A GLOBAL VISION SYSTEM FOR MOBILE MINI-ROBOTS

Kelson Rômulo Teixeira Aires
Universidade Federal do Rio Grande do Norte, LECA – CT – UFRN, Lagoa Nova, Natal – RN, Cep.: 59072-970
kelson@engineer.com

Pablo Javier Alsina
Universidade Federal do Rio Grande do Norte, LECA – CT – UFRN, Lagoa Nova, Natal – RN, Cep.: 59072-970
pablo@leca.ufrn.br

Adelardo Adelino Dantas de Medeiros
Universidade Federal do Rio Grande do Norte, LECA – CT – UFRN, Lagoa Nova, Natal – RN, Cep.: 59072-970
adelardo@leca.ufrn.br

***Abstract**. The navigation based on visual feedback for a robot team working in a closed environment can be attained through an onboard camera in each robot. A global vision system is a cheapest solution for this problem. This paper presents the implementation and experimental results of a global vision system for a mobile robot team. The proposed experimental system consists of a top camera, a frame grabber, a PC for image processing, and a team of six mobile robots. The PC is responsible for the team motion control. In order to the system be able to univocally recognize each robot, each one has two circular labels on its top. The arithmetic mean of their centroid coordinates gives the actual position of the robot. The vector lying both centroids is utilized to obtain the robot orientation. A great problem of the proposed system was the classification of each pixel color of the robot labels images in real time and under time-variant illumination conditions. To overcome this problem, automatic camera calibration software, based on clustering K-means algorithm, was implemented. This method guarantees that similar pixels will be clustered around a unique color class. The obtained experimental results show that the implemented system updates the robot with acceptable accuracy (position error less than 0.5 cm). The developed system presents a great robustness under large illumination changes.*

***Keywords:** robot vision, image processing, k-means.*

1. Introduction

With the advent of the Artificial Intelligence, researchers began to think about the possibility of endowing machines with the capacity to take more complex decisions, so as to adapt the machines to unexpected variations in their environment. It was the first step for the development of Autonomous Robot Systems. An important property of these systems is the fact that they can interact, in an intelligent way, with their work space, feeling what happens around them and acting in their environment in agreement with the felt information. For this, several sensors can be installed in the robot for the acquisition of every kind of information data.

In order to increase the robot autonomy, different sensorial systems were utilized in practice. A Vision System can be considered as another sensor, between the several ones that can be coupled to a robot system, that allow it to sense the world around. Digital image processing is a vast field, with several applications in the science and engineering. Image Processing assures the possibility of development of last generation machines, capable of executing several functions of the human vision (Jain, 1989). Vision Systems are capable of supplying a large amount of useful information about the robot workspace and computer vision techniques are utilized to extract, in the most possible efficient way, a great amount of useful information from the sensed images.

In cooperative robotic systems, the information about robot position and orientation is indispensable. In order to obtain this information, a vision system can be used. The vision system should be able to sense the dynamic environment where the robot team navigates and to supply the necessary information for the robot control system. An efficient and robust vision system should be capable to act in different environments, with time-variant illumination and color conditions, maintaining, at the same time, processing speed and accuracy.

This paper describes the implementation details of a global vision utilized for the navigation of a mobile robot team. The proposed experimental system consists of a top camera covering all the robots workspace, a frame grabber, a host PC for image processing, and a robot team, whose images are captured by the camera. The PC is responsible for the robot team motion control based on visual feedback, sending commands to them through a radio link. In order to each robot be univocally recognized by the vision system, each one has a label on its top, consisting of two colored circles. One of the most complex tasks on the system implementation was to classify, in real time, each image pixel color of the robot labels under time-variant illumination conditions. In order to overcome this problem, automatic camera calibration software, based on clustering K-means algorithm (Alsabti et all, 1997), was implemented. This method guarantees that similar pixels will be clustered around unique color class. In this paper, the general architecture of the proposed vision system is described in Section 2. Section 3 presents several issues related to the choice of the adequate color model to be adopted for image processing in the proposed vision system. The implemented calibration method is detailed in Section 4. The adopted robot localization and identification algorithms are described in Section 5. Section 6 presents a discussion about the experimental results obtained from the implemented vision system. Concluding remarks are presented in Section 7.

2. Vision system architecture

The proposed vision system can be divided in three modules, as shown in the Fig. (1). The acquisition module obtains the image of the workspace through a CCD camera and a frame grabber. The Image Processing module computes, from the acquired image, the position and orientation of each robot. . As results of the data processing, position, orientation and identification of the objects in the workspace are supplied for robot motion control purposes. In order to adequate the vision system to local illumination conditions, some color calibration is need before any image processing. The calibration of the system can be made off-line or, when necessary, in real-time. The following subsections detail the modules of the proposed system.

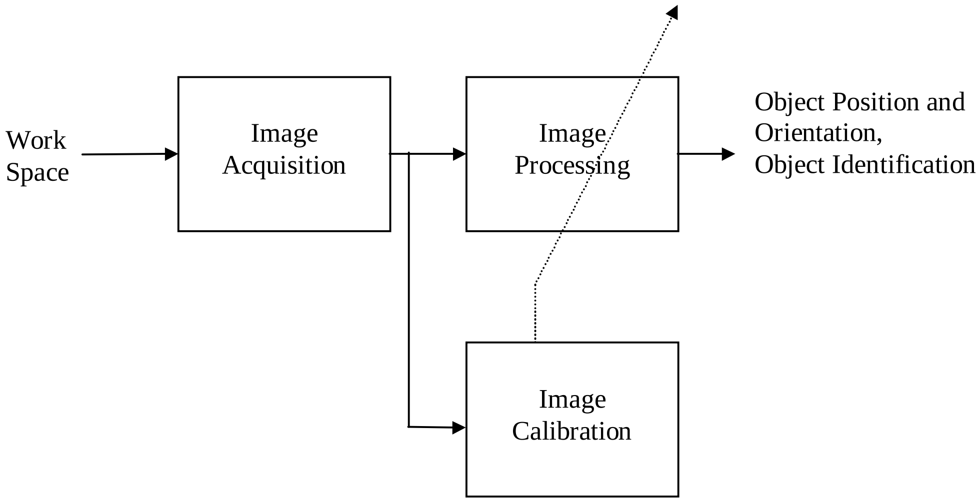


Figure 1. Vision system architecture.

2.1. Image acquisition

There are two primordial elements in a digital image acquisition system (Gonzales and Woods, 2000). The first is a physical device sensitive to a band of the electromagnetic spectrum and that produces an electric signal proportional to the energy level of the sensed radiation. The second element, called frame grabber, is a device for conversion of the electric signal into a digital value. In this implementation, it was utilized a CCD (Charged Coupled Device) colored camera located 2 m above the workspace, and a frame grabber, which furnishes the acquired digital image to a PC (Personal Computer), which is responsible by the rest of the image processing. The frame grabber supplies data in three matrixes of color components, where each of their elements is associated to the correspondent color component of each image pixel. The standard color models furnished by the utilized frame grabber are RGB and HSL.

2.2. Color calibration

The calibration stage frequently is performed off-line, previously to the image processing. Its goal is the tuning of the image processing module, in order to optimize the object detection. After the choice of an appropriate color model, thresholds for classification of the image colored objects are defined for each of their color components. The calibration process can be made manually, by the operator, or in an automatic way, through a calibration algorithm. The calibration consists, basically, in the adjustment of these thresholds, sometimes including other processing parameters used in the vision system, such as: brightness, saturation, contrast and image resolution. These data are supplied as initialization parameters for the Image Processing Module.

2.3. Image processing

After a good calibration, the acquired image is ready for processing. The localization and identification routines should be performed. The localization algorithm supplies the position and the orientation of all robots in scene. The identification process consists on determinate who is each object located in the image, i.e., the identification algorithm should recognize the detected objects, associating them to well-known models. The used algorithm should act in a fast and efficient way, analyzing the largest possible amount of pictures per second.

3. The choice of the color model

3.1. Color models

The choice of an appropriate color model to codify the image information is the first step in the design of a vision system. A color model determines a subspace of a three-dimensional coordinate system, where each point represents a color. The main objective of a Color models is to facilitate the representation of the colors in some acceptable standard form. They can be classified in two types: hardware guided models (RGB, YUV and CMY) and users guided models (HSV, HSL and HVC), (Foley et all, 1990).

The frame grabber utilized in the implemented system furnishes both RGB and HSL video output. It is possible to use another color model through an additional conversion process, which can be easily implemented by software, although with a larger computational cost.

3.2 RGB color model

RGB is the most used color model. It is based on a cartesian coordinate system, where each color is represented by its primary components: **R**ed, **G**reen and **B**lue. The mixture of these three primary colors with full intensity results in the white color and its absence results in the black color. Any possible color is composed as a mixture of specific amount of red, green and blue. RGB color model can be represented, with normalized values in the interval [0,1], in agreement with the subspace of a three-dimensional coordinate system, as we can see in the Fig. (2).

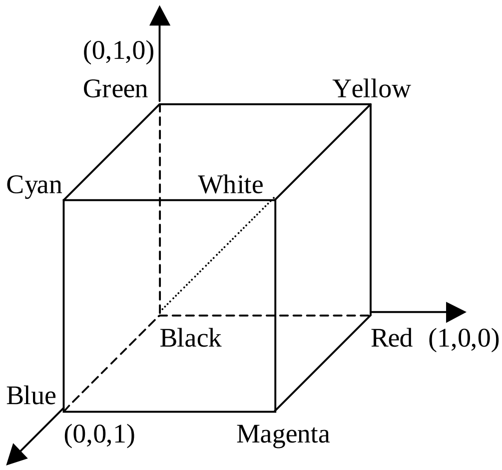


Figure 2. RGB color cube.

The origin of the coordinate system corresponds to the black color, being the white represented by the vertex (1,1,1) of the color cube. In this cube, the gray scale varies from black to white, along the straight line that joins these two vertexes. The primary colors (red, green and blue) are represented by the vertexes of the cube on the axes of the coordinates system. The remaining vertexes represent the complementary colors (yellow, cyan and magenta) (Foley et all, 1990).

3.3. HSL color model

HSL (Hue, Saturation, and Lightness) color model is based on intuitive color parameters, being derived from the RGB color cube. It is represented by a double hexagonal pyramid as can be seeing in Fig. (3) (Hearn et all, 1994). Hue (H) specifies an angle about the vertical axis of the pyramid, varying from 0^o, that corresponds to the red, to 360^o. The hexagon vertexes correspond to the primary colors and its complementary ones. Each vertex of complementary color is shifted 180^o from its respective primary color vertex. The parameter H possesses indefinite value for the gray scale which varies from black to white. Saturation (S) is measured along the horizontal radius of the pyramid and specifies the relative purity of the color. This parameter varies from 0 (gray scale) to 1 (pure colors). Lightness (L), measured along the vertical axis, possesses value 0 for black and 1 for white. It specifies the amount of light in the color.

3.4. Color model choice

The choice of the color model to be utilized in image processing is very important. In order to localize and to identify colored objects in an image, a component or a relationship among components of the color model must be chosen to allow the best possible differentiation among the colors of the different elements in the image. This allows the definition of specific thresholds that will be utilized for the classification of the colored objects in the scene. An adequate choice of these thresholds is fundamental to guarantee the robustness of the vision system in the presence of light changes.

Besides the robustness to illumination variations, a very important factor to be considered in the project of the vision system is the execution time of the image processing routines. In this context, the chosen color model must supply the largest amount of color information, in the smallest amount of data, in order to turn more efficient the process of classification of the colors of each object. HSL model presents a great separation between color (component

H) and lightness (component L) characteristics. Therefore, light variations in the image interfere directly in the component L, having little influence in the values of H (Gonzales and Woods, 2000). As the separation of color characteristics and luminance is not complete, a safety margin must be considered when working with HSL model. A disadvantage of the HSL model is that the parameter H is undefined in the gray scale. This is the case in the present implementation, where the robot team navigates on a black floor. This is the reason because the RGB model was chosen for our vision system. On the other hand, RGB model necessarily uses the three components to define the colors and the amount of light in the image, which implies that, when using the RGB model, all the three components, or a relationship among them, must be utilized, increasing the computational effort.

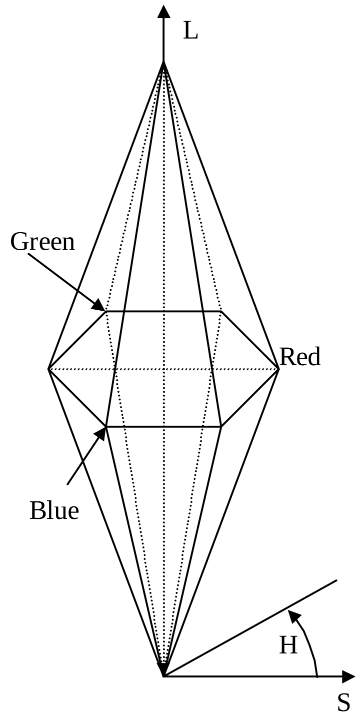


Figure 3. HSL double hexagonal pyramid.

4. Automatic calibration method

Color calibration one of the most important stages of the vision system. The main objective of the calibration is to facilitate the identification and the classification, with respect to the color, of all image pixels (MIN98). The calibration stage consists, basically, in an analysis of the data of an image model under certain light conditions in order to define the optimal parameters utilized by the vision system for pixel classification. This stage is very important, because the system must be calibrated in agreement with the available illumination. This work proposes a form of calibrate the vision system automatically based on the K-means clustering algorithm.

4.1. K-means clustering algorithm

Several clustering techniques have frequently been recommended as tools for pattern classification. Clustering is a partitioning process of a set of patterns in separated groups (clusters), or classes. Patterns with similar characteristics are classified in the same group, patterns with different characteristics are classified in disjoint groups (Alsabti et all, 1997). The final purpose of all clustering techniques is to find natural groupings through the database analysis. The most popular clustering algorithm is probably the K-means algorithm.

K-means algorithm produces a partition of *n* objects in K clusters, generally optimizing an objective function. Some advantages exist: the possibility of changes on the pertinence of objects in relation to a group during the cluster formation process and the possibility of working with great database. This method requests computation time of order *O(n)*, where *n* is the number of objects in the database (Costa, 1999). K-means algorithm is sensitive to the choice of initial classes. The number of classes K must be known *apriori*, and different solutions could be generated depending on the choice of number K. This is a great disadvantage of K-means. The problem of a wrong choice of K is that the method will impose an arbitrary structure to the data instead of looking for an inherent data structure (Alsabti et all, 1997). Each class C_i is represented by a point (center) w_i and, initially, that class is constituted only by this point. After having chosen the centers, each data is marked as belonging to the class whose center is closer to it. After processing all data according this method, the mean of each class is calculated and used as its new center. Then, the data is processed again. This process is repeated until no changes occur in the found centers or a user defined number of iterations is attained. The whole process is described in the Algorithm I.

Algorithm I

1. Initialize the K centers of the classes (w_1, ... , w_k)
2. Associate each class C_i to a center w_i
3. Include each input data p_j to the class C_i whose center is closer to p_j.
4. For each class C_i, actualize its center w_i as the arithmetic mean of all input data $p_j \in C_i$.
5. Compute de error function:

$$E = \sum_{i=1}^{k} \sum_{p_j \in C_i} (p_j - w_i)^2$$

6. Repeat steps 3, 4 and 5 until error E do not change significantly.

4.2. System calibration

The tuning of some image parameters, (such as brightness, contrast, saturation and resolution), was necessary before color calibration. In The implemented system, this process was manually executed, through the software that accompanies the frame grabber. Another important parameter is the height of the camera above the robot team, which was previously adjusted, because it directly influences the values of the other mentioned parameters.

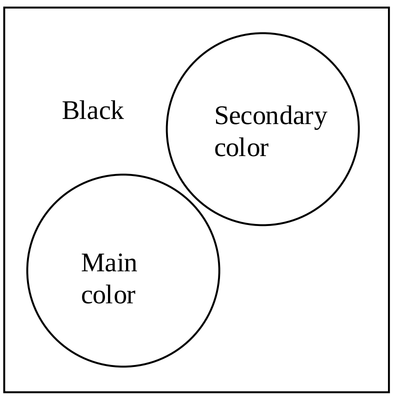
Color calibration was made using K-means algorithm. In order to classify each color as belonging to a center, a number of classes K was chosen larger than two times the number the main colors present in the scene, being the extra centers responsible by the classification of image spurious pixels. This was made in order to avoid the imposition of an erroneous data structure, containing spurious pixels in a same class of object color pixels. Initially, 20 centers were chosen, in a random way, among the group of pixels that forms the image to be processed.

After the choice of the centers, the pixel cluster processing was initiated. Due to real time constrains, the established stop condition was a program iteration number equal to 100. This criterion guaranteed a reasonable error in the pixel classification process, small enough for this application. At the end of the 100 iterations, 20 centers were selected representing 20 pixels groupings, where each object color class is represented by a center, and the remaining centers represent undesirable pixels. The choice of the correct groupings that classify, among the 20 final groupings, the desired colors, was realized taking into account the number of pixels belonging to each grouping. For this, a range of values can be established analyzing image resolution, label size and amount of each color label in the image. In order to relate each class with its respective color, its normalized component values were utilized. Each image pixel was classified as belonging to the color class with the nearest center. After system calibration, all colored objects in the image can be classified with respect to the color, with good precision, allowing a more efficient execution of the localization and identification routine.

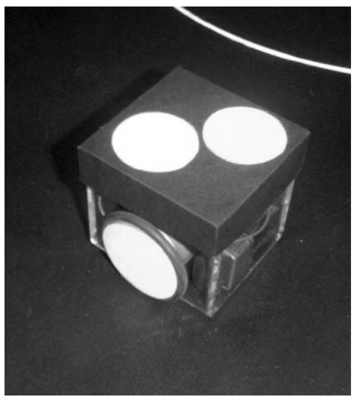
5. Localization and identification

5.1. Labels

Labels were attached on the top of each robot to permit its identification and the computation of its position and orientation. There are several possible robot labeling methods. The chosen labeling method is shown in the Fig. (4). Two circular labels were placed in the diagonal line of the robot top. This label consists of a main blue or yellow circle and a secondary green, pink or cyan circle, both placed on a black background. In this way, up to six robots can be univocally labeled: blue-green, blue-pink, blue-cyan, yellow-green, yellow-pink and yellow-cyan. A sufficiently large black background margin is provided to separate labels belonging to different robots when two or more are in collision. This avoids that a pair of circles, each one belonging to a different robot in a collision, could be identified as belonging to a same robot. Thus this strategy avoids a false robot's identification.



(a)



(b)

Figure 4. Robot label. a) Robot labeling method. b) Image of a real robot.

5.2. The localization and identification algorithm

The adopted algorithm for robot localization and identification is a modification of the algorithm proposed by Bianchi et all (2000). The image pixels are analyzed with sampling step smaller than the half of the smallest object diameter (in pixel) in the scene. When a main color pixel (blue or yellow) is found, a centroid calculation algorithm of circular objects is applied.

The centroid calculation algorithm is quite simple. After being found a pixel P_{ini} with a desired color, a diagonal segment is traced until finding the border points P_1 and P_2 of the colored circle. The center P_c of the previous segment is calculated and, starting from it, a perpendicular segment is traced until reaching the border points P_3 and P_4. The second segment center, $P_{central}$, corresponds to the centroid of the colored circular label, as depicted in the Fig. (5).

After finding a main color circle, the secondary color circle is searched around the first, at a distance equal to the circle diameter, analyzing the neighborhood using a sampling step smaller than the half of center diameter. When a secondary color pixel is found, the same centroid calculation algorithm is used. The arithmetic average of the circle centroids coordinates supplies the robot position, as depicted in Fig. (6). The segment that links the main circle centroid to the secondary circle centroid is utilized to obtain the robot's orientation.

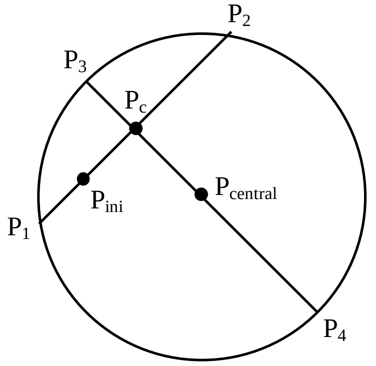


Figure 5. Centroid calculation algorithm of circular objects.

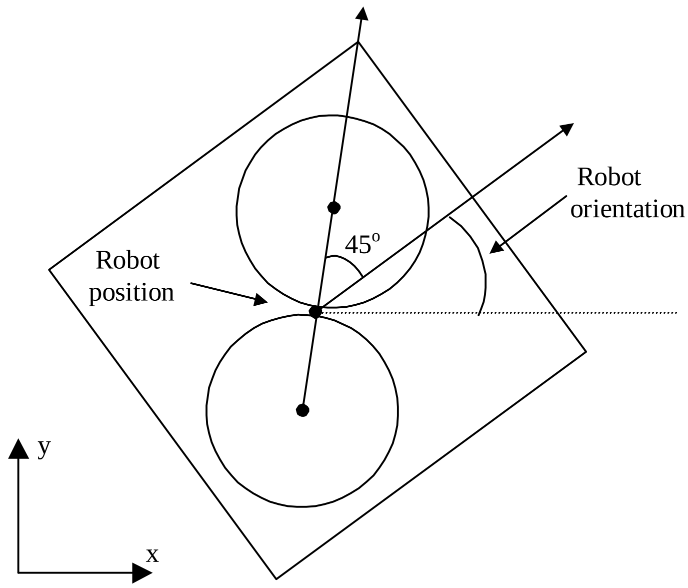


Figure 6. Robot position and orientation.

6. Experimental results

In order to validate the proposed vision system, some tests were performed with the calibration and localization modules. Color calibration was realized in RGB and HSL robots' images, under several illumination conditions. Twenty initial centers were randomly chosen and the algorithm stop condition established as 100 iterations. In all tested situations, the system was calibrated in less than 2 minutes for definition of the 20 groups and less than 1 minute for correct choice of the groups that classify the colors in use (blue, yellow, cyan, rose and green).

Calibration based on the HSL color model was very difficult, due to white and black colors of the floor, which are predominant in the image, resulting in undefined values for the H parameter. K-means algorithm imposes an erroneous structure to HSL pixels. The RGB color model was shown more efficient for this kind of calibration. All used colors were correctly clustered and the RGB color model was chosen as the more adapted for this calibration application.

In order to verify the precision of the proposed localization system, 100 stopped robot images were acquired and its position and orientation was computed by the implemented algorithm. The computed results were compared with the more precise values, obtained by direct measure, (by means of a millimetric ruler), of position and orientation. These results are shown in the Tab. (1), where SD means standard deviation and ME is the absolute mean error. Robot position was obtained with precision less than 2 mm and robot's orientation with precision less than 4^o.

Another experiment was realized in order to verify the system robustness in the presence of illumination variations and possible optical distortions of the camera along the workspace. A robot was placed in different locations of the workspace and its position and orientation was computed by means of the vision system. The results for this test are shown in Tab. (2). The results were similar for all the tested positions, proving the robustness of the proposed vision system to small illumination changes due to irregular spatial lights distribution.

A third experiment was realized to verify the robustness of the proposed vision system under severe changes in the illumination. The vision system was calibrated under 50% of the standard illumination. After this, the vision system was operated under standard illumination (1000 lux). The results obtained from this test are shown in Tab. (3). According to these results, the measured position and orientation errors don't differ significantly of those of the other experiments, showing that the proposed system is robust to great illumination changes.

Table 1. Experimental results for test 1 – Measure of position and orientation error.

	Position (x,y) (cm)				Orientation (^{o})			
Robot	Real	Mean	SD	ME	Real	Mean	SD	ME
Yellow-cyan	(12,-1)	(11.98,-1.13)	0.13	0.17	145	143.66	2.39	2.47
Yellow-green	(-19.3,32.5)	(-19.26,32.38)	0.16	0.19	-125	-125.28	2.89	2.52
Blue-pink	(-30,25.5)	(-29.94,25.34)	0.12	0.20	160	156.97	2.43	3.46

Table 2. Experimental results for test 2 (blue-cyan robot) – Robustness to spatial illumination changes.

Workspace position	Position (x,y) (cm)				Orientation (^{o})			
	Real	Mean	SD	ME	Real	Mean	SD	ME
1st quadrant	(21,20)	(20.83,20.32)	0.18	0.37	-165	-163.79	4.08	2.78
2nd quadrant	(-50,45)	(-50.58,45.94)	0.26	0.48	125	120.53	4.84	5.27
3rd quadrant	(-35,-35)	(-35.99,-34.65)	0.14	0.32	-30	-28.44	4.17	3.58
4th quadrant	(21,-18)	(21.93,-18.16)	0.14	0.25	-105	-106.97	1.44	2.12
Center	(0,0)	(0.08,0.16)	0.13	0.20	0	1.38	2.80	2.30

Table 3. Experimental results for test 3 (yellow-pink robot) – Robustness to 100% of illumination changes.

Position (x,y) (cm)				Orientation (^{o})			
real	Mean	SD	ME	Real	mean	SD	ME
(24,22)	(24.78,21.98)	0.15	0.38	127	125.98	2.58	2.33

In all performed tests, the position was obtained with precision always less than 5mm. The error in the computed orientation was always less than 5 degrees in all experiments. Both robot's position and orientation were updated in real time, analyzing 30 frames per second. In all cases, the localization routine computed the position and orientation, and identified all robots, approximately in 14 milliseconds, which is smaller than the image acquisition time interval of the frame grabber (33 milliseconds). This leaves enough computation time for the additional robot control routines.

7. Conclusions

Analyzing the experimental results of the proposed global vision system, we can present some features of our vision system:

- *Easy and fast off-line calibration.* The system was quickly calibrated in less than three minutes.
- *Fast processing.* Our system determinate position and orientation all robots in real time, analyzing 30 images per second.
- *Reliability*. The proposed vision system can determinate the robot position and orientation without previous frame information. If any robot position can not be determined in the current frame, this will be possible in the next frame but, until now, in all tests, all robot positions and orientations were found successfully.
- *Robustness*. The vision system was robust to the time-varying illumination. All colors were separated and clustered, also under severe illumination changes.
- *Precision.* The robot's localization was calculated with precision less than 5mm for position and 5^o for orientation.

One problem faced in K-means algorithm is the choice of the initial classes. Depending oh this choice, the final values can converge to a local minimum. In spite of this, in our implementation, this kind of problem did not occur.

The proposed calibration technique can be easily adapted to other robot vision applications. The proposed calibration technique can be utilized in other applications involving classification of pixels according its color. At the moment, the use of this calibration system in visual servo controller for robot manipulator is under study.

8. References

Alsabti, K., Ranka, S., and Singh, V., 1997, "An Efficient K-Menas Clustering Algorithm", disponible at *http://www.cise.ufl.edu/~ranka.*

Bianchi, R. A. C. and Costa, A. H. R., 2000, "O Sistema de Visão Computacional do Time FUTEPOLI de Futebol de Robôs", Proceedings of XIII Congresso Brasileiro de Automática, Florianópolis, Brazil, pp 2156-2161.

Campos, M., Anício, M., Carvalho, M., Dias, R., Hartman, A., Nagem, D., Oliveira, V., Oliveira, E., Pereira, G., Ribeiro, A., Sanches, F., and Silveira, M. A., 1998, "MIneiROSOT – The Development of a Centralized Control Set of Soccer-playing Micro-robots", Proceedings of FIRA Robot World Cup, Paris, França, pp 57-62.

Costa, J. A. F., 1999, "Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis", PhD Thesis, University of Campinas – UNICAMP.

Foley, Van Dam, Feiver and Hughes, 1990, "Computer Graphics: Principles and Pratice", Addison-Wesley.

Gonzales, R. C. and Woods, R. E., 2000, "Processamento de Imagens Digitais", Editora Edgard Blücher LTDA.

Hearn, Donald and Baker, M. Pauline, 1994, "Computer Graphics: Principles and Practice", Prentice Hall.

Jain, A. K., 1989, "Fundamentals of Digital Image Processing", Prentice Hall.