Tekhnê
January - June 2017, Vol. 14, No. 1, pp. 73 – 86

ISSN 1692-8407

Adjustment of visual identification algorithm for use in stand-alone robot navigation applications

Ajuste de algoritmo de identificación visual para uso en aplicaciones de navegación autónoma de robots

Jordan S. Castañeda B.
Universidad Distrital Francisco José de Caldas
jordanstv@hotmail.com

Yeison A. Salguero L.
Universidad Distrital Francisco José de Caldas
yasalguerol@correo.udistrital.edu.co

This article describes the adjustment of an algorithm developed by the research group for the visual identification of geometric shapes in real time on embedded systems. This algorithm was proposed inside the group, and in spite of its correct theoretical support, it presented problems when working in real applications. The purpose of the adjustment was to increase the robustness against different levels of lighting and implement it on a real robot for performance tests in the laboratory. The tasks programmed in the robot include the identification of landmarks in the environment for the activation of navigation policies. From the results achieved, a better behavior of the algorithm is observed, making its use very promising.

Keywords: Embedded system, geometric shapes, real time, visual identification

Este artículo describe el ajuste de un algoritmo desarrollado por el grupo de investigación para la identificación visual de formas geométricas en tiempo real sobre sistemas embebidos. Dicho algoritmo fue propuesto al interior del grupo, y pese a su correcto soporte teórico, presentó problemas al funcionar en aplicaciones reales. En el ajuste realizado se buscó incrementar la robustez frente a diferentes niveles de iluminación, e implementarlo sobre un robot real para pruebas de desempeño en laboratorio. Las tareas programadas en el robot incluyen la identificación de *landmarks* en el ambiente para la activación de políticas de navegación. A partir de los resultados alcanzados se observa un mejor comportamiento del algoritmo haciendo muy promisoria su utilización.

Palabras clave: Formas geométricas, identificación visual, sistema embebido, tiempo real

Article typology: Research

Date manuscript received: May 26, 2017
Date manuscript acceptance: June 30, 2017

Research funded by: Universidad Distrital Francisco José de Caldas.

Digital edition: http://revistas.udistrital.edu.co/ojs/index.php/tekhne/issue/view/798
How to cite: Castañeda, J., Salguero, Y. (2017). *Adjustment of visual identification algorithm for use in stand-alone robot navigation applications*. Tekhnê, 14(1), 73 -86.

Introduction

Nowadays robots have become a tool capable of carrying out tasks efficiently and with greater precision than human beings, in such a way that today the need arises to implement increasingly autonomous and sophisticated robots that are capable of simplifying human work, thus achieving greater efficiency in the development of any process that is intervened by such robots.

One of the points of greatest interest is the detection of obstacles in unmanned vehicles (Samadi & Othman, 2013) since they make use of the omnidirectional system to have a better vision in mobile robots (Lee, Feng, Yeh, & Ivan, 2013). In addition the use of artificial vision makes a great contribution to this area of research where the IBDMS method (Tsai, Chuang, Lu, & Wang, 2013), helps us to determine the collision distance and the early reaction of the machine by means of a camera, as it helps us to recognition based on CAD (Louis & Ricky, 2011), all this is done with the need to maneuver in places with objects and obstacles (Samadi & Othman, 2013), the results that are obtained are very useful because they get it without using GPS, and is very well received because its implementation is in unmanned aerial vehicles (Samadi & Othman, 2013).

In addition, the approach to autonomy is very large, as it is a fundamental part of robots (Jacinto, Martínez, & Martínez, 2016). Recently an air vehicle (MAV) (Schoellig, Barfoot, & Pfrunder, 2014) is being developed, with 3D vision sensors, helps to develop a great variety of tasks without the use of GPS, this autonomous flight system presents great success in its tasks of path planning and exact positioning.

Also, the systematic development of an unmanned helicopter (Ponte, Queenan, Gong, & Mertz, 2014), makes its contribution in the detection of objects, using the hardware as vision system and being able to coordinate several tasks, giving an immediate response to the terrestrial objective in movement.

In the utility of artificial vision is necessary to make applications capable of reading data in both static and real-time images, in order to detect shapes or colors for tasks of image quality improvement or restoration of degraded images, as well as to have simple and efficient results according to the interest sought.

For the identification of figures (Aslan, Abdelmunim, & Farang, 2011), it proposes a geometric and dynamic method based on the search of forms in some sets of images. The algorithm joins the different shapes contained in these sets to make segmentation of the image. The image is transformed into a two-dimensional space, and an analysis of the main components is carried out to represent the variation of the shape and to calculate a sufficient number of possible projections in the directions that an object can have and thus carry out its detection.

Image detection obtains geometric shapes and external features of the target (Chegtian, Keyong, & Lian, 2008) in a very short distance and provides relevant information about the target. But due to a large amount of information provided in real time by the camera, sometimes there is some noise or distortion of the lens. To solve this problem, a laser indicator is used to mark the region of interest. The algorithm binarizes the image to obtain and orient the skeleton. Secondly, the shape of the lens is extracted. Finally, a mapping is performed using the WMF (Wave Mapping Feature) technique to obtain the state of the target.

In binary images, the pixel can take exactly one of two values (Burger & Burge, 2008). It is often thought that these values represent the foreground and background in the image, although these concepts are often not applicable to natural scenes. Thus we focus on regions in images and how to isolate and describe such structures. Our main task, then, is to design a program to interpret the number and type of objects in a figure. As long as we continue to consider each pixel in isolation, we will not be able to determine how many objects there are in general in the image, where they are, and whose pixels belong to which objects. Therefore, our first step is to find each object by grouping all the pixels that belong to it. In the simplest case, an object is a group of pixels taken from the foreground; that is, a connected binary region.

Algorithm development is central to image processing and computer vision (MathWorks, n.d.), as each situation is unique, and good solutions require multiple design iterations. MathWorks provides a complete environment to delve into image and video data, develop algorithms, and explore the advantages and disadvantages of implementation. These tools become essential when seeking the development of autonomous robotic applications (Martínez, Jacinto, & Zárate, 2015), particularly when used in research and education (Martínez, Montiel, & Jacinto, 2016).

A digital image is nothing more than data numbers indicating variations of red, green, and blue in a particular location in a grid of pixels (Processing, n.d.). Most of the time, we see these pixels as miniature rectangles interspersed together on a computer screen. With a little creative thinking and a minimum level of coding for pixel manipulation, we can display that information in several ways.

Problem formulation

For the adjustment of these identification algorithms, we start from a functional analysis and performance tests to the two codes previously developed by the members of the research group Jhon León and Henry Valderrama (León & Valderrama, 2016). With these algorithms, they showed results with three types of luminosity on the target, for which they reported the results of Figs. 1 to 11.

ISSN 1692-8407

Figure 1. Low light intensity. First test performed on code 1 (León & Valderrama, 2016).

Figure 2. Low light intensity. Second test performed on code 1 (León & Valderrama, 2016).

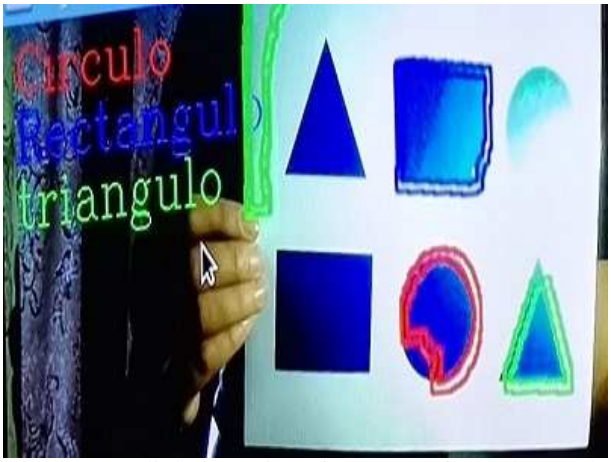

Figure 3. Low light intensity. First test performed on code 2 (León & Valderrama, 2016).

Figure 4. Low light intensity. Second test performed on code 2 (León & Valderrama, 2016).

Figure 5. Medium light intensity. Test performed on code 2 (León & Valderrama, 2016).

Figure 6. Medium light intensity. First test performed on code 2 (León & Valderrama, 2016).

Figure 7. Medium light intensity. Second test performed on code 2 (León & Valderrama, 2016).

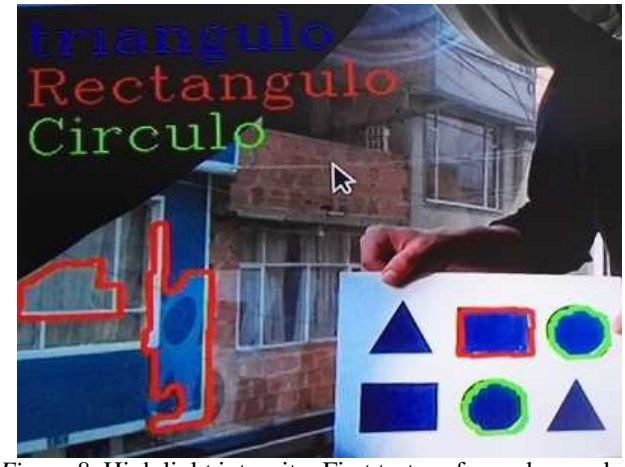

Figure 8. High light intensity. First test performed on code 1 (León & Valderrama, 2016).

Figure 9. High light intensity. Second test performed on code 1 (León & Valderrama, 2016).

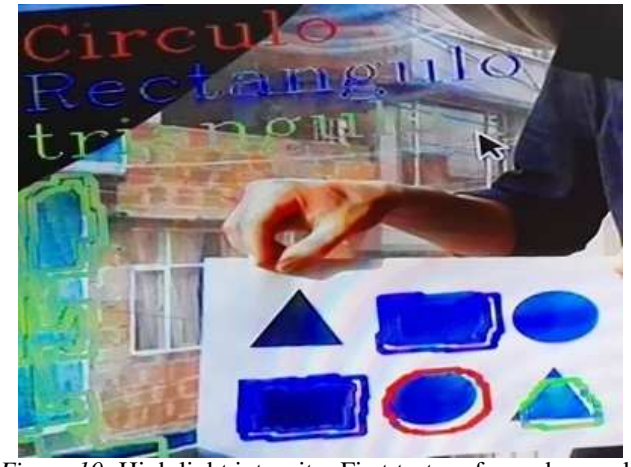

Figure 10. High light intensity. First test performed on code 2 (León & Valderrama, 2016).

Figure 11. High light intensity. Second test performed on code 2 (León & Valderrama, 2016).

In order to carry out the above tests, the luminous intensity of the environment was taken into account. The lighting conditions were checked in each case with the aid of measuring instruments (luxmeter). Table 1 summarises the test results.

Table 1
Laboratory test results with initial algorithms.

Environmental characteristics	Luminous intensity [lx]
Low light intensity	50
Medium light intensity	370
High light intensity	730

As a final result of the performance of the two previous algorithms it was determined that the amount of light was an important factor in the recognition of the figures, since the camera exposed in this work presented a low resolution, which prevented a better response. As can be seen in the

ISSN 1692-8407

images taken from the tests, the best environmental condition for the recognition of these figures involves a high amount of light, and with the help of the luxometer was defined a luminous intensity of 730 lx.

As work projection from the results, we started adjustments with the first algorithm, since it presents a better recognition and detection of the figure, and also allows us to make a better redesign and implementation of the codes. It should also be noted that the camera for such work must present a better resolution and field of vision, in order to make a better recognition in any environment.

Methodology

The processing system was assembled in order to have a low-cost platform to perform the tasks of recognition and visual representation of geometric figures. The parts that compose it are (Figs. 12 and 12):

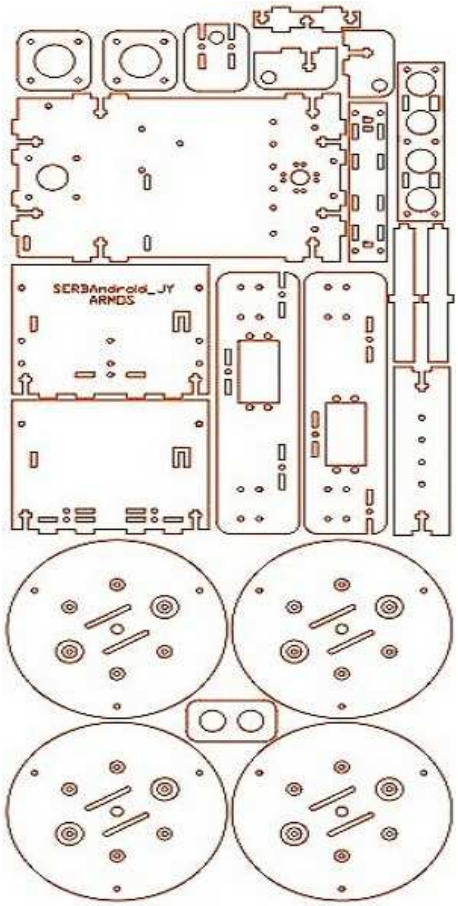

Figure 12. AutoCAD template of the robot's mechanical structure.

Figs. 14 to 30 show the assembly steps of the SERB robot.

Figure 13. Parts of the robot's mechanical structure cut from acrylic sheet.

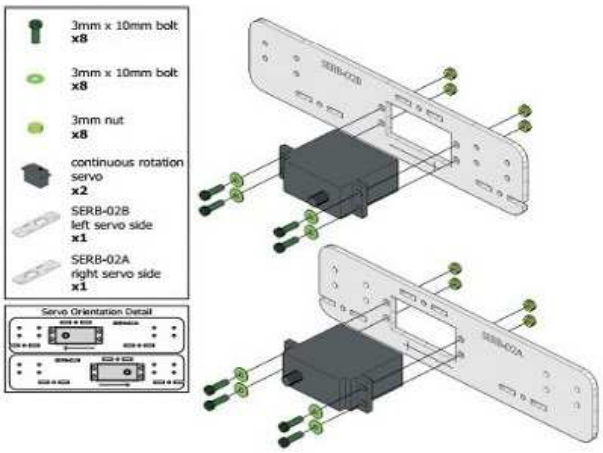

Figure 14. Assembly of servomotors Martínez, Montiel, and Valderrama (2016).

Embedded system

For the direct control of actuators and sensors, we have selected a small 8-bit microcontroller on an

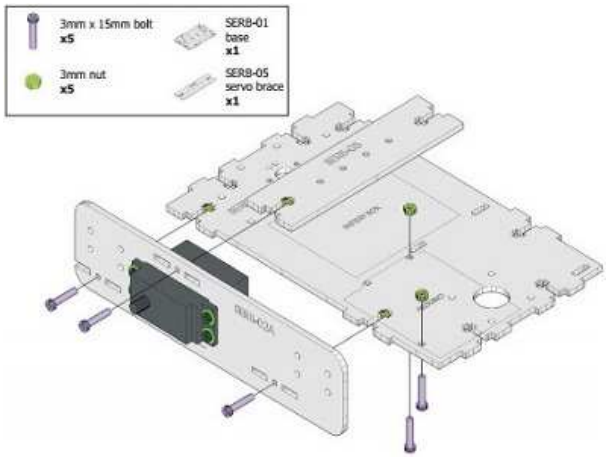

Figure 15. Protoboard mounting bracket assembly Martínez, Montiel, and Valderrama (2016).

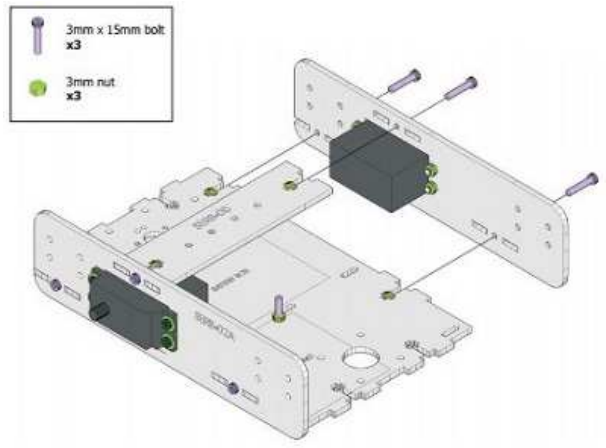

Figure 16. Main base assembly Martínez, Montiel, and Valderrama (2016).

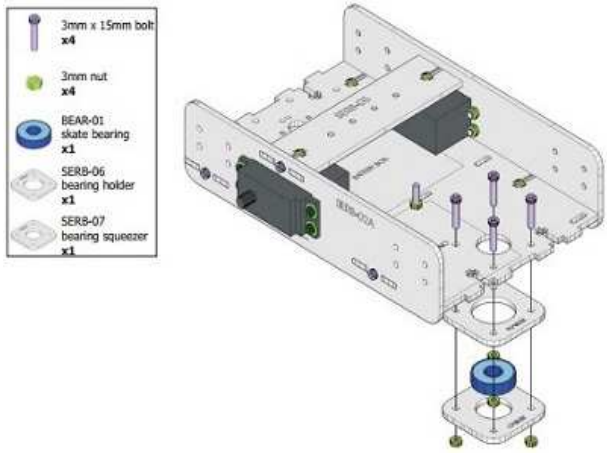

Figure 17. Support wheel base assembly Martínez, Montiel, and Valderrama (2016).

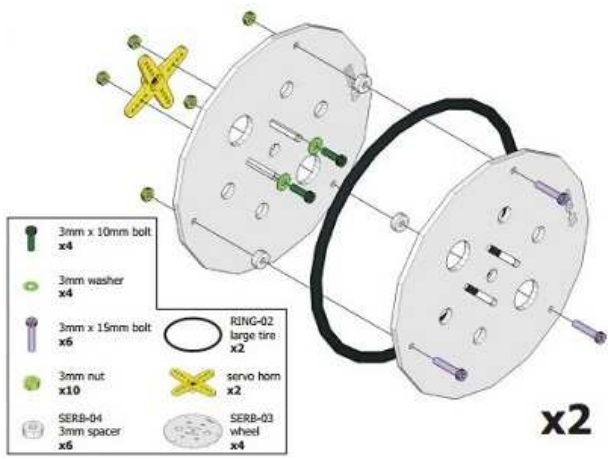

Figure 18. Wheel assembly Martínez, Montiel, and Valderrama (2016).

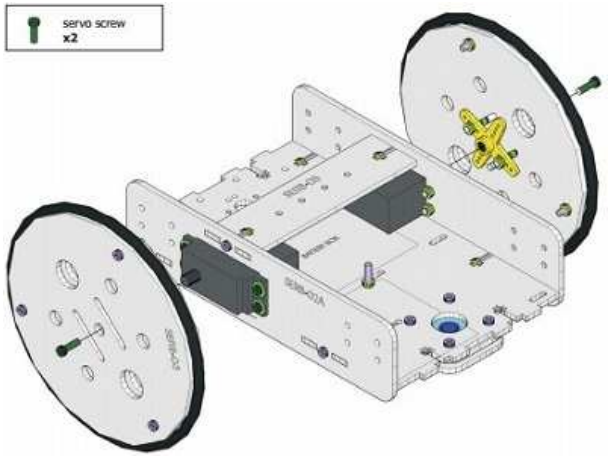

Figure 19. Assembly of the wheels on the main base Martínez, Montiel, and Valderrama (2016).

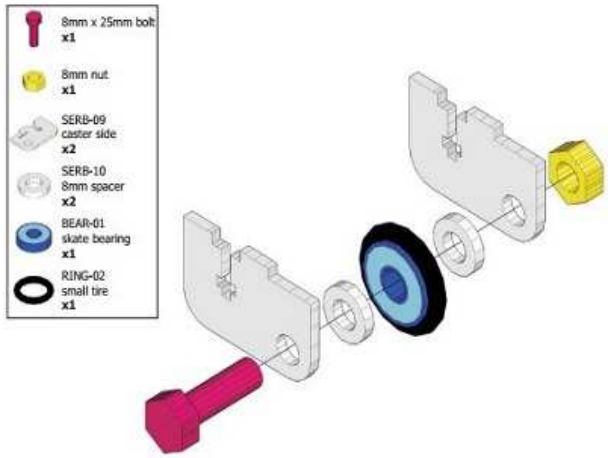

Figure 20. Support wheel assembly Martínez, Montiel, and Valderrama (2016).

ISSN 1692-8407

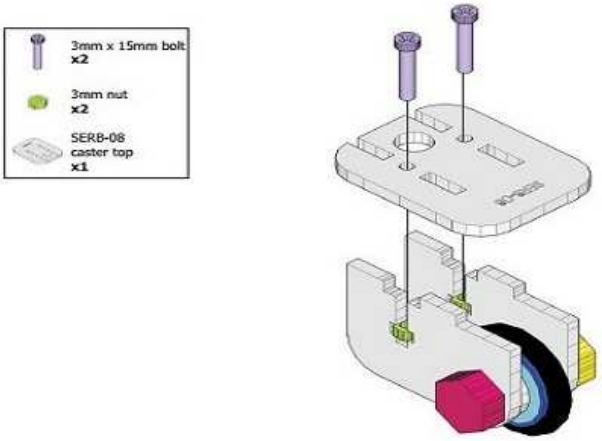

Figure 21. Support wheel bracket assembly Martínez, Montiel, and Valderrama (2016).

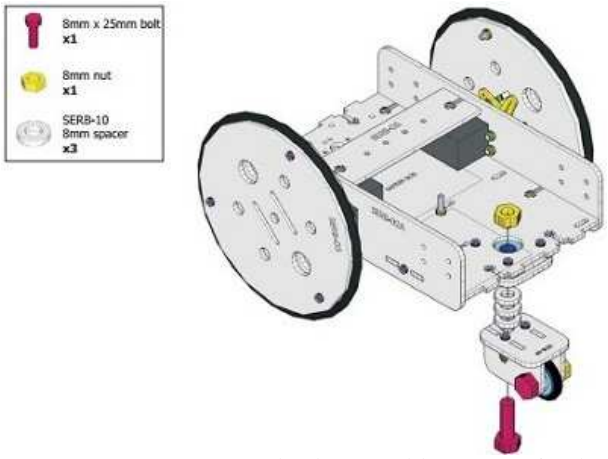

Figure 22. Support wheel assembly on main base Martínez, Montiel, and Valderrama (2016).

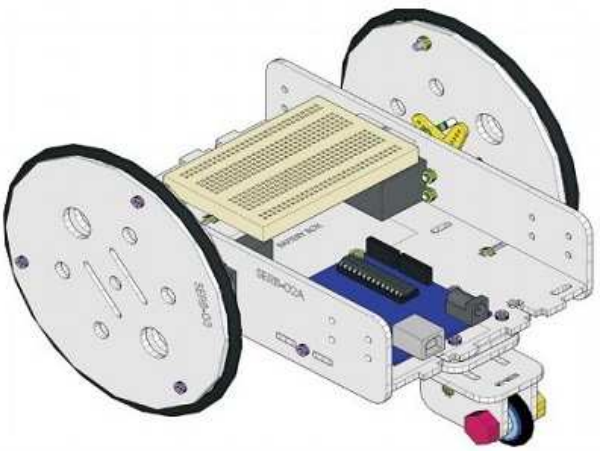

Figure 23. Scheme of the fully assembled SERB robot Martínez, Montiel, and Valderrama (2016).

Figure 24. Real SERB Robot assembled.

Figure 25. Front face of the support structure for the smartphone.

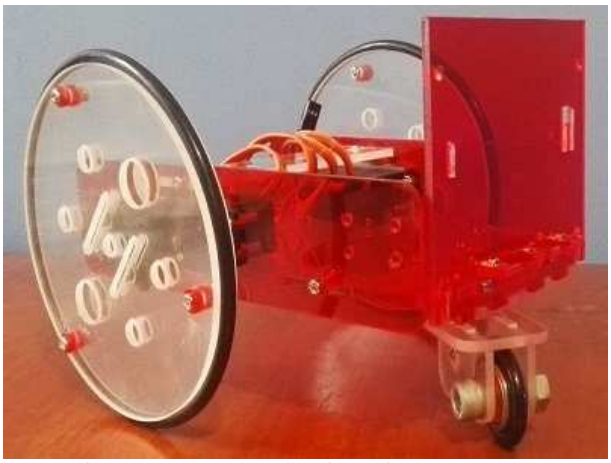

Figure 26. Smartphone bracket front face assembly.

Arduino/Genuino board. The system consists of an ATmega328P microcontroller. It has 14 digital input/output pins (6 of which can be used as PWM outputs), 6 analog

Figure 27. Rear face of the support structure for the smartphone.

Figure 28. Smartphone bracket rear face assembly.

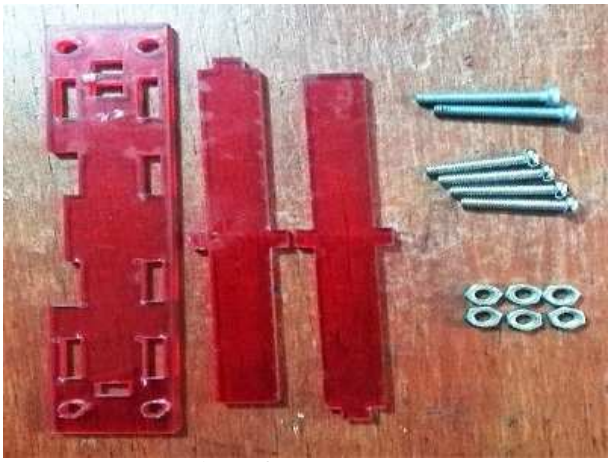

Figure 29. Front and rear face support.

inputs, a 16 MHz crystal, a USB connection, a power connector, an ICSP header, and a reset button. This is programmed by means of its own IDE, in a language that

Figure 30. SERB robot fully assembled with smartphone holder.

is a variant of C/C++. It is linked to AVR Libc, a library that allows the use of any of the functions of the microcontroller (Arduino, n.d.).

Android device with IP camera

With the advancement of technologies today, the IP camera has become a very functional tool in projects where it involves the use of it, has also been developed to be a very simple task to implement, and has great coverage in low-end devices as it is compatible with versions of Android 1.6 onwards.

The application must first be downloaded from the Play Store located at the following link: `https://play.google.com/store/apps/details?id=com.pas.webcam&hl=en_419`. It is then necessary that the device and the computer are connected to the same WI-FI network. You can see that the application has a variety of configurations among the most important this allow choosing the video resolution, image quality, among others. All this depends on the need for which it will be used, once this is implemented, only choose the option to start the server, which generates an IP address of the type `http://192.168.0.#:8080/`, which must be implemented in the Matlab code to be able to make the image reproduction in real time.

Engineering of image processing algorithms

For the realization of the image processing algorithm for the recognition of basic geometric figures, we select as target figures the triangle, circle, rectangle, and square. This algorithm has been rewritten under the Matlab environment (the original code is written in Python) and was adjusted so that it only detects blue figures, regardless of the background

ISSN 1692-8407

texture. In this way, better results are obtained in the identification of the figure.

The figure detection process consists of several stages. First, the algorithm finds a rectangular bounding box for each region. The bounding rectangle can be oriented with an arbitrary angle using the `minbounrect` function (Matlab toolbox) and will take the following approaches for each geometric shape (Fig. 31).

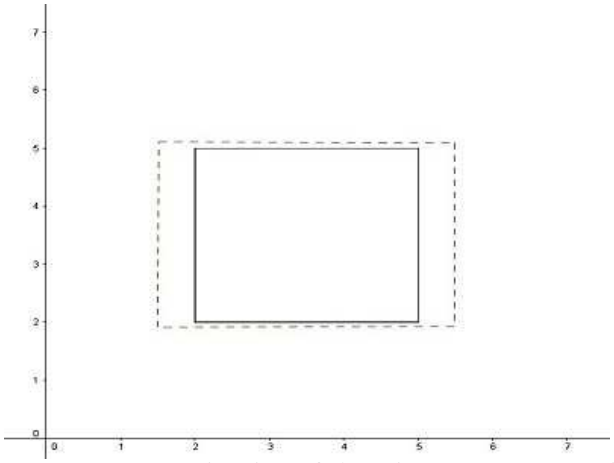

Figure 31. Approximation of detection square geometric shape.

The square detection takes into account the aspect ratio between width and height of the rectangular box found thanks to the `minboundrect` function (Matlab toolbox). The shape of the square is almost proportional to the shape of the rectangular box, therefore it is necessary to identify the aspect ratio width/height with a value below the unit. On the other hand, when compared with the other geometric shapes, it will be close to the unit by approximately 80%.

For the detection of the triangle, the area of the rectangular box found is taken into account, as well as the area occupied by the triangular-shaped region. Therefore, the metric relation of the triangle is assigned as (area of the region)/(area of the rectangular box). The space occupied by this form shall not exceed 60% with respect to the space occupied by the area of the rectangular box. Thus we take as values lower than 60% the geometric shape of the triangle, which in turn will serve to be compared with the other geometric shapes (Fig. 32).

The detection of the circle takes into account the metric relation of the perimeter of the circle for the regions found. This ratio produces numerical values greater than one, so it is greater than the width/height ratio of the rectangular box found and in turn the area occupied by it. This will differentiate it from other geometric shapes (Fig. 33).

For the identification of the rectangle is taken into account its closeness according to the aspect ratio of height/width of the rectangular box found, so the proportional value will be very close to the unit (Fig. 34).

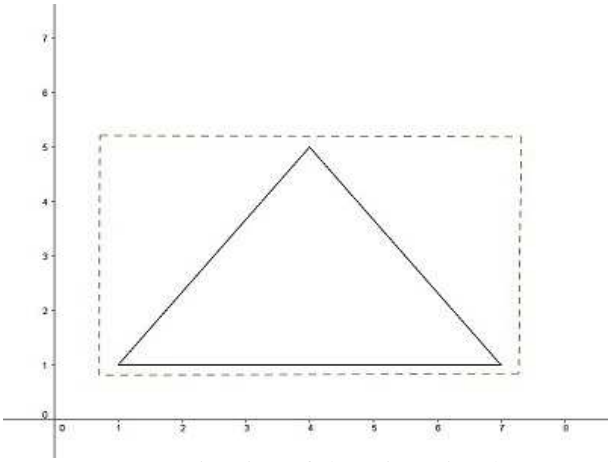

Figure 32. Approximation of detection triangle geometric shape.

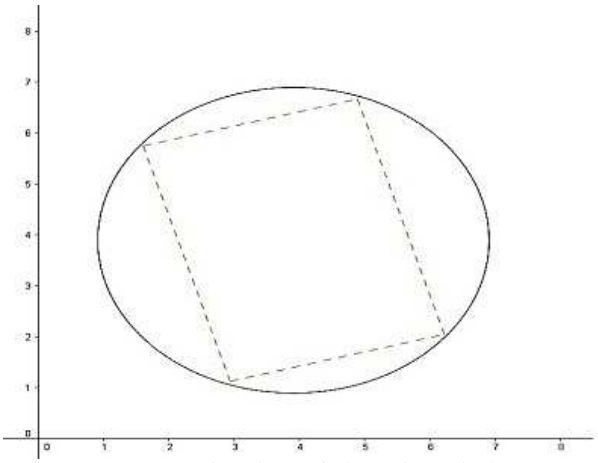

Figure 33. Approximation of detection circle geometric shape.

Deployment

The operation of the algorithm is shown in the flow diagram in Fig. 35. The figures shown in Fig. 36 were used for the laboratory tests.

Figs. 37 to 42 show the result of the step-by-step processes applied to the image in Fig. 36.

The flow diagram in Fig. 43 correspond to the final process of labeling regions and determining the geometric figure.

According to the results of the tests carried out on the original algorithm, and in addition to what was stated in the initial work (León & Valderrama, 2016), we assume the following elements as the basis for the adjustment of the

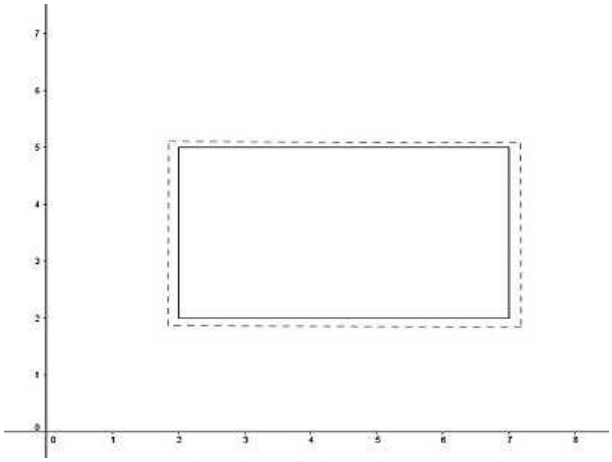

Figure 34. Approximation of detection rectangle geometric shape.

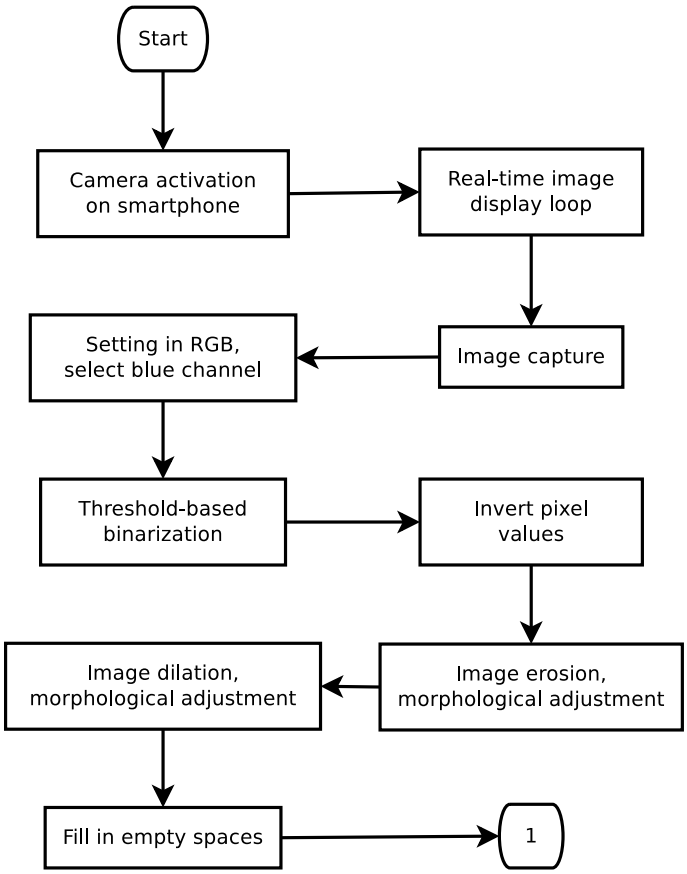

Figure 35. Flowchart of the geometric figure identification algorithm.

Figure 36. Geometric test figures.

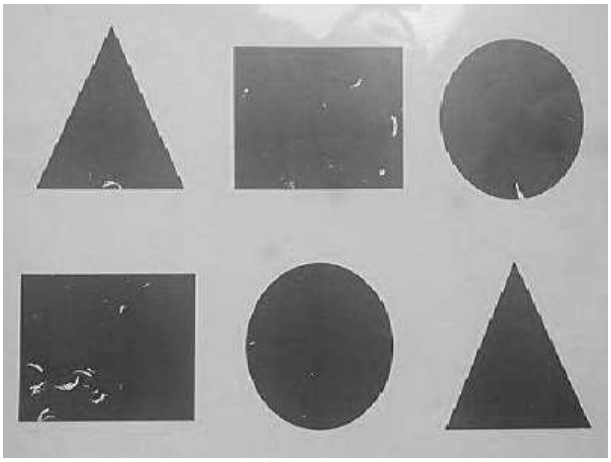

Figure 37. Blue channel of the image in RGB format.

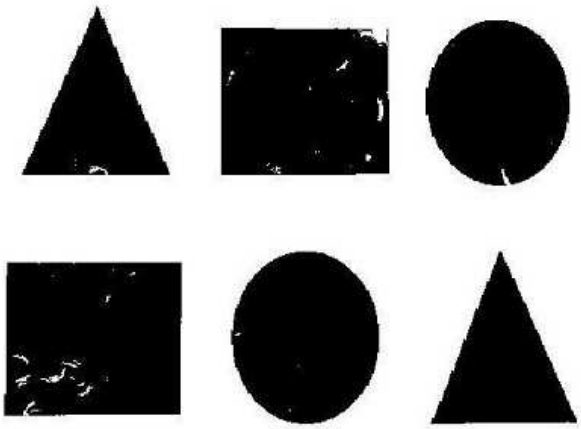

Figure 38. Binarized image.

algorithm in order to improve the detection of geometric figures.

One of the aspects attacked for the improvement was the device used for the process of capture of the real environment, for the case of Leon and Valderrama (León & Valderrama, 2016), was used the Raspberry Pi Model B+ board, offering limitations in its peripheral device of visualization, its 5 Megapixel camera. For our case, we

ISSN 1692-8407

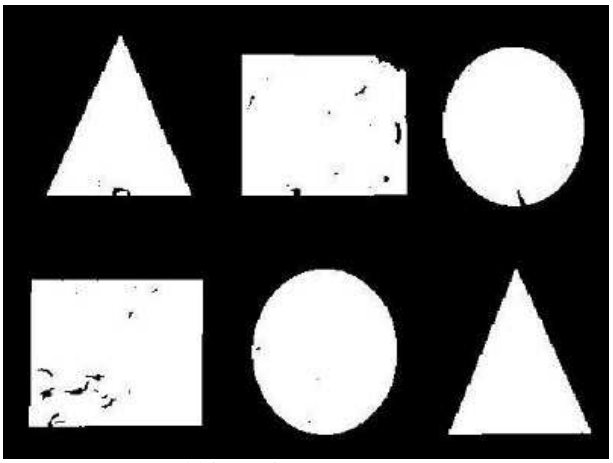

Figure 39. Inverted binary image.

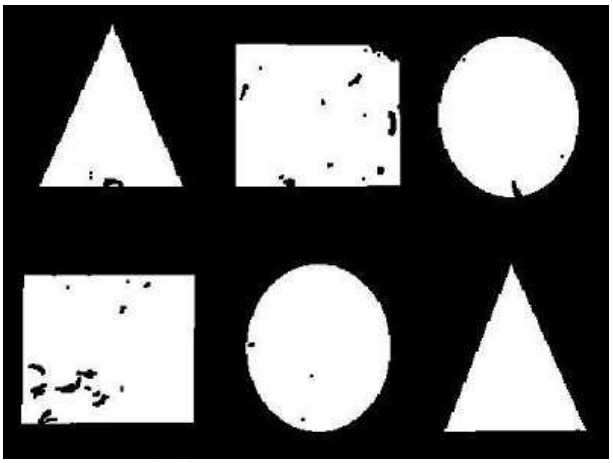

Figure 40. Eroded image.

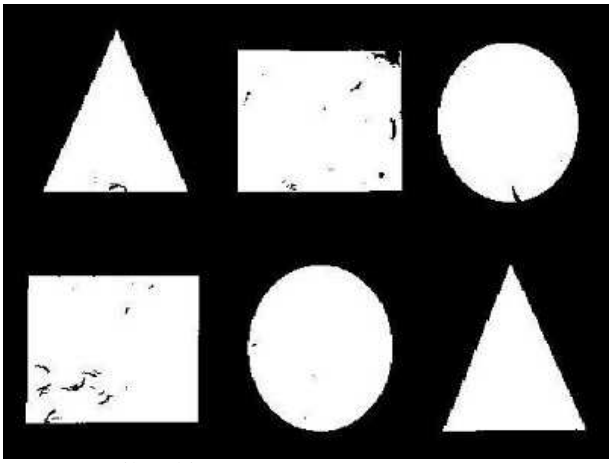

Figure 41. Dilated image.

have as the device of capture the camera of a Smartphone, which thanks to the application used (IP Webcam) allows the

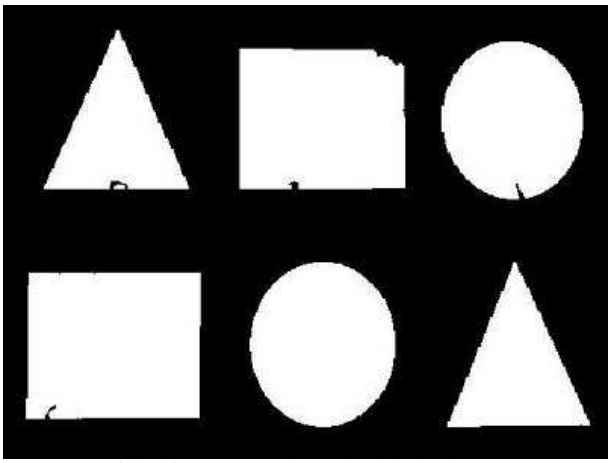

Figure 42. Filled image, ready to be labeled.

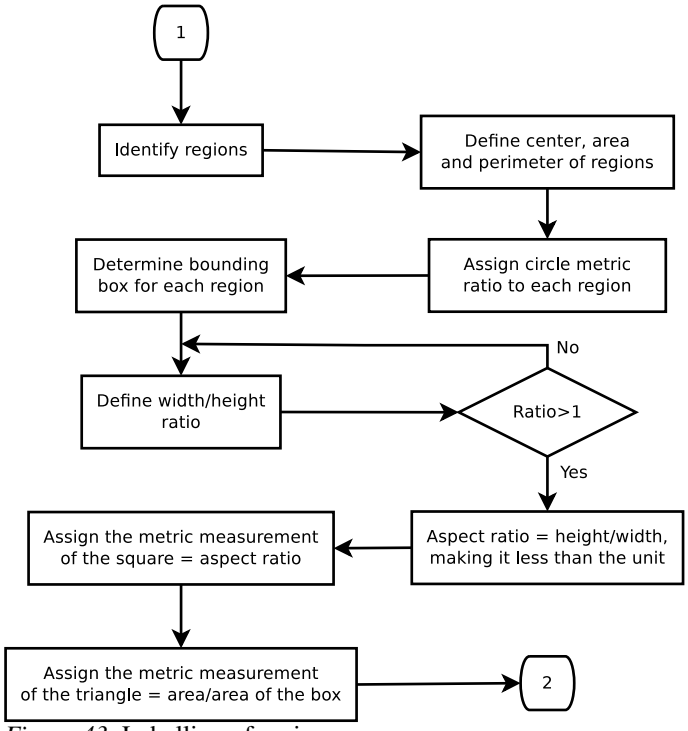

Figure 43. Labelling of regions.

use of the resolution of the camera between 8 Megapixels to 13 Megapixels.

The image format is one of the important elements for image processing. A certain format allows some easiness in operations that minimize the search of what we need, or it allows to transform the image to convenience. In the case of León and Valderrama (León & Valderrama, 2016), HSV was used and for our case the RGB format. For the RGB format, we take into account the intensity of the primary colors of the light (Mingjing, Lili, & Jingke, 2010), in HSV is taken into account the degradation

in the colors seen by the RGB but in terms of Hue (given in degrees) and saturation (given in percentage) (Santos, Marcolino, Lopes, Matheus, & Assis, 2016).

To correct irregularities in the image, it is necessary to use methods based on morphological operations, such as erosion, dilation, and filling of shapes that give a soft touch to the image, allowing the search to be minimized. In the case of León and Valderrama (León & Valderrama, 2016), as well as our case, the same operations were used.

The detection of shapes varies in the parts, both for Leon and Valderrama and for us. The approach of León and Valderrama (León & Valderrama, 2016), was to analyze from the contours (edges) of the filtered areas according to the color channel, at the same time compared with a polynomial approximation according to the found contours taking into account the number of points contained in the found contour, with this established the geometric figure. On the other hand, the second algorithm took into account the contours but measured the length of the found contours and compared them with the shape of a circumference drawn in a theoretical way. Then, according to a comparison of excess areas, the geometric figure is determined.

In our case, we take into account the labeling of regions or search for regions, in which mathematical relations are carried out that compare their results below or above the unit due to a minimum rectangle found for each region. There is a similarity with that used by León and Valderrama (León & Valderrama, 2016), in their second algorithm.

It is important to use robust software with the ability to perform image processing effectively. For the case of Leon and Valderrama (León & Valderrama, 2016), they were supported in Python and the OpenCV 2.4.11 API on the Raspbian operating system, which uses the Raspberry Pi Model B+ embedded system. This generated some difficulty in real-time image processing, as its response speed is somewhat slow. In our case, and as a strategy to increase performance, the platform used was the Android operating system in a mid-range Smartphone. This allowed the installation of the right application to give usefulness to the camera, at the same time taking advantage of by MATLAB.

The performance of the León and Valderrama (León & Valderrama, 2016) algorithm, evaluated in this research, showed irregularity according to the luminous intensity of the environment. Its detection is optimal at higher luminosity, but in medium and low luminosity the detection deteriorates. At the same time, the detection is limited to three or four areas found. For our case, we used an average threshold in the blue channel of the RGB format, which behaves in a stable way to medium and high light intensity. On the other hand, region detection helps us to obtain more detected shapes.

Results and performance

The algorithm implemented in Matlab, and with the help of the high-resolution camera of the Smartphone, was achieved the recognition of the desired figures. Some of the results achieved are shown below (Figs. 44 to 47).

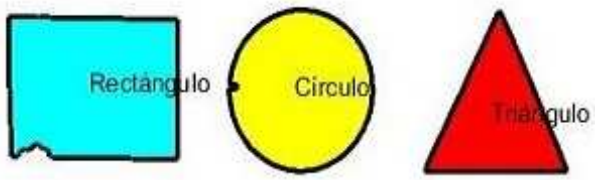

Figure 44. Three figures case 1.

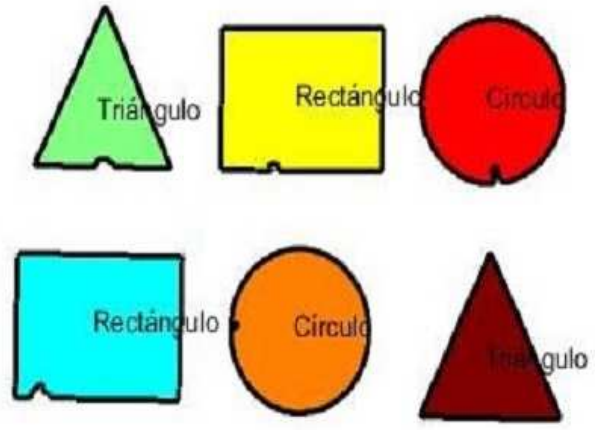

Figure 45. Six figures case 1.

Figure 46. Three figures case 2.

It is observed that the recognition of the figures of the form one paints of a random color the found regions. For this case, it corresponds to geometric figures specifying the name of each one. In the second form, a border is made on the image of green color and also indicating us the name of each one of them.

Conclusions

We designed a low-cost prototype very versatile, which allows us to make a better sketch and recognition of the desired geometric figures. The real-time geometric figure recognition strategy is based on a bounding rectangle. This

ISSN 1692-8407

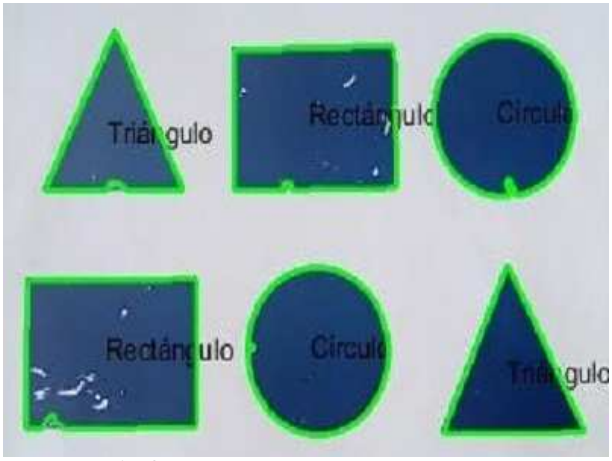

Figure 47. Six figures case 2.

is formed from the regions found and allows to determine the figure that is in it thanks to the minbounrect function (Matlab toolbox). It is useful when the contour of the figure is continuous, i.e. there are no cut lines and it works for closed regions. Each of the proposed objectives was achieved, providing an improvement in the recognition of geometric figures with respect to previous algorithms. We determined that our algorithm is 20% faster for figure detection using Matlab than Python and OpenCV. In terms of performance, this new implementation has a very low margin of error and less sensitivity to the luminous intensity of the environment. However, since the development goal is to achieve an embedded solution for small robots, it is necessary to continue this research in order to implement the algorithms on a smaller platform.

Acknowledgements

We thank and recognize both the Universidad Distrital Francisco José de Caldas in general, as well as the ARMOS research group, which gave us its collaboration and support during the realization of the project. The ideas and postulates described here do not compromise in any way the opinions of either the District University or the ARMOS research group.

References

Arduino. (n.d.). *Language reference.* Retrieved from `https://www.arduino.cc/en/Reference/HomePage`

Aslan, M., Abdelmunim, H., & Farang, A. (2011). Probabilistic shape-based segmentation using level sets. In *Ieee international conference on computer vision workshops (iccv workshops 2011)* (p. 182-194).

Burger, W., & Burge, M. (2008). *Digital image processing* (1st ed.; Springer, Ed.).

Chegtian, S., Keyong, W., & Lian, Z. (2008). The image processing and target identification of laser imaging fuze. In *3rd international conference on intelligent system and knowledge engineering (iske 2008)* (p. 3589-3594).

Jacinto, E., Martínez, F., & Martínez, F. (2016). Learning strategies for cryptography using embedded systems. *Smart Innovation, Systems and Technologies*, *59*(1), 495-505.

Lee, M., Feng, L., Yeh, E., & Ivan, I. (2013). Visual sensor integration on servoing the autonomous mobile robot. In *Ieee/sice international symposium on, system integration (sii 2013)* (p. 185-189).

León, J., & Valderrama, H. (2016). *Sistema minimalista embebido de reconocimiento en tiempo real de figuras geométricas*. Unpublished master's thesis, Universidad Distrital Francisco José de Caldas.

Louis, C., & Ricky, M. (2011). Global path planning in mobile robot using omnidirectional camera. In *International conference on consumer electronics, communications and networks (cecnet 2011)* (p. 4986-4989).

Martínez, F., Jacinto, E., & Zárate, D. (2015). Anthropometric humanoid robot concept for research in control. *Tecnura*, *19*(E), 55-65.

Martínez, F., Montiel, H., & Jacinto, E. (2016). Inductive teaching and problem-based learning as significant training tools in electrical engineering. *Smart Innovation, Systems and Technologies*, *59*(1), 179-188.

Martínez, F., Montiel, H., & Valderrama, H. (2016). Using embedded robotic platform and problem-based learning for engineering education. *Smart Innovation, Systems and Technologies*, *59*(1), 435-445.

MathWorks. (n.d.). *Image processing and computer vision.* Retrieved from `https://es.mathworks.com/solutions/image-video-processing.html`

Mingjing, A., Lili, Z., & Jingke, W. (2010). Region based inter-color intra prediction for rgb signals. In *3rd international conference on computer science and information technology* (p. 41-44). (https://es.wikipedia.org/wiki/RGB)

Ponte, H., Queenan, M., Gong, C., & Mertz, C. (2014). Visual sensing for developing autonomous behavior in snake robots. In *Ieee international conference on, robotics and automation (icra 2014)* (p. 2779-2784).

Processing. (n.d.). *Images and pixels.* Retrieved from `https://processing.org/tutorials/pixels/`

Samadi, M., & Othman, M. (2013). Global path planning for autonomous mobile robot using genetic algorithm. In *International conference on signal-image technology & internet-based systems (sitis 2013)* (p. 726-730).

Santos, A., Marcolino, F., Lopes, M., Matheus, V., & Assis, Z. (2016). Object tracking by color and active contour models segmentation. *IEEE Latin America Transactions*, *14*(3), 1488-1493.

Schoellig, A., Barfoot, T., & Pfrunder, A. (2014). A proof-of-concept demonstration of visual teach and repeat on a quadrocopter using an altitude sensor and a monocular camera. In *Canadian conference on computer and robot vision* (p. 238-245).

Tsai, C., Chuang, C., Lu, L., & Wang, W. (2013). Machine-vision based obstacle avoidance system for robot system. In *International conference on system science and engineering (icsse 2013)* (p. 273-277).