A modular real-time vision module for humanoid robots

Alina Trifan, António J. R. Neves, Nuno Lau, Bernardo Cunha
IEETA/DETI
Universidade de Aveiro, 3810–193 Aveiro, Portugal

ABSTRACT

Robotic vision is nowadays one of the most challenging branches of robotics. In the case of a humanoid robot, a robust vision system has to provide an accurate representation of the surrounding world and to cope with all the constraints imposed by the hardware architecture and the locomotion of the robot. Usually humanoid robots have low computational capabilities that limit the complexity of the developed algorithms. Moreover, their vision system should perform in real time, therefore a compromise between complexity and processing times has to be found. This paper presents a reliable implementation of a modular vision system for a humanoid robot to be used in color-coded environments. From image acquisition, to camera calibration and object detection, the system that we propose integrates all the functionalities needed for a humanoid robot to accurately perform given tasks in color-coded environments. The main contributions of this paper are the implementation details that allow the use of the vision system in real-time, even with low processing capabilities, the innovative self-calibration algorithm for the most important parameters of the camera and its modularity that allows its use with different robotic platforms. Experimental results have been obtained with a NAO robot produced by Aldebaran, which is currently the robotic platform used in the RoboCup Standard Platform League, as well as with a humanoid build using the Bioloid Expert Kit from Robotis. As practical examples, our vision system can be efficiently used in real time for the detection of the objects of interest for a soccer playing robot (ball, field lines and goals) as well as for navigating through a maze with the help of color-coded clues. In the worst case scenario, all the objects of interest in a soccer game, using a NAO robot, with a single core 500Mhz processor, are detected in less than 30ms. Our vision system also includes an algorithm for self-calibration of the camera parameters as well as two support applications that can run on an external computer for color calibration and debugging purposes. These applications are built based on a typical client-server model, in which the main vision pipe runs as a server, allowing clients to connect and distantly monitor its performance, without interfering with its efficiency. The experimental results that we acquire prove the efficiency of our approach both in terms of accuracy and processing time. Despite having been developed for the NAO robot, the modular design of the proposed vision system allows it to be easily integrated into other humanoid robots with a minimum number of changes, mostly in the acquisition module.

Keywords: Robotics; robotic soccer; computer vision; object recognition; humanoid robots; color classification.

1. INTRODUCTION

Humanoid robotics is the branch of robotics that focuses on developing robots that not just have an overall appearance similar to the human body but can also perform tasks that until now were strictly designated for humans. From taking care of the sick and/or elderly people, to playing football or even preparing for inhabiting a space shuttle, humanoid robots can perform some of the most common, yet unexpected tasks that humans undergo daily. Most humanoid robots are fully autonomous, which means that human interaction is needed only for their maintenance. They should be able to perform in unstructured environments and to continuously learn new strategies that can help them adapt to previously unknown situations. Their overall appearance imitates the human body, this meaning that their physical architecture includes a head, a trunk, two legs and two arms.

Probably the most important sense for a humanoid robot is vision. Just like in the case of humans, the only way for a robot to understand the world with every visible objects that are surrounding it is by means of vision. The vision system is responsible for creating an accurate representation of the surrounding world, allowing the classification of objects so that they can be recognized and understood by the robot. Implementing a robust vision system for a humanoid robot is not an easy task since its performance is strongly influenced not just by the hardware architecture of the robot but mostly by its body movements. In this paper we provide a detailed description of a real-time modular vision system based on color

{alina.trifan,an,nunolau}@ua.pt,mbc@det.ua.pt

classification for a humanoid robot. The main physical environment for testing our software was the one of robotic soccer. Moreover, as a second approach, we have used the vision module implemented as the relying sense of a humanoid robot that navigates through a maze. We start by presenting some features of the RoboCup Standard Platform League[1] and of the Micro Rato[2] competition. We continue by presenting an overview about the system, outlining its modularity which makes it intuitively easy for being exported to other humanoid platforms. Then we propose an algorithm for self-calibration of the parameters of the camera. The algorithm uses the histogram of intensities of the acquired images and a white area, known in advance for estimating the most important parameters of the camera, such as: exposure, gain and white balance. For the color segmentation algorithms a lookup table and horizontal or vertical scan lines are used. Finally, we present some validation approaches for a good recognition of the objects of interest in both situations previously described.

2. ROBOCUP STANDARD PLATFORM LEAGUE AND THE NAO ROBOT

One of the most challenging research areas in humanoid robotics is humanoid soccer, promoted by the RoboCup organization. The overall goal of RoboCup is that, by 2050, a team of fully-autonomous robots wins a soccer game against the winner of the most recent World Cup. Even though the goal might seem slightly unrealistic and might not be met in the near future, it is important that such a long range goal be claimed and pursued. One of the most popular soccer league in RoboCup is the Standard Platform League (SPL). In this league all teams use identical, standard robots which are fully autonomous. Therefore the teams concentrate on software development only, while still using state-of-the-art robots. Omnidirectional vision is not allowed, forcing decision-making to trade vision resources for self-localization and ball localization. The league replaced the highly successful Four-Legged League, based on Sony's AIBO dog robots, and is now based on Aldebaran's NAO humanoids.[3] Even though this paper presents a modular vision system that can be applied to a wide range of humanoid robots, a platform for it to be tested was needed. The first chosen solution was to integrate the vision system into the NAO robots of the Portuguese Team, a newly formed team of SPL soccer from the University of Porto and University of Aveiro.[4] The team started in 2010 and attended the first RoboCup competition in July 2011 in Istanbul, Turkey.

In SPL, robots play on a field with a length of $7.4m$ and a width of $5.4m$, covered with a green carpet. All robot-visible lines on the soccer field (side lines, end lines, halfway line, center circle, corner arcs, and the lines surrounding the penalty areas) are $50mm$ in width. The center circle has an outside diameter of $1250mm$. In addition to this, the rest of the objects of interest are also color coded. The official ball is a Mylec orange street hockey ball. It is $65mm$ in diameter and weights 55 grams. The field lines are white and the two teams playing can have either red or blue markers. The red team will defend a yellow goal and the blue team a sky-blue goal.

(a) (b)

Figure 1: On the left, a NAO robot used in the SPL competitions. On the right, an image from the SPL RoboCup 2010 final, between B-Humans and NimbRo.

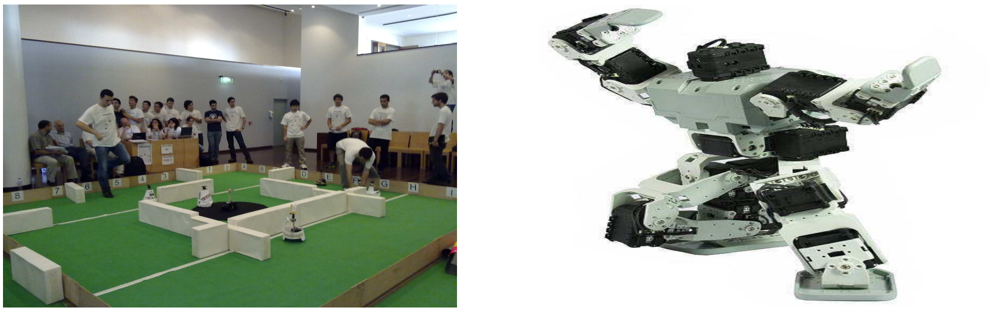
For a soccer playing robot, vision is the only way of sensing the surrounding world. During the game, the playing field provides a fast-changing scenery in which the teammates, the opponents and the ball move quickly and often in an unpredictable way. The robots have to capture these scenes through their cameras and to discover where the objects of interest are located. Everything has to be processed in real time. Since a SPL game is still played in a color coded environment, we propose an architecture of a vision system for a SPL robot based on color classification. The robot can locate the objects of interest like the ball, goals and lines based on color information.

An overview of the work developed so far in this area of robotic vision was needed in order to better understand the context, the challenges and the constraints that robotic vision implies. The structure of the vision system that we are proposing was based on our previous experience in other robotic applications[5] as well as on other related papers such as[6] and.[7] We consider that our approach is an important contribution mainly due to the modularity of our proposal, the real-time capability and the reliability of our system.

3. THE MICRO RATO COMPETITION AND THE BIOLOID HUMANOID ROBOT

The Bioloid platform represents a robotic kit produced by the Korean robot manufacturer Robotis,[8] which consists of several components, namely small servomechanisms Dynamixel, plastic joints, sensors and controllers which can be used to construct robots of various configurations, such as wheeled, legged, or humanoid robots.

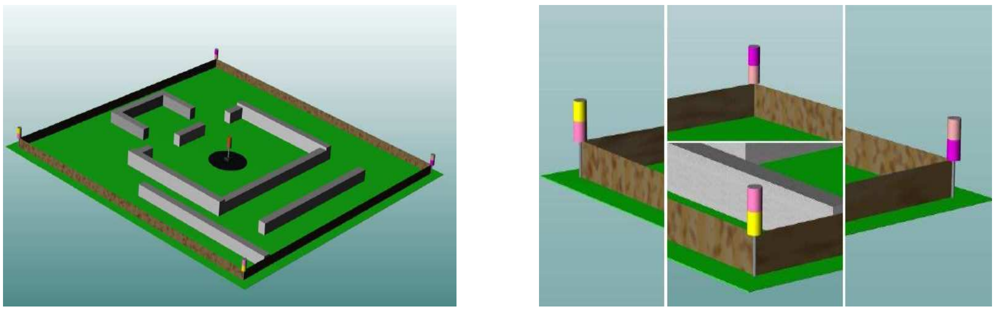
The Micro Rato competition, held at the University of Aveiro is a competition between small autonomous robots whose dimensions do not exceed $300 \times 300 \times 400mm$ (Fig. 2). The competition is divided into two rounds: in the first one, all robots move from a starting area with the purpose of reaching a beacon, in the middle of a maze. In the second round, the robots have to return to the starting area or at least to get as close as possible to it, using the information that they acquired during the first round.



(a) (b)

Figure 2: On the left, an image from the Micro Rato 2011 competition. On the right, an image of the Bioloid robot used.

Most of the robots used in this competition do not rely on vision for accomplishing their tasks. It is more common the use of sensors for detecting the walls of the maze and the area of the beacon, which is an infrared emittor of 28cm high. However, the use of a vision system is possible since there are several elements that allow the detection of the obstacles and the beacon and that can provide information about the localization of the robot.



(a) (b)

Figure 3: On the left, an image of the Micro Rato field. On the right, a graphical representation of the four corner posts and the beacon.

The robots have to move on a green carpet and the walls of the maze are white (Fig. 3 (a)). Moreover, in each of the four corners of the maze there is a two-colored post and the beacon has also two predefined colors. Thus, the corner posts can have either one of the following color combinations: pink-blue, blue-pink, pink-yellow, yellow-pink, while the beacon is half orange, half pink (Fig. 3(b)). The information about the color combination of the posts is helpful for the localization of the robot, in the challenge of reaching the beacon.Therefore, by relying on visual information, it is possible to have a competitive humanoid robot in the context of Micro Rato.

4. SYSTEM OVERVIEW

The architecture of the vision system can be divided into three main parts: access to the device and image acquisition, calibration of the camera parameters and object detection and classification. Moreover, apart from these modules, two applications have also been developed either for calibrating the colors of interest (CalibClient) or for debugging purposes (ViewerClient). These two applications run on an external computer and communicate with the robot through a TCP module of the type client-server that we have developed. The current version of the vision system represents the best trade-off that the team was able to accomplish between processing requirements and the hardware available in order to attain reliable results in real time.

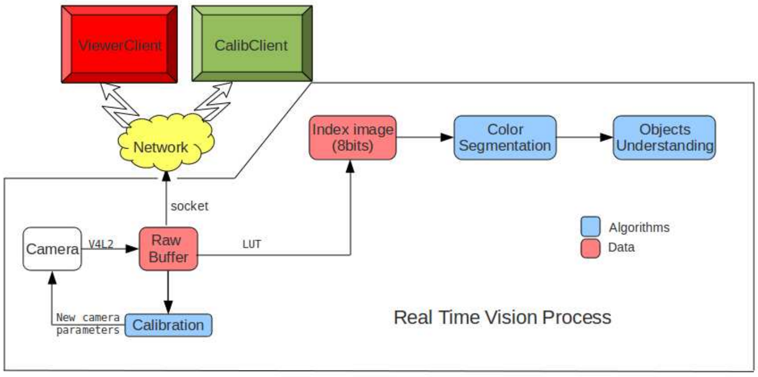


Figure 4: Block diagram of the proposed vision system.

NAO has 2 identical video cameras that are located in the forehead and in the chin area respectively (Fig. 1(a)). They provide a 640×480 resolution at 30 frames per second. The forehead camera can be used to identify objects in the visual field such as goals and balls, while the chin camera can ease NAO's dribbles during a soccer game. The native output of the camera is YUV422 packed. In the current version of the software only the lower camera of the robots is being used since it can provide more meaningful information about the surroundings. However, the software allows to switch between cameras in a small amount of time (29ms). This can be very useful when more evolved game strategies will be developed.

The camera is accessed using V4L2 API, a kernel interface for analog radio and video capture and output drivers. The V4L2 driver is implemented as a kernel module, loaded automatically when the device is first opened. The driver module plugs into the "videodev" kernel module. The access and acquisition module of the system that we are presenting is the only one that might suffer small changes when used with different humanoid robots. Different video devices connected by different technologies to the rest of the hardware can be accessed by making small adaptations to the module that we are proposing. All the other modules can be used as they are on any humanoid robot since their construction is very generic and is not related to any particularities that the NAO robot might have compared to other humanoids.

The video camera that was used with the Bioloid robot was a standard Logitech USB webcam and the process of acquiring images was different than in the case of NAO. The access of the device for the Bioloid camera was done by means of OpenCV, which provides several instinctive methods for accessing and displaying the images. The methods used by OpenCV also rely on Video For Linux v.2. This method was chosen instead of the acquisition module developed for the NAO robot since the NAO camera configuration is accessed through the I2C bus due to its special connection on the processing unit of the robot. The native output of the Bioloid webcam is RGB and it provides the same resolution as the NAO camera.

The calibration module is not continuously running on the robot because of the processing time limitations. It is run just once whenever the environment or the lighting conditions change, having the purpose of setting the parameters of the camera so that the images acquired give the best possible representation of the surrounding world. Details of the algorithm for self-calibration of the camera are presented in Section 5.

For the detection process, with the use of a look-up table, and by means of the OpenCV library, the raw buffer can be converted into an 8-bit grayscale image in which only the colors of interest are mapped using a one color to one bit relationship (orange, green, white, yellow, blue, pink and blue, while gray stands for no color). These colors were common to both applications but our software can be easily adapted to work with a very diverse palette of colors. The next step is the search for the colors of interest in the grayscale image, which we call an index image, by means of vertical or horizontal scan lines, and the formation of blobs. The blobs are then marked as objects if they pass the validation criteria which are constructed based on different measurements extracted from the blobs (bounding box, area, center of mass of the blob). The color segmentation and object detection are detailed in Section 6.

Having the possibility of running the vision module as a server, the two applications that we have developed, CalibClient and ViewerClient can act as clients that can receive, display and manipulate the data coming from the robot. Thus, ViewerClient is a graphical application that allows the display of both the original image as well as the corresponding index image containing the validation marks for each object of interest that was found. This application was essential in terms of understanding what the robot "sees" since most humanoid robots, including NAO, do not have any graphical interface that allows the display and manipulation of images. Also considering the limited resources of these robots the choice of building a graphical interface on the robot was out of the question. CalibClient is a very helpful application that we developed for the calibration of the colors of interest and it is presented in more details in Subsection 5.2.

5. CALIBRATION OF THE VISION SYSTEM

Being still a color coded environment, during a SPL game the color of a pixel in the acquired image is a strong hint for object validation. Also in the Micro Rato competition, each of the four posts has a specific combination of two colors that are known in advance. Because of this, a good color classification is imperative. The accuracy of the representation of the colors in an image captured by the camera of the robot is related to the intrinsic parameters of the camera such as: brightness, saturation, gain, contrast or white balance. By controlling these parameters relatively to the illumination of the environment we can acquire images that accurately represent the real world.

5.1 Self-calibration of the camera intrinsic parameters

The use of both cameras in auto-mode has raised several issues which made the segmentation and validation of objects hard to be performed. By using the camera in auto-mode the images acquired were far from being accurate, mainly due to the environment in which they are used. In both cases, the huge amount of green that is present in the images affect the white-balance of the camera. These kind of applications are synthetic representations of the real world. Moreover, the light in these environments is normally flickering, due to the chosen source of illumination.Thus, the classification of colors was difficult to perform and the process of a robot "learning" a certain color was almost impossible under these conditions.

We propose an algorithm for self-calibration of the camera that is both fast and accurate and requires a minimum amount of human intervention. The algorithm uses the histogram of intensities of the images acquired for calculating some statistic measurements of the images which are then used for compensating the values of the gain and exposure by means of a PI controller. Moreover, a white area, whose location in the image is known in advance, is used for calibrating the white balance. The human intervention is only needed for positioning a white object in the predefined area. The algorithm needs an average number of 20 frames to converge and the processing time of each frame is approximately 300ms.

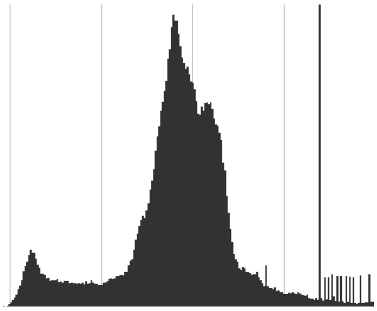
The intensity histogram of an image, that is the histogram of the pixel intensity values, is a bar graph showing the number of pixels in an image at each different intensity values found in the image. For an 8-bit grayscale image there are 256 different possible intensities, from 0 to 255. Image histograms can also indicate the nature of the lighting conditions, the exposure of the image and whether it is underexposed or overexposed. The histogram can be divided into 5 regions. The left regions represent dark colors while the right regions represent light colors. An underexposed image will lean to the left while an overexposed one will be leaning to the right. Ideally most of the image should appear in the middle region of the histogram.

From the intensity histogram the Mean Sample Value (MSV) can be computed based on the following formula and it represents a useful measure of the balance of the tonal distribution in the image:

$$MSV = \frac{\Sigma_{j=0}^{4}(j+1)x_j}{\Sigma_{j=0}^{4}x_j}$$

where x_j is the sum of the gray values in region j of the histogram. The histogram is divided into five regions. The image is considered to have the best quality when the MSV ≈ 2.5. MSV is a mean measure which does not take into account regional overexposures and underexposures in the image. The values for the gain and exposure are compensated with the help of the PI controller until the value of the MSV for the images acquired is ≈ 2.5.

(a)



(b)

Figure 5: On the left an image acquired by the NAO camera after the intrinsic parameters of the camera have converged. On the right, the histogram of the image. As expected, most of the image appears in the middle region of the histogram.

For the calibration of the white balance, the algorithm that we are proposing assumes that the white area should appear white in the acquired image. In the YUV color space, this means that the average value of U and V should be close to 127 when both components are coded with 8 bits. If the white-balance is not correctly configured, these values are different from 127 and the image does not have the correct colors. The white-balance parameter is composed by two values, blue chroma and red chroma, directly related to the values of U and V.

The parameters of the PI controller were obtained experimentally, based on the following reasoning: first, the proportional gain is increased until the given camera parameter would start oscillating. The value chosen for the proportional gain will be 70% of the value that produced those oscillations and the integral gain is increased until the convergence time of the parameters reaches an acceptable value of around 100ms.

An exemple of the use of the proposed algorithm is presented in Fig. 6. As we can see, the image on the right has the colors represented in the same way that the human eye perceives them. On the oposite, in the image on the left the colors are too bright and a distinction between black and blue is difficult to be made.

The algorithm is depicted next:

```
do
 acquire image
 calculate the histogram of intesities
 calculate the MSV value
 while( MSV < 2.3 or MSV > 2.7)
   apply PI controller to adjust gain
   if( gain == 0 or gain == 255)
     apply PI controller to adjust exposure
 end while
set the camera with the new gain and exposure parameters
while exposure or gain parameters change
do
 acquire image
 calculate average U and V values for the white area
 while ( U < 125 or U > 127)
   apply PI controller to adjust red chroma
 end while
 while ( V < 125 or V > 127)
```

```
   apply PI controller to adjust white chroma
 end while
set the camera with the new white balance parameters
while white-balance parameters change
```

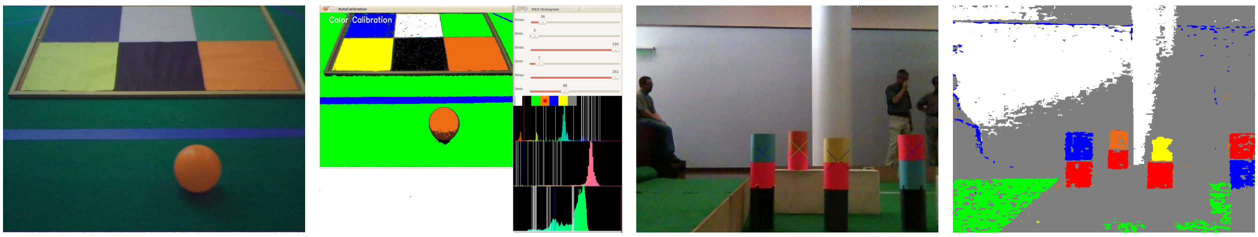
(a) (b) (c)

Figure 6: On the left, an image acquired with the NAO camera used in auto-mode. The white rectangle, in the top middle of the image, represents the white area used for calibrating the white balance parameters. In the middle, an image acquired after calibrating the gain and exposure parameters. On the right, the result of the self-calibration process, after having also the white balance parameters calibrated.

5.2 Calibration of the colors of interest

Along with the calibration of the parameters of the camera (presented in the previous subsection), a calibration of the color range associated to each color class has to be performed whenever the environment changes. These two processes are co-dependent and crucial for image segmentation and object detection.[9] Although the image acquisition is made in YUV (for the NAO robot) and RGB (for the Bioloid robot), the representation of the color range for each of the colors of interest is made in the HSV color space, due to its special characteristics of separating the chromaticity from the brightness.

CalibClient is an application created after a model used by CAMBADA, the RoboCup Middle-Size League team of the University of Aveiro.[10] It allows the creation of a configuration file that contains the Hue, Saturation and Value minimum and maximum values of the colors of interest. Figure 7 presents an example of its use. The configuration file is a binary file that apart from the H, S and V maximum and minimum value also contains the current values of the intrinsic parameters of the camera. It is then exported to the robot and loaded when the vision module starts. These color ranges are used to create the look-up table that for each triplet, RGB or YUV, contains the color information.

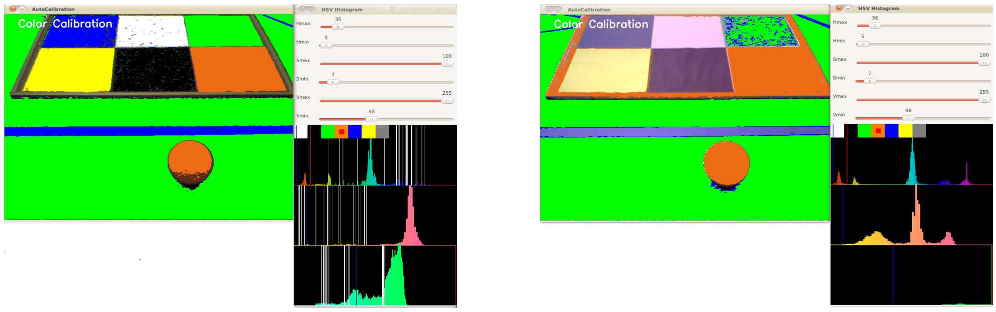


(a) (b) (c) (d)

Figure 7: On the left, the first image is an original image acquired by the NAO camera followed by the same image with the colors of interest classified by means of the CalibClient application. Next, the original image with the markers for all the posts acquired by the Bioloid camera. On the right, the color segmented image.

6. OBJECT DETECTION

For a SPL soccer player robot the objects of interest are: the orange ball, the white lines of the field and the yellow and blue goals. For the Bioloid robot, the objects of interest were the four posts situated in the four corners of the maze and the



(a) (b)

Figure 8: On the left a color calibration after the intrinsic parameters of the camera have converged. On the right, the result of color classification considering the same range for the colors of interest but with the camera working in auto-mode. Most of the colors of interest are lost (the blue, the yellow, the white and the black) and the shadow of the ball on the ground is now blue, which might be confusing for the robot when processing the information about the blue color.

walls that are to be avoided. The four posts have the following combination of colors: yellow and pink, pink and yellow, pink and blue, blue and pink while the beacon is orange and pink. The white walls can be seen as transitions from green (the carpet on which the robot navigates) to white. In this section we present our approach for the detection and validation of the objects of interest, based on color segmentation followed by blob formation and measurements computations for the validation of the blobs.

6.1 Look-up table and the image of labels

In the two contexts chosen for testing the proposed vision system, the color of a pixel is a helpful clue for segmenting objects. Thus color classes are defined with the use of a look-up table(LUT) for fast color classification. A LUT represents a data structure, in this case an array used for replacing a runtime computation with a basic array indexing operation. This approach has been chosen in order to save significant processing time. The image acquired in the YUV format is converted to an index image (image of labels) using an appropriate LUT.

The table consists of 16,777,216 entries (2^{24}, 8 bits for Y, 8 bits for U and 8 bits for V). Each bit expresses whether one of the colors of interest (white, green, blue, yellow, orange, red, blue sky, gray - no color) is within the corresponding class or not. A given color can be assigned to multiple classes at the same time. For classifying a pixel, first the value of the color of the pixel is read and then used as an index into the table. The 8-bit value then read from the table is called the ”color mask” of the pixel.

The resulting index image is a grayscale image with the resolution of 320×240 pixels. A smaller resolution was obtained with the purpose of reducing the classifying time and further decreasing the time spent on scanning and processing the image. In the case of the Bioloid robot, this resolution was obtained by ignoring one in two columns and one in two rows of the original image. For the vision system of the NAO robot, the reduced resolution was obtained by using a subsampling approach. By using the YUV422 packed format of the image, we obtain a subsampling of the image across the image line. For the Y sample, both horizontal and vertical periods are 1 while for the U and V samples the horizontal period is 2 and the vertical one is 1. This means that the two chroma components are sampled at half the sample rate of the luma: the chroma resolution is halved. Moreover, we are presenting an innovative solution for reducing both the processing time and the access to the memory in the process of subsampling the original image acquired by the NAO camera. By converting the YUV422 buffer, which is an unsigned char buffer to an integer one, thus making possible the reading of 4 bytes at the same time, we ignore one column in 4 of the image, by reading only half of the luminance information (Fig. 9). Even though for the human eye the luminance is the component of a color that has more significance, this is not valid in the case of robotic vision. Moreover, using this approach we access 4 times less the memory.

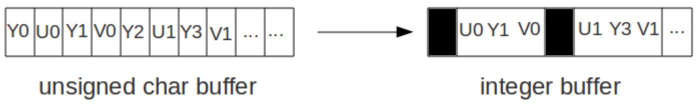


Figure 9: An illustration of the conversion of the unsigned char buffer to an integer one, allowing thus the reading of 4 bytes at the same time. Using this approach we can obtain a reduced resolution of the images.

6.2 Color segmentation and blob formation

Further image processing and analysis will be performed on the index image. Having the colors of interest labeled, scan lines are used for detecting transitions between two colors of interest.[11] For the vertical search in order to improve processing time only every second column is scanned while for the horizontal scan only every second row is scanned with the purpose of finding one of the colors of interest. For each scan line the initial and final point of the lines are saved. Both types of scan lines start in the upper left corner of the image and go along the width and the height, respectively, of the image. For every search line, pixels are ignored as long as they are not of the first color of interest. Once a pixel of the colors of interest is found, a counter of the pixels of the same color is incremented. When no more pixels of the first color are found, pixels of the second color of interest will be searched. If there are no pixels of the second color of interest, the scan line is ignored and a new scan line will be started in the next column/row. Otherwise, a counter of the pixels having the second color of interest will be incremented. Before validating the scan lines the values of the two counters are compared to a threshold. All the valid scan lines are saved and after their validation the next step of the processing pipe is the formation of blobs.

The notion of blob is different in the case of the two applications presented. In the case of humanoid soccer, transitions between green and white, green and orange, green and blue, green and yellow are searched. The information about the green color is used just for a validation that we are looking for the colors of interest only within the limits of the soccer field, thus diminishing the probability of taking into account false positives. Blobs are formed from validated neighbor scan lines that are parallel, taking into consideration only the pixels of one of the colors of interest. The mass center for each scan line, without including the run-length information about the green pixels,is calculated. By calculating the distance between the center of mass of consecutive scan lines we can decide whether or not they are parallel. If they are parallel and the distance between them is smaller than a predefined threshold the scan lines are considered as being part of the same blob and they are merged together.

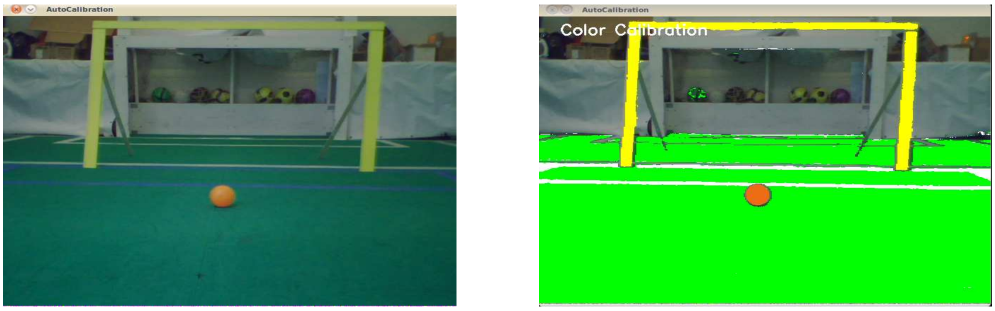
Having the blobs formed, several validation criteria are applied in the case of the orange ball and of the blue or yellow goals, respectively. In order to be considered a yellow goal, a yellow blob has to have the size larger than a predefined number of pixels. In the situation in which the robot sees both posts of the goals, the middle point of the distance between the two posts is marked as the point of interest for the robot. In the case when just one of the posts is seen, its mass center is marked. For the validation of the ball, the area of the orange blobs are calculated and the blob validated as being the ball will be the one that has the area over a predefined minimum value and it is closest to the robot. In order to calculate the distance between the robot and the orange blobs without having an estimation of the pose of the robot, the center of mass of the robot is considered to be the center of the image.

For the vision system of the Bioloid robot, transitions between yellow and pink, pink and yellow, pink and blue, blue and pink, orange and pink are searched for the detection of the posts and of the beacon. Also transitions between white and green are used for the detections of the walls of the maze which are to be avoided during the movements of the robot. Repeated experiments showed that an acceptable value for the threshold is 20 pixels. Clusters are formed from valid scan lines containing the same two colors of interest. The scan lines are grouped into clusters if they have the two colors of interest, in the same order and they are found at a distance of at most 50 pixels one from another. In this case, the clusters do not have the common meaning of a uniform region having a certain color, they stand for a region in the image having the sequence of two colors of interest. For each cluster, the area is calculated and in order to be validated as one of the posts, its area has to be in the range of [500,2000] pixels. For each valid cluster its mass center is computed. The size of the cluster is a good hint for the distance of the robot from the object. For the white-green transitions, clusters are not necessary and the information saved for further use is an array of scan lines containing transitions from white to green. The array of white-green transitions as well as the coordinates of the mass center for each post and for the beacon are then shared with the other modules that are responsible for computing the localization of the robot.

6.3 Results

In this subsection we present some images that show every steps of our algorithms for object detections: from acquiring a frame, calibrating the color of interest, forming the index image with all the colors of interest labeled, to color segmentation and detection of the objects of interest (in this case the objects of interest were the orange ball and the yellow goals).

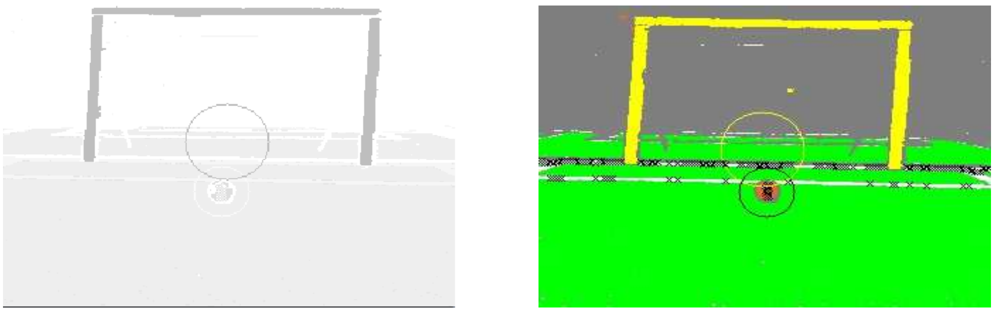
The first step is acquiring an image that can be displayed with the use of our ViewerClient application (Fig. 10(a)). Having an image acquired, we move on to classifying the colors of interest with the help of the CalibClient application, as it was previously described in Section 5.2. The result of the color classification can be seen in Fig. 10(b).



(a) (b)

Figure 10: On the left an image captured by the NAO camera. On the right, the same image with the colors of interest classified.

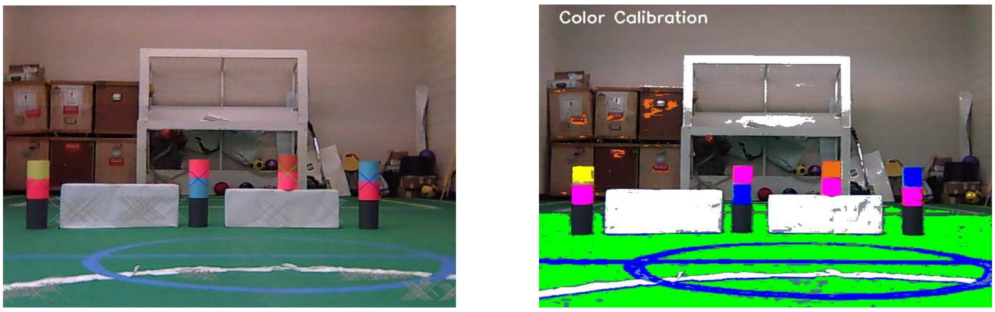
The next step of our algorithm, is the conversion of the YUV/RGB image into an index image. Figure 11(a) presents the index conversion of the previous frame while Figure 11(b) represents the equivalent “painted” image according to the labels in the grayscale image. The painted image is a 3-channels RGB image of the same resolution as the index image. The index image is scanned and for each pixel labeled as having one of the colors of interest, the color of the corresponding pixel in the RGB image is set as having the respective color of interest. If there are pixels that do not have any of the colors of interest they will be painted as gray. Both images already contain the markers that identify the objects of interest. The black circle stands for a valid ball while the yellow circle is a marker for the yellow goals. The yellow circle is constructed having the center in the middle of the distance between the two yellow goals. The black crosses are markers for the white lines of the field.



(a) (b)

Figure 11: On the left, the index image. On the right, the equivalent image “painted” according to the labels in the grayscale image.

Figure 12 shows similar results obtained using the Bioloid robot in the Micro Rato competition.



(a) (b)

Figure 12: On the left, the original image having a marker for each color blob detected and also a mark for the mass center of each post as well as for the walls. On the right, the color segmented image.

Figure 13(b) presents the processing times spent by the vision system that we are proposing, in a worst-case scenario. The low processing times were obtained using the NAO robot in a real soccer game and they are strongly influenced by the internal structure of the NAO robot. NAO comes equipped with only a single core processor of 500MHz and with 512MB of RAM memory. Even with these low processing capabilities we are able to use the camera at 30 fps while processing the images in real-time and achieving reliable results. The Bioloid robot is used with an IGEP board, based on a similar architecture as the one of NAO and running Ubuntu 10.04. The board is equipped with a DM37301000Mhz processor and 512MB RAM. The total processing time spent by the Bioloid architecture is on average, 98ms, thus allowing the use of the camera at 10fps. These results are also related to the fact that the webcam used is connected to the board through a USB hub which introduces delays that are remarkable especially in the process of acquiring an image. The results of the image processing algorithm are fast, each object of interest is being detected on average, in 2ms.

Acquiring an image	55ms
Conversion from RGB to index	30ms
Object detection - for each object of interest	2ms

Task performed	Time
Acquiring an image	1ms
Conversion from YUV to index	15ms
Orange detection	4ms
Yellow detection	2ms
Blue detection	2ms
White lines detection	4ms

(a) (b)

Figure 13: On the left, the processing times obtained with the Bioloid robot. On the right, a table with the processing times spent. The total processing time of a frame is 28ms, which allows us to use the camera at 30fps.

7. CONCLUSIONS AND FUTURE WORK

This paper presents a real-time reliable vision system for a humanoid robot. From calibrating the intrinsic parameters of the camera, to color classification and object detection the results presented prove the efficiency of our vision system. The main advantages of our approach is its modularity, which allows it to be used with a large number of different humanoid robots and the real-time capabilities allow us to use the camera at 30fps even with a low processor as the one used in the NAO robot. We presented an efficient and fast algorithm for self-calibration of the parameters of the camera which is extremely helpful for any vision system that aims at providing a reliable representation of the real world in images. Moreover, the algorithms for object detection based on color classification that we propose can be used in a wide range of real time applications for the detection of color-coded objects.

Future developments of our work include more validation criteria based on circular histograms and classifiers training which are more generic and are not color dependent. Also the algorithm for the self-calibration of the camera parameters will be improved in order to be used in real-time.

REFERENCES

[1] official website, R. `http://www.robocup.org`. Last visited June, 2011.

[2] www.microrato.ua.pt, "Rules of the Micro Rato competition," (2011).

[3] Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B., "The NAO humanoid: a combination of performance and affordability," *ArXiv e-prints* (July 2008).

[4] Neves, A., Lau, N., Reis, L., and Moreira, A., "Portuguese Team Team Description," (2011).

[5] Neves, A. J. R., Pinho, A. J., Martins, D. A., and Cunha, B., "An efficient omnidirectional vision system for soccer robots: from calibration to object detection," *Mechatronics* **21**, 399–410 (March 2011).

[6] Khandelwal, P., Hausknecht, M., Lee, J., Tian, A., and Stone, P., "Vision calibration and processing on a humanoid soccer robot," in [*The Fifth Workshop on Humanoid Soccer Robots at Humanoids 2010*], (December 2010).

[7] Lu, H., Zheng, Z., Liu, F., and Wang, X., "A robust object recognition method for soccer robots," in [*Proc. of the 7th World Congress on Intelligent Control and Automation*], (June 2008).

[8] Website, R. O. `http://www.robotis.com/xe/`. Last visited June, 2011.

[9] Caleiro, P. M. R., Neves, A. J. R., and Pinho, A. J., "Color-spaces and color segmentation for real-time object recognition in robotic applications," *Revista do DETUA* **4**, 940–945 (June 2007).

[10] Neves, A., Azevedo, J., B. Cunha, N. L., Silva, J., Santos, F., Corrente, G., Martins, D. A., Figueiredo, N., Pereira, A., Almeida, L., Lopes, L. S., and Pedreiras, P., [*CAMBADA soccer team: from robot architecture to multiagent coordination*], ch. 2, I-Tech Education and Publishing, Vienna, Austria (In Vladan Papic (Ed.), Robot Soccer, 2010).

[11] Neves, A. J. R., Martins, D. A., and Pinho, A. J., "A hybrid vision system for soccer robots using radial search lines," in [*Proc. of the 8th Conference on Autonomous Robot Systems and Competitions, Portuguese Robotics Open - ROBOTICA'2008*], 51–55 (April 2008).