# Embedded Distributed Vision System for Humanoid Soccer Robot

F. Blanes, P. Muñoz, M. Muñoz, J.E. Simó, J.O. Coronel, M. Albero

*Abstract*—**Computer vision is one of the most challenging applications in sensor systems since the signal is complex from spatial and logical point of view. Due to these characteristics vision applications require high computing resources, which makes them especially difficult to use in embedded systems, like mobile robots with reduced amount memory and computing power. In this work a distributed architecture for humanoid visual control is presented using specific nodes for vision processing cooperating with the main CPU to coordinate the movements of the exploring behaviours. This architecture provides additional computing resources in a reduced area, without disturbing tasks related with low level control (mainly kinematics) with the ones involving vision processing algorithms. The information is exchanged allowing to linking control loops between both nodes.**

*Index Terms*— **Humanoid robots, visual servoing control, distributed system, embedded computing.**

## I. Introduction

THE firsts experiments in artificial vision systems dates back to 1950 and many of the concepts currently considered essentials have been stated in 1980. Since then computer vision system is one of the most growing topics in the last years with applications in many fields like supervision in industry process, health, security, classification and recognizing, agriculture and robotics.

In all of these applications the final goal is to understand an image obtained using vision sensors (image-information transformation) going beyond the image treatment (image-image transformation).

In the case of robotics, vision systems are one of the most used sensor information sources and in the case of humanoid robots, by definition, is the main source for environment interpretation. In these robots there are many other problems to solve, mainly those related with locomotion stability issues that increase computing requirements. At the same time many other topics like human-machine interaction, robot cooperation, mission and behaviour control give to humanoid robot a higher level of complexity like no any other robots.

In this paper an active vision system is presented under the paradigm of distributed control architecture. This vision system provides high level information after image processing, and shares this information with other nodes using the Ethernet network.

Nowadays, many robot competitions are being used as benchmark test in robotics. This is the case of Robocup which encourages the developments of researchers in all the topics in humanoid robotics, applying the results in a robot soccer competition.

Robocup organization promotes, with the change of rules every year, to imitate in robots the human mechanism for perception. In this sense vision sensors are basic in this competition where active perception is more necessary every time due to limitations in the field of view to 180º.

## II. Developments in Humanoid Vision Systems

One of the most productive ways to analyze the behavior of the engineering designs are competitions because developments can be compared with those of other research groups and draw conclusions.

The annual RoboCup competition is a meeting point of many research groups coming from different part of the world. Participants test the new developed algorithms and designs against adversary teams focusing in a well know high level goal: "win the match". Since 1997, the development of humanoid robots, in particular, vision systems and their integration into control architectures has been a driving line to performance improvements.

In recent years one of the most active areas in the integration of interdisciplinary research efforts has been the humanoid robotics. The humanoid robots development provides a new shape of classical embedded systems challenges (such as stability, sensor network and integration of vision systems) while inspiring new challenges related to the natural goal of human imitation such as artificial intelligence, human-machine interaction and cooperation between robots and other cyber-physical systems. This is a very rich platform inspiring research in many cutting-edge technologies.

Vision systems are becoming the main source of sensory information in robotic systems. Talking about image processing, there are available developments reached in many application fields like medicine, astronomy, transportation, entertainment and, of course, robotics. This state-of-the-art in image processing should be the starting point of RoboCup image sensors.

The RoboCup Soccer League, which we intend to participate with the development proposed here, consists of a series of technical tests and soccer matches between robots. In order to make the environment images processing easier, the objects of interest (goals, ball, field and lines) are indicated by colours that can be clearly distinguished in the colour

F. Blanes, P. Muñoz, M. Muñoz, J.E. Simó, J.O Coronel and M. Albero are with the Universidad Politécnica de Valencia, Instituto de Automática e Informática Industrial.
E-mail: pblanes@ai2.upv.es

spectrum. The promoters claim the RoboCup humanoid robots get increasingly resemble a human, so they include restrictions on the design of the robot. In the field of vision, for example, they have restricted the field of view of cameras and limited the number of cameras to two.

For RoboCup 2010 edition, the proposals of video processing subsystems can be organized into three main categories attending to the attached hardware and the architecture integration capabilities.

- webcam attached to a general purpose Pocket Computer [1],[13],[22].
- webcam [4],[5],[10],[16],[17],[19],[21], CMOS [3],[6],[7],[9],[11],[12],[15],[20] or CCD [2],[18] cameras directly attached to the main board.
- CMOS or CCD cameras attached to a dedicated processor or micro-controller acting as a kind of intelligent camera [8],[14].

The vast majority of the groups present proposals using monoscopic vision (single camera), except for [11],[15] that present a stereo vision system (two cameras). Stereo vision systems intend to look like human vision and allow much more precise estimates of lengths and positions in 3D environments at the cost of increased use of resources.

Mainly inherited from general purpose vision systems, the basic steps in video analysis followed by all participating teams are depicted in Fig 1.

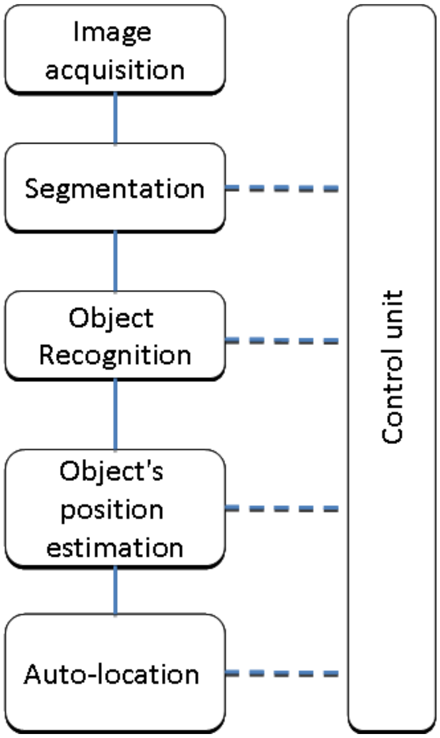


Fig. 1. 5 Image processing stages

All the groups use to follow this strategy (with different image resolutions and different colour spaces), but final implementation can differ in:

- Segmentation "image scoring" instead of the standard for colour thresholds [17]
- Adding some features as detecting extra lines [15], [16]
- Path planning of robots from the characterization of the environment [11]
- Identification of robots [13] and exchange of messages between robots for coordination and information fusion [19].

This paper presents a proposal for an embedded active vision system integrated into the distributed control architecture of the robot. The vision subsystem is an intelligent sensor capable of interpreting the environment and generate information to share with the rest of the control subsystems.

The used scenarios in this paper is a kind of RoboCup environment, but the proposed hardware and control architecture is designed to be applied to any changing environment and ensures the characterization of any real-time dynamic environment.

## III. ROBOT PLATFORM

A primary issue in robotics field is the environment perception and robot interaction. The configuration on their mechanical design, accuracy, low weight, etc., is a critical issue for biped robots where stability must be ensured in every moment. This goal must be present during the design process to obtain reliable robot architecture.

Taking on mind these goals, the final dimensions and mechanical proportions for MicroBIRO-II robot are based on a Robocup mid-size league rules. That means that MicroBIRO-II has a total height of 54 cm, and a total weight of 4 Kg. The system is self-contained, holding aboard all the necessary systems to achieve autonomous movements, control, and real world interaction.

MicroBIRO-II could operate up to 21 DOF. This number of DOF guarantees high mobility for the robot platform, and ensures a versatile locomotion. The DOF distribution that we can observe are six DOF per leg, one for the trunk motion, three in each arm, and two for the vision subsystem. This DOF distribution, enables a large number of different movements, like walking, kicking to a ball, getting up from floor, etc.

One of the main rules for robot design is the use of mechanical material that makes their structure lightweight, but strong, and without mechanical deformations. For these reasons some of the MicroBIRO-II mechanical pieces have been designed and mechanized under several constraints to obtain a 3D structure, with low weight and high rigid constraints. As Fig 2 shows, the main legs pieces structure have been designed in only one mechanized piece, without screws or unions. This design ensures us a high rigid structure, with a very low weight.

Humanoids design involves the need to obtain a mechanical structure with a human appearance, in order to operate into a human real world. MicroBIRO-II mechanical structure provides a final design with structural proportions similar to humans, achieving a human appearance, as illustrated in Fig 3.

Fig 3

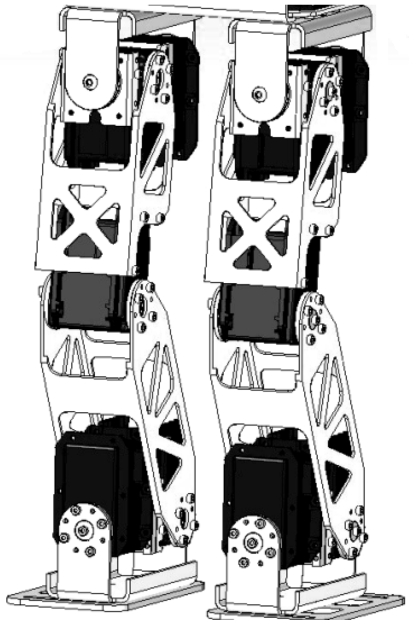


Fig. 2. MicroBIRO-II DoF legs distibution.

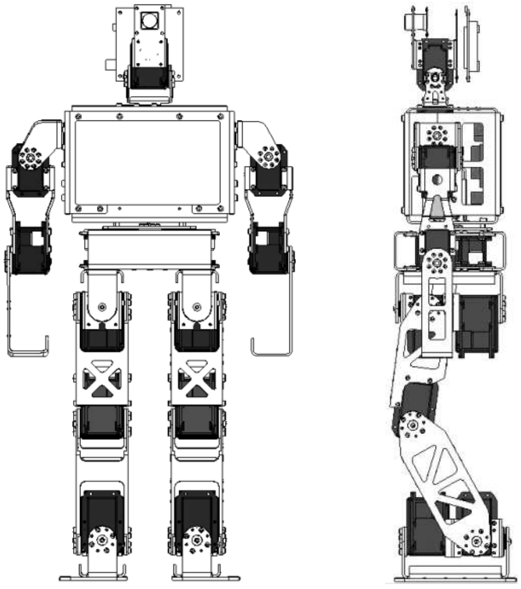


Fig. 3. MicroBIRO-II mechanical views.

Another important point of the design process is the correct selection of the actuators. The weight of the actuators is quite significant, as they add up to 60% of the total weight, but they should move without error all the joints in the robot, for this reason we need to use a high ratio weight/torque actuators. Another important issue on actuators features is their communication topology and bus expandability. This election can determine the main control period on the robot. In this way, Robotis servomotors with RS-485 bus topology communication have been used like actuators on the robot, achieving up to 36 Kg/cm of torque in some of them, and up to 1Mbps. Actually the internal MicroBIRO-II motor control network its capable to obtain a periodicity of 10msec, based on a write/read processes on all nodes.

MicroBIRO-II includes all the power systems in its platform in order to achieve autonomous operation. The power system is composed by the Li-Po batteries and the power supply control board. This system gives to MicroBIRO-II a total energy storage of 40 Watts-hour. This power system is divided into different voltage system: digital supply system (5V), high voltage drive supply (18V), middle voltage drive supply (12V) and low voltage drive supply (8V). Each one of this power supplies are designed for different servomotor models running on robot. To achieve each one of the voltage output subsystem and minimize power losses, the power control board has been designed using DC-DC converters, with high efficiency and power output.

## IV. HARDWARE

### A. Vision System

The vision system is supported on an embedded module. This autonomous device contains the necessary peripherals for communications with the robot through an RJ45-Ethernet connector, and the connection to the CMOS camera. It also has the power system in order to adjusting the different tension levels that this board needs. The computing capability of this embedded module is given by the core module CM-BF537E, that is part of the Blackfin family from Analog Devices. Concretely the processor DSP Blackfin 537. Their main capabilities are:

- Up to 600 MHz clock speed,
- 132 Kbytes of L1 memory peripherals access,
- 32 MB SDRAM,
- 4 MB Flash
- 25 Mhz crystal.

The sensor used in this module is an Omnivision OV7660 CMOS camera. The camera is connected to the embedded system through the SCCB (Serial Camera Control Bus).This sensor support different image resolution modes and formats.

The vision system is situated in the robot's head (Fig 3). It can adjust its orientation relative to the axis pitch and yaw. The vision system must be able to handle these actuators to set the camera in the correct orientation to track the objects.

### B. Main Control Unit

The main control unit of the robot is supported on another embedded module. The processor module integrates an XScale microcontroller, concretely the PXA320 processor type that outstands for its high performance and low power consumption. This board allows to the system to manage the SSP-SPI bus, USB, Ethernet or WiFi connection among others.

A novel approach is currently under development designing a new board that contains an FPGA mapped in the processor memory. This FPGA will multiplex the actuator bis RS-485 in 6 branches in order to manage each one of these in parallel. It is expected that this device will allow real-time management of communication buses, freeing up the main microprocessor from this difficult task. In this way to get the information, the processor reads directly as if it were an external memory. This new configuration allows to dispose all the information contained in the actuators in each one of the control cycles.

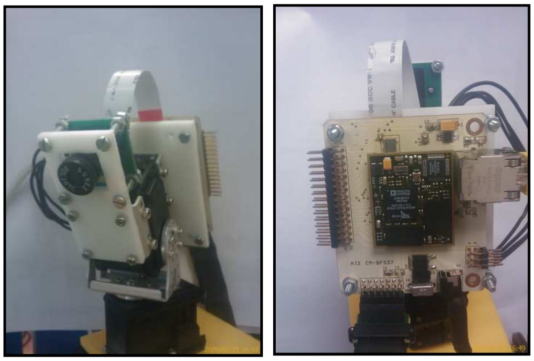


Fig. 4. Vision system Hadrware.

### C. Communication

In order to allow the information exchange between the vision system module and the main control unit, a messaging protocol based on TCP / IP has been implemented. The characterization of the environment of the robot detained after the image processing is sent to the control module, which is responsible for taking appropriate decisions according to the state of the system and the sensory input.

## V. SYSTEM ARCHITECTURE

The proposal is based on the integration of an embedded vision system into the distributed control architecture of a humanoid robot. A message protocol has been defined for the communication between the main control unit and the vision system.

The main control unit selects the task to be performed by the vision system among a predefined set of task. The information extracted by the vision system is shared with the main control unit and the rest of the distributed control system via Ethernet.

### A. Active vision system

The vision system is mounted with two servomotors, allowing movements with two degrees of freedom. The motors are connected directly to the DSP through its UART port using serial TTL communication. So, the vision system has the capability of deciding its own movements, becoming then an active vision system and receiving from the main control unit just higher level instructions.

The entire computation referred to the visual perception of the environment is carried out in the DSP. The application in the DSP has been implemented as a multithreading process.

The main thread takes over the camera and communication initialization and runs the acquisition and processing threads, which are synchronized using semaphore mechanisms.

The acquisition thread reads the image delivered by the camera sensor through the driver and stores it in a shared buffer. The processing thread picks the stored image and applies the algorithms to extract the desired information. This information is classified in a matrix, according to the messages protocol, and sent to the main control unit through the Ethernet communication.

Threads are implemented using the VisualDSP Kernel, a very light and robust operative system, provided with the development environment of the Blackfin DSP, which allows, among other things, to define threads, priorities, critical regions, semaphores and messages.

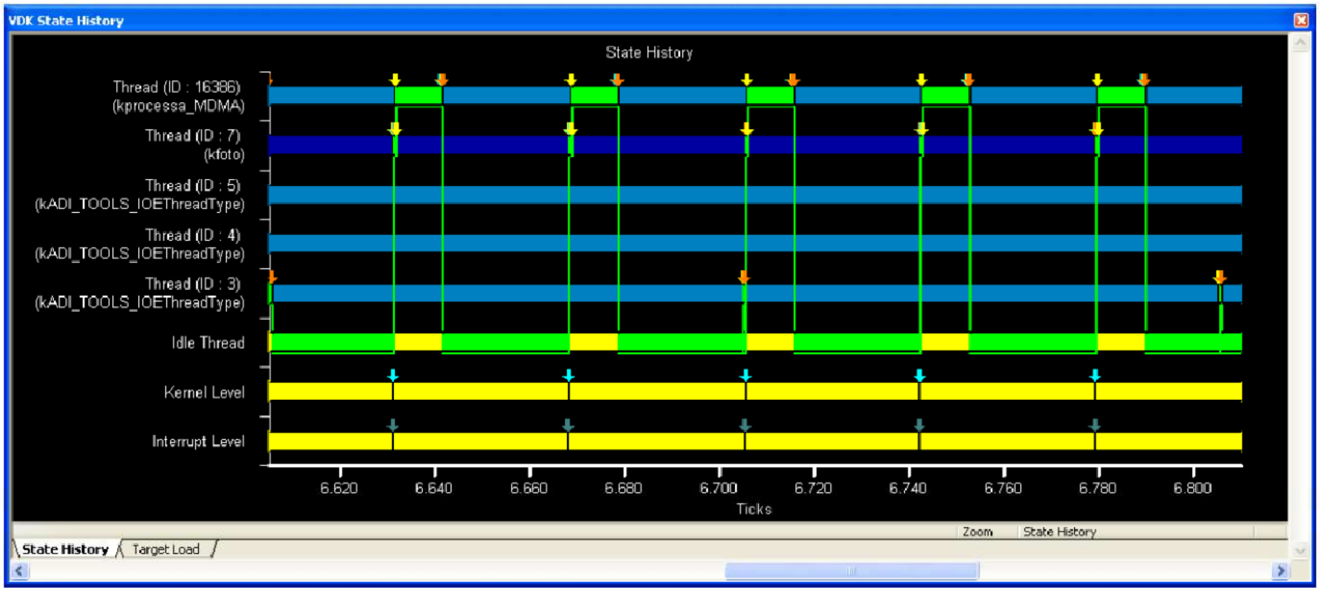


Fig. 5. Execution chronogram of the DSP application.

The computation time (CT) of the acquisition thread is minimal, since the image storage is carried out using DMA (*Direct Memory Access*) without any processor usage, while the CT of the processing thread depends on the environment characterization needs and the algorithms used.

The execution period of both threads is related to the CT of the processing thread and must be multiple of the camera frame rate.

Working on QVGA resolution (320 x 240 pixels), the frame rate is 30 fps (one image each 33 ms) and the CT processing thread is 10 ms, with the algorithms developed until the moment. Thus, the execution period would be 33 ms. If the CT processing thread would any time overcome the 33 ms, the execution period would increase to 66 ms (two times the frame rate).

So now, with QVGA resolution, one image is processed each 33 ms with a processor usage of 30 %. The execution chronogram is shown in Fig 5.

The vision system extracts the position of the objects of interest from each image, stores it in a structure and sends it to the main control unit each execution period. This shared structure, shown in Table 1, is composed of:

a) Position of the servomotors in the moment of taking the image and objects recognized in the image
b) Position of objects respect to the robot's centre of mass.

TABLE I
MESSAGE WITH POSITION AND ORIENTATION OF THE VISION MODULE

| struct DSP_VISION | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| struct OBJETCS [NO_OBJETCS] | | | | | | | | |
| SERVO_H_POS<br>[0,1024] | SERVO_V_POS<br>[0,1024] | update1 | ρ1 | φ1 | ... | upn | ρn | φn |
| 2 bytes | 2 bytes | 1 byte | 2 bytes | 2 bytes | | 1 byte | 2 bytes | 2 bytes |

TABLE II

| struct COLIBRI_VISION | | | | | | |
|---|---|---|---|---|---|---|
| ACTION | DATA | | | | | |
| 0x00: servomotors access<br>0x01: Behaviours | SERVO_H_POS<br>[0,1024] | SERVO_V_POS<br>[0,1024] | BEHAVIOURS<br>0x01: BALL<br>0x02: SCAN<br>0x03: IDLE | Z_HEAD<br>[0,1024] | OY_HEAD<br>[0,1024] | OZ_HEAD<br>[0,1024] |
| 1 byte | 2 bytes | 2 bytes | 1 byte | 2 bytes | 2 bytes | 2 bytes |

One of the behaviours performed by the vision system is the image-based object tracking. Based on the error between current and desired object coordinates in the image plane, the servomotors move the camera until the object is in the centre of the image.

### B. Main control unit

The control architecture of the robot is divided in 3 basic layers: mission, tactic and reactive.

- The mission layer is defined by behaviours, consisting of various basic skills, for example, look for the ball and go to it.
- The tactic layer is defined by basic skills, consisting of various trajectories, for example, walking.
- The reactive layer defines trajectories or articulations positions.

The vision system task can be selected from any of the layers, but it is intended to be done from the highest layers (mission or tactic). On the other hand, the vision system will maintain the last behaviour defined until a new one is received from the main control unit.

The main control unit decides, according to the state of execution and the information knowledge, the behaviour to be performed by the vision system.

The actions defined until now are:

- Main control unit access to the servomotors.
- Adopt a predefined behaviour (scan, tracking, idle).

The position and orientation of the vision system coordinates origin (under the two servomotor axes origin) respect to the centre of mass of the robot, solved by direct kinematics, is also sent to the vision system to calculate the distance and orientation of the objects.

All this information is classified into a structure, shown in Table 2, containing: action to be performed, servomotors desired position, behaviour, and position and orientation of the vision system coordinates origin.

It is intended that in a future, the main control unit, according to the needs and the available resources, will decide the type of processing to apply in terms of time and reliability.

## VI. Image Processing

The work developed until now in image processing covers the steps of image segmentation and object recognition (ball and goals).

### A. *Colour space and resolution*

The selected colour space of the image delivered by the sensor is YUV, which splits the pixels' information into components of illuminance (Y) and chrominance (U, V). The resolution adopted is QVGA (320 pixels width and 240 pixels height)

### B. *Image segmentation and pixel connectivity*

The goal of image segmentation techniques is to isolate pixels of the image following predefined criteria in order to simplify the later processing and analysis.

In our case, the segmentation criteria are based on colour, so the isolated pixels correspond to objects known in advance. From the original image, pixels whose colour is between previously defined thresholds are isolated. At the same time, contiguous pixels of the same colour are grouped. These pixels clusters are characterized with colour, number of pixels, width, height and centroid).

Five different connectivity algorithms were compared, being the *Run Length Encoding (RLE)* algorithm the one selected. The processing times of the algorithms, implemented in Matlab, are shown in Table 3.

The *RLE* algorithm implemented represents the segmented image as sequences of 1s or *runs*, characterized by the centre pixel coordinates and the number of pixels. The adjacent *runs* are grouped in object candidates.

TABLE III
SEGMENTATION AND CONNECTIVITY ALGORITHMS COMPARISON

| Algorithm | Execution time (s) |
|---|---|
| Adjacency matrixes | 1.32 |
| Flood fill approach | 1.96 |
| Equivalence class resolution | 0.98 |
| Union find | 1.04 |
| Run length encoding | 0.90 |

```
candidates run_lenght_encoding()
for each row in image

    sequence of same colour pixels
    run = {colour,no_pixels,row,centre}
    if (run is adjacent to candidate and
        has the same colour)
            update candidate
    else
            create candidate = {colour,
            centroid, height, width}
    end
end
return candidates;
```

*Algorithm 1. Run length encoding algorithm*

### C. *Objects recognition*

Once characterized the grouping pixels with colour, width, height and centroid, shape algorithms are applied in order to identify the predefined objects (ball and goals)

```
centroid object_recog(candidates, object)

for each object candidate

  filtered_candidates =
   shape_filter (candidate)

  decision(filtered_candidates)

  if (decision == ok)
    return candidate.centroid
  end

end
```

*Algorithm 2. Objects recognition*

### D. *Distance and orientation estimation*

To have a good estimation, the object must be in the centre of the image, i.e. it must be tracked.

Once there, the distance and orientation are calculated, according to the neck's origin position, the current neck's servomotors position and the position of the camera respect to the origin resulting of the design.

### E. *Information for the main control unit*

The recognized objects, with their distance and orientation estimation, are organized in a matrix and sent to the main control unit.

## VII. Calibration Tool

During the normal operation, it is very useful to have a tool that allows the user to modify the vision system performance, without modify and reload the robot software.

In this way, the system must allow to define the different objects that the robot has to identify in the environment, as well as the image features.

The need of this tool becomes evident during the RoboCup. In this robot competition, before each match, an adjustment of the vision system is required to prevent the changes, mainly in lighting, that suffers the play field. The fact of having this application allows the user to adapt quickly to changing conditions. Otherwise the robot may not be able to distinguish some of the objects (ball, goal, lines), with behaviour problems that these issues entail.

Attending to this, a PC application that can connect directly to the robot vision system to use the following capabilities:

- Adjust the settings of the camera to the lighting and environment conditions.
- Calibrate the colours of objects depending on lighting conditions and object types.
- Allow processing of images captured or not by the robot in order to show in the screen the extracted information.
- Test and Debug the new image processing algorithms.

The procedure to be followed when introducing the robot in a new environment, for instance a new field, would be:

- First, you should adjust the camera settings. For this, the application allows to select auto exposure mode, auto white balance mode, and all those image parameters that offers the camera. Otherwise, user can set these manually based on the experience and expertise of the programmer. For this task, the GUI offers a friendly interface with scroll bars, allowing to save and reload sets of parameters.
- After this image adjustment has been done, ranges corresponding to the colours of different objects are defined in order to be used in segmentation algorithms. This process is made easily by clicking on the objects of interest in the image taken by the robot. The application automatically extracts the characteristics of the clicked pixel, and uses these as a seed to extrapolate these features to similar objects.

To manage these definitions, a procedure that allow you to take a picture through the robot camera and directly assign each interest pixel to the corresponding object has been dedined. After that, the application extracts the properties of these pixels and redefines the object colour ranges. To perform the pixel selection properly a set of tools as such as zoom have been developed and implemented.

The graphical interface is shown in Fig 6, in witch can be appreciated the picture before and after the information extraction process. In the second one, the detected objects have been highlighted the detected objects. In the bottom one, the set of controls to parameterize the image is shown.

## VIII. RESULTS

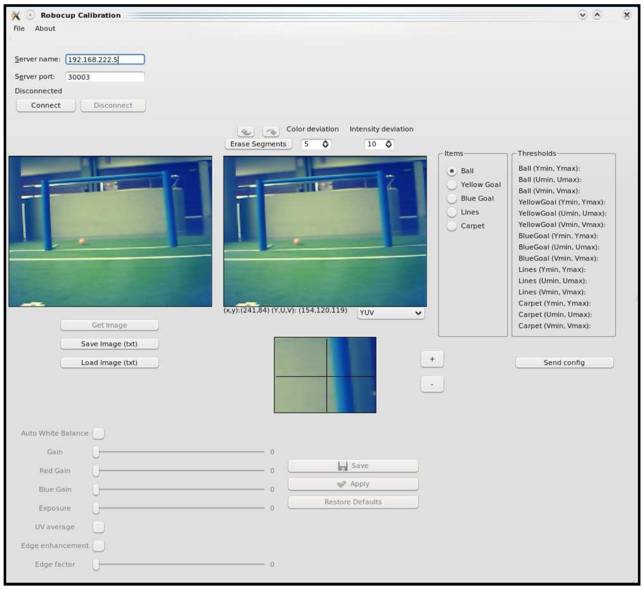
### A. Image segmentation and connectivity of pixels

Fig 7 shows several segmented images. The pixels whose colour matches the colour of objects to identify are the segmented pixels. The objects to identify are the ball and the goal.

The group of pixels to identify the objects are characterized by means of the connectivity. Additionally, it also removes groups that do not contain a minimum number of pixels.

### B. Identifying objects

The objects are identified from the characterized group of pixels. When the object is identified, a message is built which is transmitted to the control module. This message contains the position of all the objects identified. The position is transmitted in image coordinates. The results are shown in figure 7.


Robocup Calibration

Server name: 192.168.222.5

Server port: 30003

Disconnected

Color deviation: 5

Intensity deviation: 10

Items: ● Ball ○ Yellow Goal ○ Blue Goal ○ Lines ○ Carpet

(x,y):(241,84) (Y,U,V): (154,120,119) YUV

Thresholds
Ball (Ymin, Ymax):
Ball (Umin, Umax):
Ball (Vmin, Vmax):
YellowGoal (Ymin, Ymax):
YellowGoal (Umin, Umax):
YellowGoal (Vmin, Vmax):
BlueGoal (Ymin, Ymax):
BlueGoal (Umin, Umax):
BlueGoal (Vmin, Vmax):
Lines (Ymin, Ymax):
Lines (Umin, Umax):
Lines (Vmin, Vmax):
Carpet (Ymin, Ymax):
Carpet (Umin, Umax):
Carpet (Vmin, Vmax):

Auto White Balance □
Gain 0
Red Gain 0
Blue Gain 0
Exposure 0
UV average □
Edge enhancement □
Edge factor 0

Fig. 6. Graphical Calibration Tool.

## IX. CONCLUSION AND FUTURE WORK

In this paper a humanoid robot design and the developed work has been presented focusing on the embedded vision system for environment recognition. This is a long term project where many engineering areas converge (mainly mechanical engineering, electronics, and computer architecture) in a complex system which has been designed for Robocup competitions. This application oriented approach do not subtract to developments applicability for other scenarios.

In the next future we will focus on completing the basic steps shown in figure 1, i.e., the estimation of the location of other objects (apart from ball and goal) with respect to the robot and the self-localization of the robot in the environment.

Furthermore, we will work on developing of line detection algorithms to combine with the location information of the fixed objects such as goals. This will improve the robot self-location ability.

In addition, we will study the possibility of incorporating stereoscopic vision on the robots. This approach has not been use in small humanoid robots, but currently image processing dedicated embedded microcontrollers allow to expand computing power with reduced size. The effort must be done in the architecture design following distributed control principles. Finally the information fusion and sharing among robots will also be considered using communication capabilities.

## ACKNOWLEDGMENT

This work was supported from the Spanish MICINN project SIDIRELI DPI2008-06737-C02-01/02 and FEDER founds.

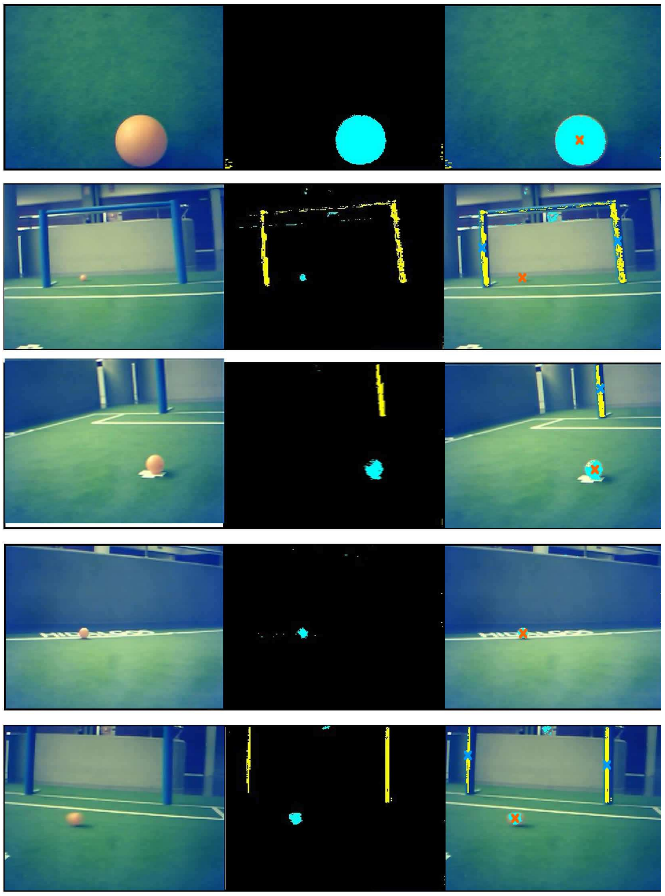


Fig. 7. Segmentation and identification algorithms of objects.

## References

[1] Tzuu-Hseng S. Li et al "aiRobots: Team Description for Humanoid KidSize League of RoboCup 2010". Workshop Robocup Singapore 2010.

[2] Chokchai Pengyasa et al. "Team BSRU-I: Team Description Paper". Workshop Robocup Singapore 2010.

[3] Aphilux Buathong et al. "Chibi Dragon Team Description Paper". Workshop Robocup Singapore 2010.

[4] Luis F Lupián et al. "Cyberlords RoboCup 2010 Humanoid KidSize Team Description Paper". Workshop Robocup Singapore 2010.

[5] M. Friedmann et al. "Darmstadt Dribblers. Team Description Paper for Humanoid KidSize League of RoboCup 2010". Workshop Robocup Singapore 2010.

[6] Bennet Fischer et al. "FUmanoid Team Description Paper 2010. Workshop Robocup Singapore 2010.

[7] Lim Sock Lip et al. "Team NYP Lions: Team Description Paper". Workshop Robocup Singapore 2010.

[8] Roberto Carlos Ramírez Márquez et al. "PIONEROS MEXICO Team Description Paper ROBOCUP 2010 Singapore". Workshop Robocup Singapore 2010.

[9] Guangnan Ye et al. "PKU-SHRC Team Description for RoboCup 2010". Workshop Robocup Singapore 2010.

[10] Buck Sin Ng et al. "Robo-Erectus Jr-2010 KidSize Team Description Paper". Workshop Robocup Singapore 2010.

[11] Keith Sullivan et al. "RoboPatriots: George Mason University 2010 RoboCup Team". Workshop Robocup Singapore 2010.

[12] Shohei Takesako et al. "SitiK KIT. Team Description for the Humanoid KidSize League of RoboCup 2010". Workshop Robocup Singapore 2010.

[13] S. Hamidreza Mohades Kasaei et al. "Persia Humanoid Robot. Team Description Paper 2010". Workshop Robocup Singapore 2010.

[14] Guillermo Villarreal-Pulido et al. "Bogobots-TecMTY humanoid kid-size team 2010". Workshop Robocup Singapore 2010.

[15] R. Gerndt et al. "WF Wolves KidSize Team Description RoboCup 2010". Workshop Robocup Singapore 2010.

[16] Jaekweon Han et al. "Team DARwIn. Team Description for Humanoid KidSize League of RoboCup 2010". Workshop Robocup Singapore 2010.

[17] Soo Theng Koay et al. "Team Description 2010 for Team RO-PE". Workshop Robocup Singapore 2010.

[18] Thavida Maneewarn et al. "Team KMUTT: Team Description Paper". Workshop Robocup Singapore 2010.

[19] Tang Qing "ZJUDancer Team Description Paper". Workshop Robocup Singapore 2010.

[20] Ching-Chang Wong et al. "Humanoid Soccer Robot Design by TKU Team for Humanoid League of RoboCup 2010". Workshop Robocup Singapore 2010.

[21] Chung-Hsien Kuo et al. "Team Description Paper: HuroEvolution Humanoid Robot for RoboCup 2010 Humanoid League". Workshop Robocup Singapore 2010.

[22] Javier Testart et al. "UChile RoadRunners 2010 Team Description Paper". Workshop Robocup Singapore 2010.