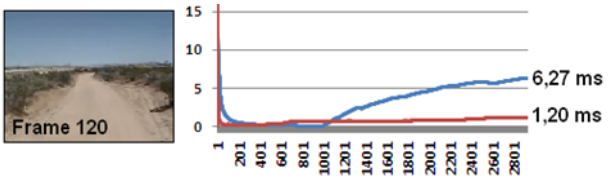


Figure 9. The computational mean time in desert roads.

## B. *Estimation of Drivable Image Area (Road Detection)*

The vision-based sensors are defined as passive sensors and it can be used for some specific applications such as: road marking localization, traffic signs recognition and obstacle identification. To investigate the information's road, different techniques on automatic and semi-automatic road extraction methods are proposed in the literature [23], [24], [25], [26].

Aware that in the majority of the navigation systems, the machine vision system is working together with others low cost sensors, we have presented a monocular vision-based system that includes a robust road detection algorithm [27]. For this task, we apply the sky removal which we have proposed in [11], accelerating the process of identifying objects ahead of the vehicle because once using a global segmentation method, not always the analysis of a great portion of the image can contribute for a better result in the most critical region (region closer to the vehicle) where obstacles should be detected and avoided as fast as possible. On the contrary, when discarding the superior portion of the original image, we are capable to get a more efficient segmentation and to distinguish with higher precision the obstacles from the navigable area.

For finding the drivable surfaces, [28] projects drivable area from the laser scan analysis into the camera image. This quadrilateral area is between 10 and 20 meters ahead of the robot. In this real-time approach, the basic idea is to consider a given region in the actual image as drivable. It assumes that the bottom center of the image contains road pixels for a large majority of the time [29], as we present in Fig. 10 (a): yellow region. This technique was first presented by [24].


Figure 10. (a) Original image after sky removal and its Otsu's result in (b); (c) negated image and its Otsu's result in (d); (e) Canny edge detection result; (f) Hough transform result; (g) a single mass from the image (f).

In our proposed method, if the image is colored, in order to utilize the most important information of the color image, the candidate color channel that was dominant in the bottom center of the image is selected to generate the histogram image. A multimodal road image in (4) is then triggered based on the weighted average of the images intensities. In order to obtain a multimodal 2D drivability free-area, the algorithm performs the following tasks:

**Task 1:** Due to the different image textures on different roads, the original image, Fig. 10 (a), and its negated, Fig. 10 (c), are submitted to the Otsu's method. The algorithm then selects an image with the highest percentage of navigable area (white points) in the bottom center of the image, as shown in Fig. 10 (a): yellow area. The original image and its Otsu's result, $Os^{+}$, can be seen in the Fig. 10 (a) and (b). The negated image and its Otsu's result, $Os^{-}$, can be seen in the Fig. 10 (c) and (d).

**Task 2:** Whereas there are homogeneous regions in the image, and in order to identify the limits of the road (which includes the obstacles), the Canny edge detector was employed as input of Hough transform [22] due to its robust performance and accurate edge localization. Respectively, the results can be seen in the Fig. 10 (e) and (f). Then, from the bottom center of the image, the algorithm concludes by finding a single image mass, $Hc$, Fig. 10 (g). It may also help to identify the textureless regions classified as road region, specular surfaces, traffic markings, etc.

$$FA_{w(x,y)} = \frac{FA_{w(x,y)} + Os_{(x,y)} + Hc_{(x,y)}}{3} \qquad (4)$$

where $FA_{w(x,y)}$ (left) is the intensity of the $i^{th}$ pixel after update in the new image $FA_{w}$, $FA_{w(x,y)}$ (right) is the intensity of the $i^{th}$ pixel in the old image $FA_{w(x,y)}$, $Os_{(x,y)}$ is the intensity

of the $i^{th}$ pixel in image obtained in Task 1, $Hc_{(x,y)}$ is the intensity of the $i^{th}$ pixel in image obtained in Task 2.

Due the small variance of shades between objects, false path-markings, false route-markings, shadows, etc, the next step is applied to re-project this 2D drivability free-navigable area by considering a drag process which we presented in [27]. In fluid dynamics, drag forces act in a direction opposite to the oncoming flow velocity, i.e. forces (or resistance) that oppose the relative motion of an object through a fluid [30]. Right after applying our drag process, an example is shown in Fig. 11.


Figure 11. (a) Original image after sky removal and its Otsu's result in (c); (b) Canny edge detection result; (d) A multimodal 2D drivability free-area by considering the drag process.

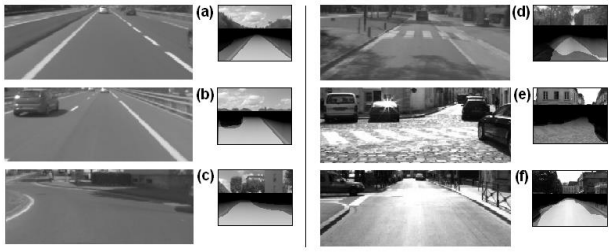


Figure 12. Real environment in France: urban and real experimental test-bank.

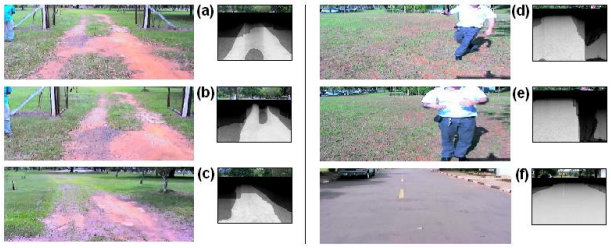


Figure 13. Real environment in Brazil: real-time autonomous displacement.

At first stage of testing, in order to evaluate the proposed algorithm performance, we used an urban and real experimental test-bank obtained using the vehicle shown in Fig. 15 (a). A typical urban environment was selected and its results are presented in Fig. 11. All false-navigable areas (red) in Fig. 11 (c) are eliminated by applying the drag process resulting in Fig. 11 (d). Additionally, different types of image texture (road surfaces) were selected and its results are presented in Fig. 12 (a) to (f). A result for a shadow context is presented in Fig. 12 (d). An occlusion case (vehicle) is shown in Fig. 12 (b).

At second stage of testing, in order to evaluate the proposed algorithm performance to autonomous displacement, the experiments on real-time conditions were performed using the vehicle VERO shown in Fig. 15 (b). Fig. 13 (a), (b) and (c) show the successful task execution to go through a gate in off-road context. Fig. 13 (d) and (e) present the obstacle detection and our open-loop reactive navigation. In all experiments there was no collision. Additionally, a different type of image texture (road surfaces) was also selected and its result is presented in Fig. 13 (f).

## IV. Autonomous Navigation: Reactive Perception

Right after getting a 2D drivable road image in (4), we have two types of classes: $c_1$ and $c_2$. The center of area is:

$$m_c = \sum_{x,y} h_c(x,y) \tag{5}$$

where $h_c(x,y) = \begin{cases} 1.se.S_{x,y} = c_1 \\ 0.se.S_{x,y} \neq c_1 \end{cases}$. The class $c_1$ (drivable area), denoted as class $o_{c1}$, is given by $xo_{c1}$ and $yo_{c1}$:

$$xo_{c1} = \frac{\sum_{x,y} x.h_c(x,y)}{m_c} \text{ and } yo_{c1} = \frac{\sum_{x,y} y.h_c(x,y)}{m_c} \tag{6}$$

Taking the origin point as:

$$x = \frac{image.width}{2}, y = image.height \tag{7}$$

and getting a destination point in (6), the steering angle correction (new direction) can be calculated in (8).

$$a = ((arc.\cos(\frac{|(yo_{c1} - y)|}{\sqrt{(xo_{c1} - x)^2 + (yo_{c1} - y)^2}}))*180)/PI \tag{8}$$

The final step is applied the sensor-based control [31] in the sensor space, where the robot's controller regulates to zero the error function of the reference trajectory. An example is presented in Fig. 14. The main objective is to drive the robot to the center of the drivable area.

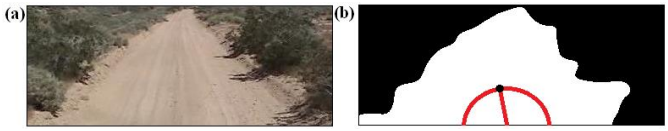


Figure 14. The center of area of the drivable area: the new robot direction.

## V. Conclusion

A real-time perception problem is applied to autonomous robots. From an image captured by a single camera, the purpose was to present a real-time machine vision algorithm capable of estimating mainly the drivable area and the risk of collision.

The experiments showed that the inclusion of an automatic image discarding method based on PCC did result in a reduction of the processing time. This technique is also presented as an environment observer method and futures work will provide a real experimental test-bank to evaluate

the real energy consumption economy in terms of electrical current used by the visual machine.

A remarkable characteristic of all methodologies presented here is its independence of the image acquiring system and of the robot itself. For the long range navigation of intelligent electric vehicles, this could represent a considerable gain in the vehicle autonomy and will be studied for future implementations and in order to validate the proposed tools, future work would be also focused to provide ground truth measurements.

Besides the experimental DARPA test-banks, the results here were obtained using experimental vehicles on real, dynamic and unknown environments. Videos showing the application of these methods are available in [32] which include: the use of Karlsruhe Institute of Technology database; ROI Selection [33]; Collision Risk Estimation [34]; Sky Removal [35]; and Road Detection [36].

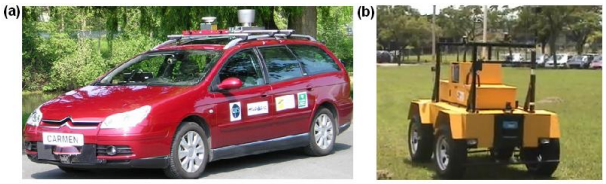


Figure 15. The experimental vehicles: (a) Carmen vehicle at Heudiasyc Laboratory in Compiègne, France; (b) Autonomous vehicle (VERO) at Renato Archer IT Center (CTI) in Campinas, Brazil.

## ACKNOWLEDGMENT

The authors wish to thank Luiz Mirisola, Samuel Bueno, Josué Ramos, Hélio Azevedo and Gerald Dherbomez for their kind attention to this work and support in data acquisition. This work has been cofounded by ANR, FAPESP, CNPq and CAPES.

## REFERENCES

[1] M. Bertozzi. A. Broggi and A. Fascioli, "Vision-based intelligent vehicles: state of the art and perspectives". Robotics and Autonomous systems 32, 1–16, 2000.

[2] F. Diego, J. M. Álvarez, J. Serrat and A. M. López, "Vision based Road Detection via On line Video Registration", Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2010.

[3] K. Yamaguchi, A. Watanabe, T. Naito and Y. Ninomiya, "Road region estimation using a sequence of monocular images", Proc. of the Int. Conf. on Pattern Recognition (ICPR), 2008.

[4] Radio Spectrum Committee, European Commission, Public Document, Brussels, 5 July 2010, RSCOM10-35 http://ec.europa.eu/information_society/policy/ecomm/radio_spectrum/_document_storage/rsc/rsc32_public_docs/rscom10_35.pdf [retrieved 02 Dec. 2010].

[5] H. Dahlkamp et al., "Self-Supervised Monocular Road Detection in Desert Terrain", Proc. of the Robotics Science and Systems Conference, 2006.

[6] K. Pearson, (1895), Royal Society Proceedings, 58, 241.

[7] A. Miranda Neto, L. Rittner, N, Leite, D. E. Zampieri and A. C. Victorino, "Nondeterministic Criteria to Discard Redundant Information in Real Time Autonomous Navigation Systems based on Monocular Vision", ISIC Invited Paper, Proc. of the IEEE Multi-conference on Systems and Control (MSC), 2008.

[8] H. Y. L. C. Yongguo Mei, Yung-Hsiang Lu, "A case study of mobile robot's energy consumption and conservation techniques", Proc. of the IEEE Int. Conf. on Advanced Robotics (ICAR), 2005.

[9] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Real-Time Dynamic Power Management based on Pearson's Correlation Coefficient", Proc. of the IEEE Int. Conference on Advanced Robotics (ICAR), 2011.

[10] DARPA 2005. "DARPA Grand Challenge", http://www.darpa.mil/grandchallenge05/ [June 10th, 2006]

[11] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Robust Horizon Finding Algorithm for Real Time Autonomous Navigation based on Monocular Vision", Proc. of the IEEE Int. Conference on Intelligent Transportation Systems (ITSC), 2011.

[12] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Automatic Regions-of-Interest Selection based on Pearson's Correlation Coefficient", Proc. of the IEEE Int. Conference on Intelligent Robots and Systems IROS/ViCoMoR, 2011.

[13] N. Otsu, (1978), "A threshold selection method from gray-level histogram". IEEE Transactions on Systems, Man, and Cybernetics.

[14] A. Miranda Neto, "Embedded Visual Perception System applied to Safe Navigation of Vehicles", PhD Thesis, UNICAMP-Brazil/UTC-France, 2011.

[15] O. J. Gietelink, J. Ploeg, B. Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-theloop simulations". Mechatronics, 19:1091–1104, 2009.

[16] D. Müller, J. Pauli, C. Nunn, S. Görmer, S. Müller-Schneiders, "Time To Contact Estimation Using Interest Points", Proc. of the IEEE Int. Conference on Intelligent Transportation Systems (ITSC), 2009.

[17] A. Negre, C. Braillon, J. Crowley and C. Laugier, "Real-time Time-To-Collision from variation of Intrinsic Scale", Proc. of the Int. Symp. on Experimental Robotics, 2006.

[18] A. Miranda Neto, A. C. Victorino, I. Fantoni and J. V. Ferreira, "Real-time Collision Risk Estimation based on Pearson's Correlation Coefficient", Proc. of the IEEE Workshop on Robot Vision (WORV), 2013.

[19] Stanford Racing Team's Entry In The 2005 DARPA Grand Challenge, http://www.stanfordracing.org/ [June 10, 2006]

[20] King Hann Lim et al., "Vision-based Lane-Vehicle Detection and Tracking", chapter 13 of IAENG Transactions on Engineering Technologies Vol. 3, pp. 157-171, 2009.

[21] S. Ettinger et al., "Vision-Guided Flight Stability and Control for Micro Air Vehicles", Advanced Robotics, Vol. 17: 617-640, 2009.

[22] D. Ballard, "Generalized Hough transform to detect arbitrary shapes", IEEE Trans. Pattern Anal. Machine Intell. 13 (2), 111–122, 1981.

[23] G. Aviña-Cervantes, M. Devy and A. Marín, "Lane Extraction and Tracking for Robot Navigation in Agricultural Applications", Proc. of the IEEE Int. Conference on Advanced Robotics (ICAR), 2003.

[24] I. Ulrich and I. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision", Proc. of the AAAI National Conference on Artificial Intelligence, 866-871, 2000.

[25] F. Diego, J. M. Álvarez, J. Serrat and A. M. López, "Vision based Road Detection via On line Video Registration", Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC), 2010.

[26] J. Chetan, K. Madhava and C. V. Jawahar, "An Adaptive Outdoor Terrain Classification Methodology using Monocular Camera", Proc. of the IEEE Int. Conference on Intelligent Robots and Systems (IROS), 2010.

[27] A. Miranda Neto, A. C. Victorino, I. Fantoni and J. V. Ferreira, "Real-Time Estimation of Drivable Image Area based on Monocular Vision", Proc. of the IEEE Intelligent Vehicles Symposium (IV), 2013.

[28] S. Thrun, et al. "Stanley, the robot that won the DARPA Grand Challenge", Journal of Robotic Systems, Volume 23, Issue 9, 661-692, 2006.

[29] F. W. Rauskolb, et al.,"Caroline: An autonomously driving vehicle for urban environments", Journal of Field Robotics, 25(9):674–724, 2008.

[30] F. M. White, "Fluid Mechanics', 2nd Ed. McGraw Hill, 1986.

[31] A. C. Victorino, P. Rives, and J. -J., "Safe navigation for Indoor Mobile Robots Part I: A Sensor-based Navigation Framework", Part II: Exploration, Self Localization and Map Building", International Journal of Robotics Research, v22, n12, 2003.

[32] http://www.youtube.com/user/kingdombr/videos[Jan., 21th 2014]

[33] http://www.youtube.com/watch?v=VcUQVC1F8Xw[Jan., 21th 2014]

[34] http://youtu.be/J8YuZlJFExk [Jan., 21th 2014]

[35] http://www.youtube.com/watch?v=8KbZ1J0txUE [Jan., 21th 2014]

[36] http://youtu.be/ZpEbRo32pY8 [Jan., 21th 2014]