Embedding the AnaFocus' Eye-RIS Vision System in Roving Robots to enhance the Action-oriented Perception

Luis Alba Soto[a], Sergio Morillas[a], Juan Listán[a], Amanda Jiménez[a], Paolo Arena[b], Luca Patané[b], Sebastiano De Fiore[b]

[a]AnaFocus (Innovaciones Microelectrónicas S.L.), Av. Isaac Newton 4, Pab. Italia, Planta 7, PT Isla de la Cartuja, 41092 Sevilla (Spain);

[b]Department of Electrical, Electronic and System Engineering, University of Catania, I-95125 Catania, Italy, e-mail: [parena, lpatane]@diees.unict.it

ABSTRACT

This paper aims to describe how the AnaFocus' Eye-RIS family of vision vystems has been successfully embedded within the roving robots developed under the framework of SPARK and SPARK II European projects to solve the action-oriented perception problem in real time. Indeed, the Eye-RIS family is a set of vision systems which are conceived for single-chip integration using CMOS technologies. The Eye-RIS systems employ a bio-inspired architecture where image acquisition and processing are truly intermingled and the processing itself is carried out in two steps. At the first step, processing is fully *parallel* owing to the concourse of dedicated circuit structures which are integrated close to the sensors. These structures handle basically analog information. At the second step, processing is realized on digitally-coded information data by means of digital processors. On the other hand, SPARK I and SPARK II are European research projects which goal is to develop completely new sensing-perceiving-moving artefacts inspired by the basic principles of living systems and based on the concept of "selforganization". As a result, its low-power consumption together with its huge image-processing capabilities makes the Eye-RIS vision system a suitable choice to be embedded within the roving robots developed under the framework of SPARK projects and to implement in real time the resulting mathematical models for action-oriented perception.

Keywords: Eye-RIS, Vision, SPARK, SIS, AnaFocus, Robot

1. INTRODUCTION

The basic principles guiding sensing, perception and action in bio systems are relying on highly organized spatial-temporal dynamics. In fact, all biological senses, (visual, hearing, tactile, etc.) process signals coming from different parts distributed in space and also show a complex non linear time dynamics. As an example, mammalian retina performs a parallel representation of the visual world embodied into layers, each of which represents a particular detail of the scene. These results clearly state that visual perception starts at the level of the retina, and is not related uniquely to the higher brain centers.

Also motion in living systems is derived from highly organized neural networks driving limbs and other parts of the body, in response to sensory inputs. In locomotion, different and differently specialized neuronal assemblies behave in a self-organized fashion in such a way that, as a consequence of certain stimuli, particular patterns of neural activity arise, which are suitably sent to peripheral fibers to generate rhythmic activities of leg motion and control. Recent results have also shown that nonlinear and complex dynamics in cellular circuits and systems can efficiently model both sensing and locomotion. After analyzing and being involved in such topics, researchers from various scientific fields came up to the same idea to glue their effort to try to go up from sensing and locomotion modeling, to perception.

Hence, the aim of the SPARK projects is to develop, evaluate, optimize and generalize an insect brain inspired computational model. This is a completely new architecture for action-oriented perception, inspired by the basic principles of information processing by living systems and based on the concept of "self-organization".

In order to reliably undertake the algorithms resulting from that computational model, in particular, those tasks involving visual stimuli, the AnaFocus' Eye-RIS vision system has been used. As it will be overview throughout the article, the Eye-RIS vision system has shown as a suitable platform to successfully cope with such computational load in real time.

In the following, the main objectives of the SPARK projects as well as the AnaFocus' bio-inspired vision system technology will be outlined. Finally, the visual routines that have been implemented and optimized to run within the Eye-RIS system and the algorithms, based on these routines, able to extract in real time different kinds of visual details useful for cognitive purposes will be described.

2. EYE-RIS VISION SYSTEM

2.1 Introduction to AnaFocus' bio-inspired Smart Image Sensor (SIS) technology

AnaFocus' Smart Image Sensor technology (SIS) goes a step beyond in the exploitation of conventional CMOS imagers processing capabilities. SIS devices are not just CMOS Image Sensors but true vision devices. In a single chip they contain all the structures needed for:

- ✓ Capturing (sensing) images,
- ✓ Enhancing the sensor operation,
- ✓ Performing spatio-temporal processes on the image flow,
- ✓ Interpreting the information contained in the image flow, and,
- ✓ Supporting decision-making based on the outcome of such interpretation.

AnaFocus' SIS devices can be connected to either microprocessors or DSPs to define powerful autonomous Vision Systems. Some of the key features that SIS technology provides vision systems with are:

- ✓ High-dynamic range image acquisition.
- ✓ Truly mixed-signal architecture, with processing carried out by following a hierarchical flow. At early stages, processing is made by an array of mixed-signal programmable sensor-processors at the SIS. At later stages, processing tasks are undertaken by digital processors.
- ✓ Huge computational power with low-power consumption. Such computational power can be exploited to handle, depending on the application, several thousands of frames per second with medium spatial resolution.
- ✓ General-purpose, all-in-one architecture including optical sensors, processors, memories, data conversion, control and communication peripherals.
- ✓ Large operational flexibility. Vision chips are C and C++ programmable to meet specific customer applications.

SIS technology is employed in the so-called AnaFocus' Eye-RIS family of vision systems. These systems are the perfect choice for those applications in which the sensing-perception-action flow has to travel in real-time and whenever system compactness, low-cost and low-power are mandatory. As we will see, this approach perfectly suits the case of roving robots presented later on.

2.2 AnaFocus vision system architecture. A differentiated technology

As stated before, vision systems are different from cameras and/or imagers. Although vision devices can operate as cameras, their functionality goes well beyond. The function of a camera is to *acquire* (sense) images. A vision device not only acquires but also processes the image flow in space and time to extract the desired information contained in such a flow.

In a conventional vision system, conversion to the digital domain occurs right after the sensors. Hence, all processing is carried out in the digital domain. Such way of processing is actually quite different from what is observed in *natural vision systems*, where processing happens already at the sensor (*the retina*), and the data are largely compressed as they travel from the retina up to the visual cortex. Also, processing in retinas is realized in *topographic* manner; i.e. through the concourse of structures which are *spatially distributed* into arrangements similar to those of the sensors and which operate *concurrently* with the sensors themselves.

AnaFocus' Eye-RIS systems borrow from nature these architectural concepts, employing a different strategy in which image-processing is accomplished following a hierarchical approach with two main levels:

- ✓ *Early-processing*. This level comes right after signal acquisition. Inputs are full-resolution images. It means huge amounts of data. Much of this data is redundant and therefore useless for the specific image processing task to be accomplished. The basic tasks at this level are meant to extract the useful information from the input image flow. Outputs of this level are reduced sets of data comprising image features such as object locations, shapes, speed, etc. In many cases these outputs consist of binary data.
- ✓ *Post-processing*. Here the amount of data is significantly smaller. Inputs are abstract entities in many cases, and tasks are meant to output complex decisions and to support action-taking. These tasks may involve complex algorithms within long and involved computational flows and may require greater accuracy than early processing.

The architecture of Eye-RIS vision systems is conceived to optimally implement such image-processing splitting into early and post-processing, achieving compact implementations capable of providing maximum overall speed with minimum power consumption. Eye-RIS vision systems have the following features:

- ✓ The border between analog data (those provided by the sensors) and digital codes is not right after the image sensor. Instead it is placed after early processing.
- ✓ *Smart Image Sensors* are employed for image acquisition and early processing. These devices contain one processor per pixel which guarantees ultra fast parallel operation, as needed to handle the huge data volumes encountered at this stage.
- ✓ Conventional digital processing is employed for post-processing.

Figure 1 illustrates the shifting of the analog/digital border that is intrinsic to such vision systems architecture. This is the specific architecture used in the Eye-RIS family of AnaFocus vision products.

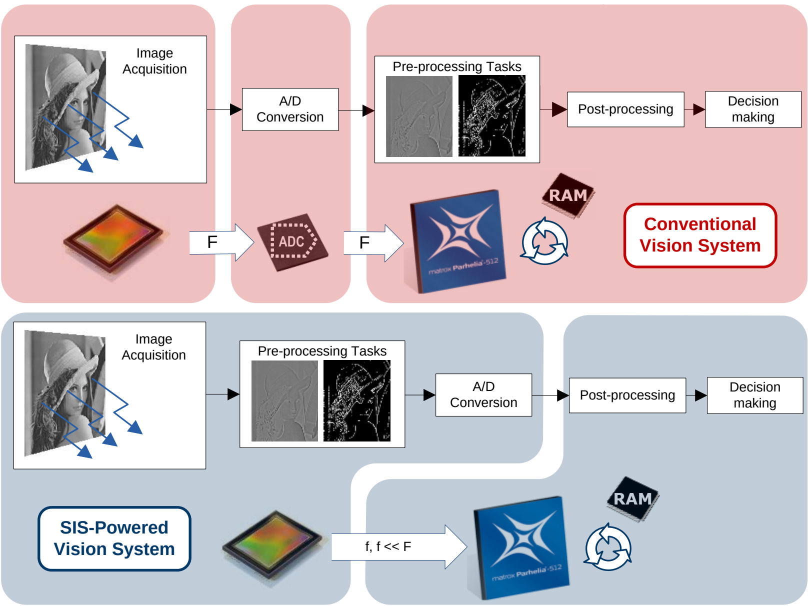


Fig. 1. Vision system approaches: Conventional vs. Smart Image Sensor

3. SPARK PROJECTS

3.1 Objetives

SPARK II is based on research undertaken in project SPARK I. In SPARK I, the main scientific objective was to formulate a new methodology for artificial cognitive systems, based on spatial-temporal patterns, able to process stimuli coming from sensors and directly influencing the motor actions. Inspiration was drawn from studies of animals, and in particular insects. We validated the approach on robot platforms equipped with specific hardware structures.

SPARK I resulted inter alia in a rough and still incomplete architecture of an insect brain [1]. Completing it requires the multidisciplinary expertise (mathematics, computational neuroscience, robotics, neurobiology / genetics) that will be mustered in SPARK II. These issues were address through:

a) **The design of a complete insect brain computational model**: In the envisaged model, complex nonlinear spatial temporal systems are used together with biologically plausible models of multi-sensory loops, for action-oriented perception.

   The framework for action-oriented perception is conceived as a hierarchical structure: it is divided into functional blocks, acting either at the same layer, or on different layers (parallel perceptual processes). Vertical hierarchy must coexist in the control architecture.

b) **The design and realization and realization of new software/hardware structures for the real-time implementation of the model:** This objective is strictly dependent on the previous one. Besides introducing a new insect brain inspired architecture, the project aims at modeling it and implementing it in a software environment, where all the capabilities and interactions will be assessed and optimized. This phase is crucial to join together the high level representation layer with the MB and CX model, and to efficiently exploit the underlying basic behaviours.

   Moreover, a major effort is envisaged in this phase towards the generalization of the applicability of the computational model to different test beds, as outlined in objective c) below.

   The project also aims at implementing the model in a hardware structure working in real-time. The hardware design will be defined on the basis of the overall architecture, together with the type and number of different sensors and actuators for the robots.

c) **The application of the model to different robotic test beds, with a view to demonstrating the emergence of new high-level sensory-motor and cognitive capabilities**: As already outlined, the new insect brain inspired computational model for action-oriented perception can be considered a general approach to perception in robots.

   From this point of view, the most direct way to show the generality of the approach consists in applying it to different test beds.

   Therefore, the use of different biologically inspired robotic structures is envisaged, to wit:

   - Legged machines, built on the basis of the already developed Gregor robot of SPARK I, and developing towards new types, like quadrupeds, and bipeds;
   - Wheeled machines, including more sensors and basic capabilities to be used as basic pre/proto-cognitive behaviours;

In order to demonstrate the model's applicability and generality, as well as its ability to lead to the emergence of new, environmentally mediated complex behaviours, we will design and build real life cluttered environments as test arenas (resembling for instance natural disaster areas). These scenarios will be set up in stages, reflecting the demand for increasingly complex behaviour.

4. VISUAL ALGORITHMS FOR COGNITION

4.1 Introduction

During the SPARK projects, several experiments involving roving robots, which goal was to validate the proposed insect brain computational model, were performed.

In particular, for a roving robot to successfully perform a navigation task, the control loop must be able to process the different stimuli of the environment in a time that must be compatible to real time applications. For a robot in the real world, the ability to interpret information coming from the environment is crucial, both for its survival and for reaching its target. The real world differs from structured environments because it is dynamic, so that it is impossible to fully program the robot’s behavior only on the basis of a priori knowledge.

In order to apply the proposed algorithms for action-oriented perception in robotic platforms, a complete hardware architecture has been designed and built. The core of the system is the SPARK board, an FPGA-based hardware designed to fulfill the requirements needed by the SPARK cognitive architecture. Besides the SPARK board, a distributed sensory system has been designed to be embedded on wheeled and legged robots. As a main part of that distributed sensory system, the Eye-RIS vision system has been selected. As it will be shown, thanks to the concourse of the SIS Q-Eye, the Eye-RIS vision system boosts the performance of a conventional vision system and is able to successfully accomplish the vision processing tasks involved within the robot behavior. This significant speedup enables real-time image processing with the system even in case of complex tasks. The functionalities of the Eye-RIS vision system enable the implementation of several advanced visual routines.

Nowadays, autonomous robots are used in a growing number of applications. A new FPGA based architecture explicitly designed to deal with a distributed sensory system for cognitive experiments is presented. The characteristics that make an FPGA-based hardware an optimal solution for our purposes are the flexibility of a reprogrammable hardware and the high computational power obtained with a parallel processing [2]. So the proposed system can independently manage several sensors with different frequencies. Dedicated channels are created for the sensors that require a high band for data transfer (visual system). In order to optimize the FPGA resources, a serial bus is used for the other sensors.

The computational power has also been considered to guarantee the implementation of the whole cognitive architecture.

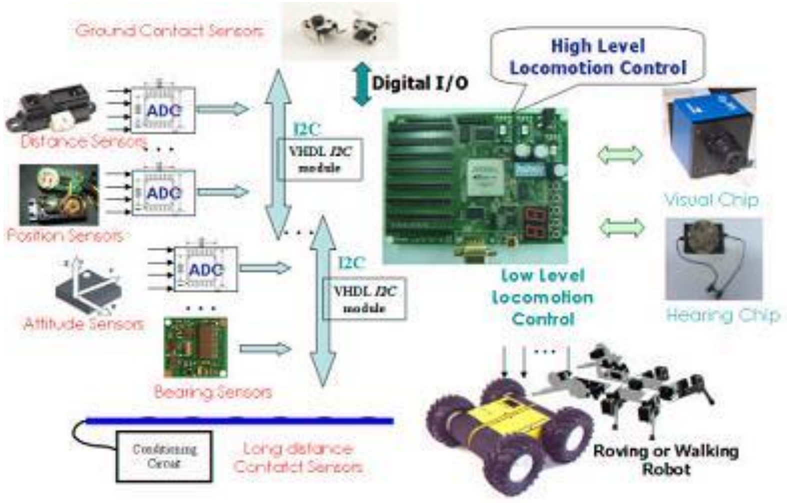


Fig. 2. Robot's central processing unit and sensory system block diagram

4.2 ***Robotic platforms***

Throughout the present section, a series of wheeled and legged robots are described and applied as test beds for the proposed action-oriented perception algorithms. The cognitive architecture has been experimentally tested at multiple levels of complexity in different robots. Details on the robotic platforms are given together with a description of the experimental results that include multimedia materials collected in the project web page [3].

The control architecture of both rovers consists in a low level layer based on microcontrollers that handle the motor system and a high level layer that includes the Eye-RIS visual system as a main controller together with the SPARK board. The rovers are also endowed with a suite of different sensors and can be interfaced with a PC through a wireless communication link.

4.3 ***Rover II wheeled robot***

Rover II, shown in Fig. 3. is equipped with a bluetooth telemetry module, four infrared short distance sensors, four infrared long distance sensors, a digital compass, a low level target detection system, an hearing board for cricket chirp recognition and with the Eye-RIS v1.2 visual system.

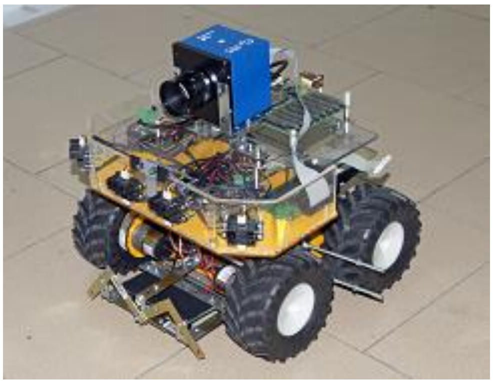


Fig. 3. Rover II wheeled robot

The low level control of the motors and the sensor handling are realized through a microcontroller. This choice optimizes the motor control performances of the robot maintaining in the SPARK board and in the Eye-RIS visual system the high level cognitive algorithms. Moreover Rover II can be easily interfaced with a PC through a bluetooth module: this remote control configuration allows performing some preliminary tests debugging the results directly on the PC.

4.4 Gregor III hexapod robot

The robot, as shown in Fig. 4, is equipped with a distributed sensory system. The robot’s head contains the Eye-RIS v1.2 visual processor, the cricket inspired hearing circuit and a pair of antennae. A compass sensor and an accelerometer are also embedded in the robot together with four infrared distance sensors used to localize obstacles. A set of distributed tactile sensors (i.e. contact switches) is placed in each robot leg to monitor the ground contact and to detect when a leg hits with an obstacle.

The core of the control architecture is the SPARK board. The robot sensory system is handled with an ADC board that is addressed by using an I2C bus, whereas the Eye-RIS v1.2 is interfaced with the main board through a dedicated parallel bus.

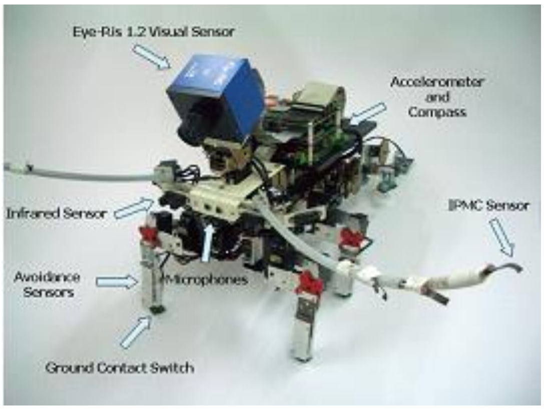


Fig. 4. Gregor III legged robot

4.5 Eye-RIS vision system software routines

A set of vision processing functions has been developed in order to build higher level applications mimicking insect behaviours. Such algorithms are extracted from the insect brain computational model.

Global displacement calculation

This function calculates the optical flow on an image sequence. The optical flow calculates local displacements of image segments on two consecutive grayscale or binary frames of an image flow. The displacement calculation is made of two steps. In the first step, two projections of the image are calculated, one for the X, and one for the Y axis. Each of these projections makes possible the calculation of the displacement along one axis. This means, that we do not have to make full search, to find the matching shift position, rather we have to do search along each axis only. Thus, the execution time is proportional to the search window border length, and not with its area, hence it is not quadratic. The projection calculation for an image depends on the window size and the number of calculated points. In navigation algorithms, it is very typical to calculate it only in a few dozens of search positions.

Examples of the operation can be seen in Fig. 5. In our example, there was a diagonal camera motion between the two consecutive frames. The algorithm correctly identified the egomotion of the camera in those search positions, where any contoured pattern was found. In those search position, where the pattern was vertical only, the horizontal component of the displacement was calculated only, because the vertical component cannot be calculated theoretically. In those locations, where the images did not contain any patter, the algorithm obviously could not identify the displacements.

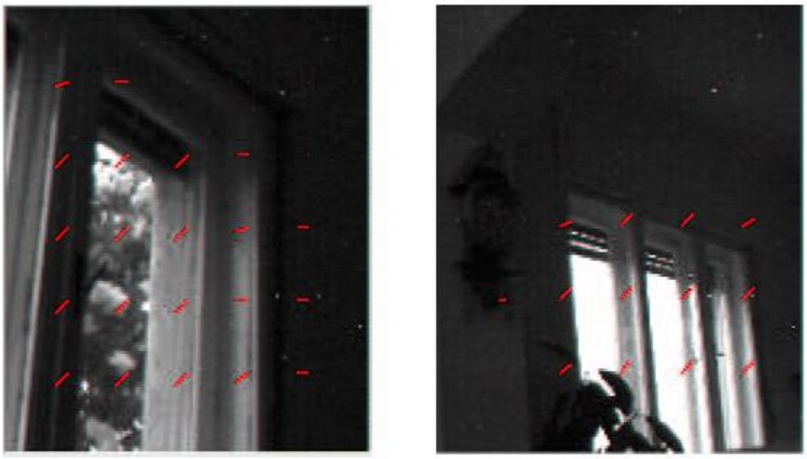


Fig. 5. Screenshots of the displacement algorithm

Foreground-background separation based segmentation

A possible way to extract the scene background from an image flow can based on the idea, that the background changes more slowly, than things we want to extract as foreground. This means, that if we consider the past of each pixel, we should find the background value(s) more frequently than the values belonging to moving or quickly changing objects.

Based on this assumption, two versions of the foreground-background separation algorithm have been implemented. One is based on temporal features of the image flow, while the second is based on spatial-temporal features. The input of these algorithms is the captured raw images, while the output is a binary marker of the moving objects.

Both of them assume stationary platform (non moving camera). The first algorithm learns the background in roughly 8 frames, while the second algorithm can be applied for consecutive frames also. Both of the algorithms can adapt to the slow changes in the background, which might be caused of the slow changes of the illumination. This means that the moving of the sun or the clouds do not causes any false classification. Moreover, both algorithms can adapt to quick changes of the illumination (e.g. switching on the light in a room), however, it might causes partial misclassification in a few consecutive frames, while the algorithm adapts to the new background. Both algorithms apply some binary morphological post processing, to clean the resulting marker.

- ✓ **Temporal Foreground-background separation**: This algorithm builds a background model, which is maintained on the Q-Eye. All the captured images are compared to it and those areas, where the differences are above a certain threshold are considered to be the moving object.

  The adaptation speed can be through the modification of the background update frequency. If the adaptation is too slow, it reacts slowly to the changing background. When it is too fast, it might adapt to the slowly moving objects, which might partially disappear.

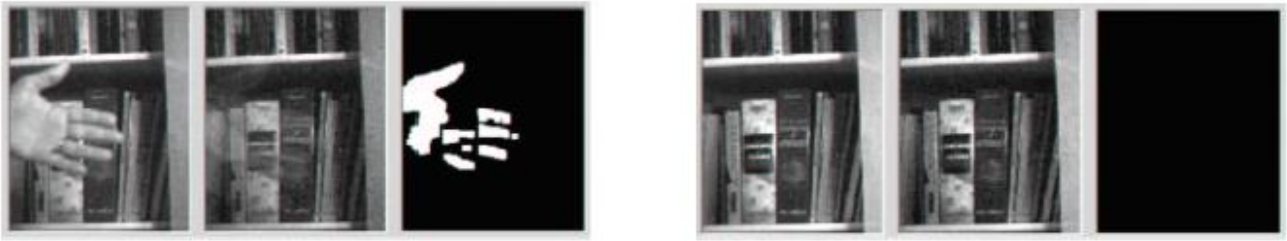


Fig. 6. The left hand side images are the input images, the middle ones are the updated background model, while the right ones are the extracted features. The algorithm cannot extract the object in those places, where the input image is in saturation.

- ✓ **Spatial-temporal foreground-background separation**: The spatial-temporal foreground-background separation algorithm is based on an image descriptor calculation. This image descriptor is a dynamic edge calculation, which means, that it extracts those locations, where the edge map is significantly different on the two images. The descriptor, which is already a binary image, contains the local features of an image. If we calculate the descriptor of the background image and the new image, the difference of the binary descriptor image will contain the moving objects. The special feature of this descriptor calculation is, that the shadows of the moving objects fools significantly less this algorithm than the first one. The algorithm is so robust, that it even provides good detection results without building a background model. In this case, one of the previous input images in the flow can be considered as a background image.

Active contour algorithm

Since they were introduced by Kass et al. [4] active contours have become a popular tool in multiple image processing tasks like segmentation, tracking and modeling.

The Vilarino's Pixel Level Snake (PLS) method [5] has been implemented. The method is based on the calculation of the internal and the external potential along the boundaries. Each pixel position then determines which way to move in each iteration step. Naturally, the pixel snake should be kept continuous in each step.

In ach iteration of the algortihm, the active contour can move one pixel. This typically indicates that after the contour is found, 2-5 iterations are enough. The execution time does not depend on the number of the contours on the image.

Multi-target tracking

The multi-target tracking algorithm itself is not dealing with images. Rather than that, its input parameters are coordinates of the candidate objects and the output parameters are the tracks. Since that, it is naturally executed on sequential processors rather than on array processors. In the Eye-RIS v1.2, we have implemented already two fast segmentation algorithms. Moreover, the morphological processing routines, and the centroid calculation is in the basic image processing library of the Eye-RIS v1.2.

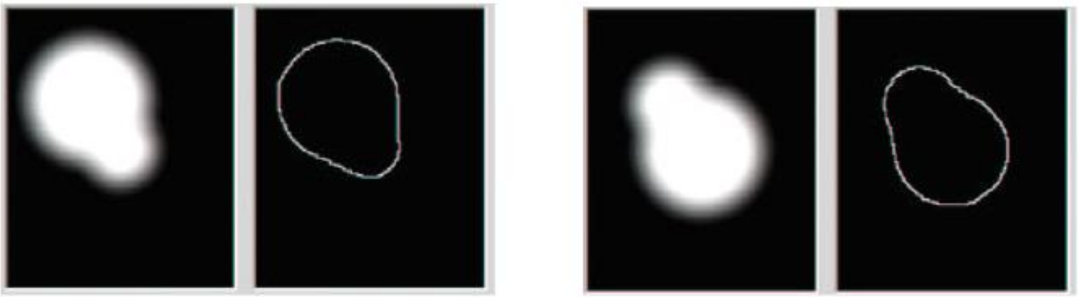


Fig. 7. Screenshot of the active contour algorithm.

In the following, different experiments implementing these visual routines derived from the computational model are presented.

4.6 Robot applications

Visual homing and hearing targeting

This demo aims at proving the reliability of the Rover II robot endowed with algorithms implementing both visual and hearing routines and related circuits.

The task to be accomplished consists in a phonotaxis behaviour performed until the battery level goes under a warning threshold. In this condition, the system triggers a basic behaviour “inherited” for survival purposes: *homing*. In the proposed experiment the robot adopts a visual homing behaviour using the visual system in a panoramic configuration. The homing mechanisms is a “life saving” behaviour that is triggered by the battery level sensor.

The robot moves in a 3x3 *m*2 arena, attracted by the sound sources that reproduce the cricket calling song. Two speakers are placed near two opposite walls, whereas a recharging station is located in a corner of the arena.

In the proposed experiments the “home” is represented by a recharging station located in a corner of the arena. At the beginning of the experiment, the robot acquires information about the home position, saving in its memory a panoramic view of the arena acquired from the home position.

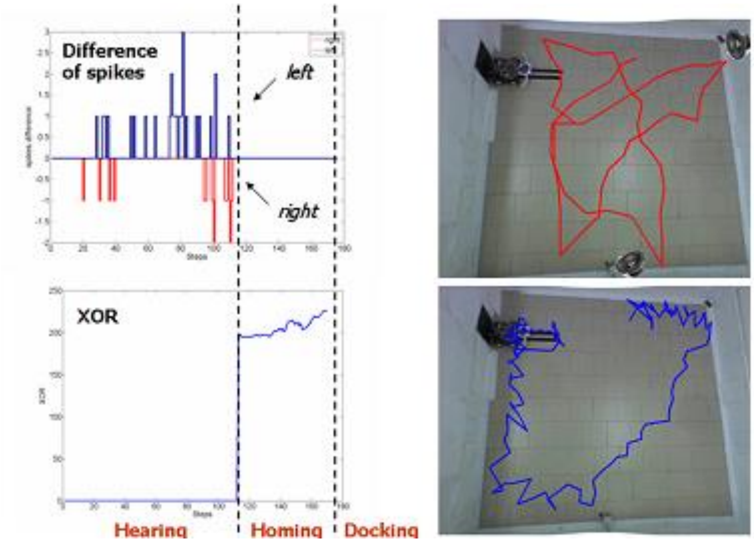


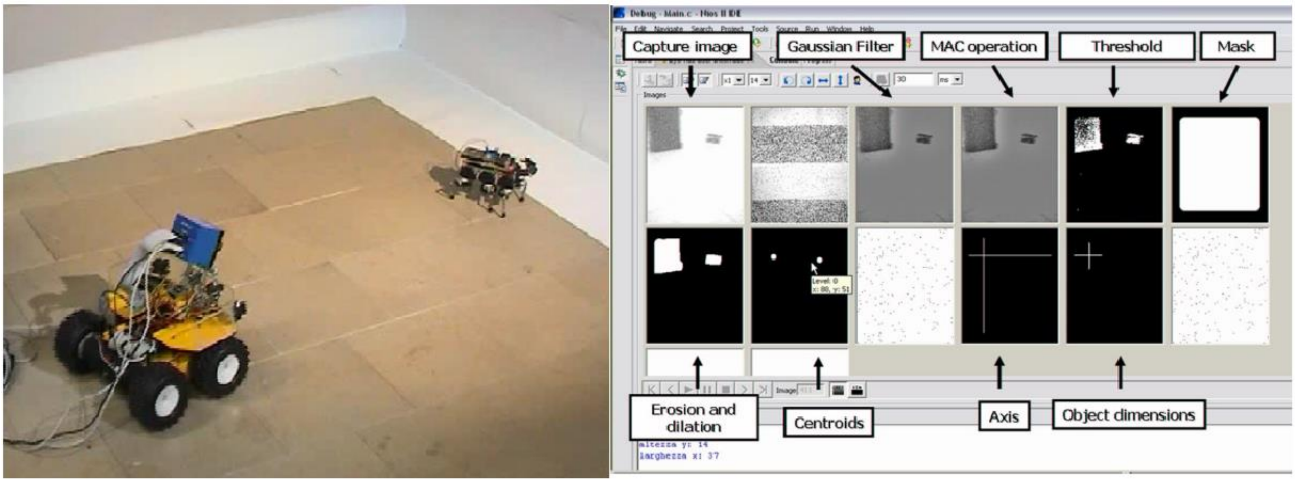
Fig. 8. Results of the hearing targeting and homing experiment.

When the homing procedure is activated, the home image is compared with the actual image. The direction to be followed is obtained using a gradient-based algorithm that is developed inside the Eye-RIS vision system device and is based on the XOR function (for details see [6]). Following the ascending direction of the XOR-based index, the robot can find the recharging station position. When the low level sensors detect black strips on the ground, the homing algorithm is stopped.

Visual perception and target following

This demo is focused on the application of visual perceptual algorithms on Rover II.

This demo emphasizes the processing capabilities of the Eye-RIS v1.2 system. The visual system, equipped on Rover II is able to process in real-time the images acquired by the robot aiming at recognizing the presence of the MiniHex robot among different other objects visible in the scene.



```
altezza y: 14
larghezza x: 37
```

Fig. 9. Image processing on Eye-RIS vision system.

The designed visual algorithm is based on a sequence of operators that deeply exploit the parallel processing capabilities of the system. In Fig. 10, the output of each step executed within the interframe rate is given. A gaussian filter together with some other operations is applied to the acquired image to increase the contrast in order to easily identify the different objects in the scene. A dynamic threshold is then used to eliminate the background from the image and a mask is applied to leave out the edge from the processing avoiding to process objects that are only partially seen in the acquired image. Templates for erosion and dilation are successively applied to filter noise and fill holes creating well defined blobs. The position of each blob is then obtained by using the centroid operator. The dimensions of the blobs can be also found tracing horizontal and vertical lines starting from the centroid, making a logical operator with the output of the erosion and dilation step, and finally counting the number of pixels. The ratio between the horizontal and vertical dimension of each object is then used as a characteristic feature to identify the MiniHex robot in the scene. The addition of further filtering functions can enhance the detection of the detection of the MiniHex structure among different kinds of objects.

Landmark navigation

The focus of this demo is to outline the robot capability to learn to discriminate between relevant and useless pictures in the arena. The relevant pictures can work as landmarks for homing purpose. Once learned the landmarks, homing takes place, even in front of partially obscured landmarks.

Rover II is equipped with the Eye-RIS v1.2, distance sensors and low level target sensors. The arena contains a target (i.e. nest) and five different possible landmarks (i.e. black picture frame with objects inside, attached to the walls).

The landmark navigation algorithm is characterized by two distinct phases.

Phase I: Landmark identification

In this phase, the roving robot randomly explores the arena filled with different types of visual cues. At each step the robot acquires information about the presence of different visual cues, provided by the Eye-RIS vision system. The robot is also able to detect the presence of the nest only within its proximity. The most reliable visual cues (three in our demo) will have, at the end of the learning phase, the highest values for their synaptic weights (through STDP learning), and will be selected as landmarks for the next phase. The arena used for the experiments is shown in Fig. 11.

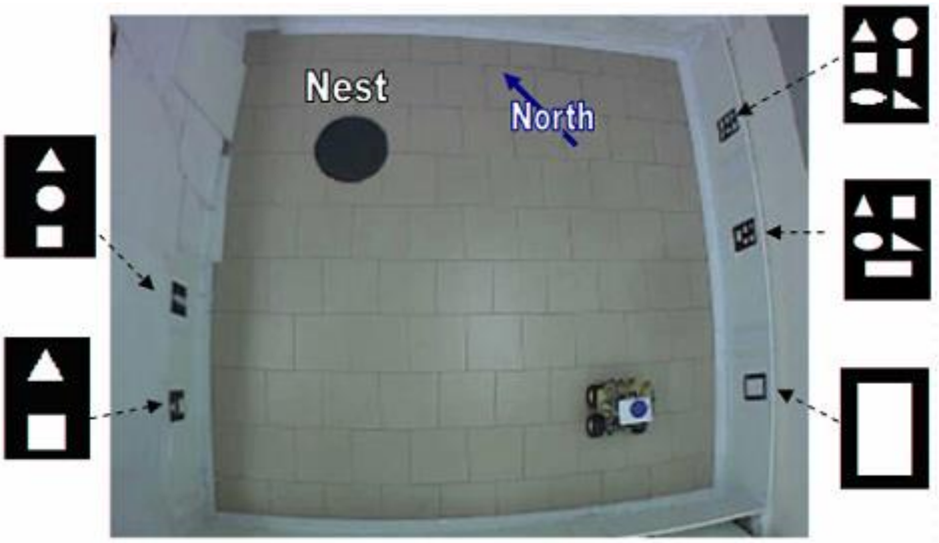


Fig. 10. Enviroment used for landmark navigation.

Phase II: Landmark navigation

In this phase the roving robot, placed on the nest position, acquires, via the Eye- RIS system and a compass sensor, information about the three most reliable landmarks, discovered in the previous phase, in order to build a map of the geometrical relationships between landmarks and nest. The error due to measurement noise (in particular on the compass sensor) is not a problem for the navigation algorithm, thanks to the filtering capabilities of the RNN structure.

After that, we let the robot forage for some steps and, once it needs to come back to the nest, it turns around looking for a landmark. The information (distance and angle) of the relative position between robot and landmark is acquired via the visual system and enters in the RNN. The RNN output is a vector which is translated in a motor command. After some iterations, the rover will reach the nest position.

5. ACKNOWLEDGEMENTS

Authors would like to thank the contribution from AnaLogic Computer Kft. regarding the development of optimized functions to be implemented within the Eye-RIS vision system.

REFERENCES

[1] J. Wessnitzer, B. Webb, Multimodal sensory integration in insects - towards insect brain control architectures, Bioinspiration and Biomimetics, Vol 1, No 3, pp 63-75, 2006.

[2] Chappell, S., Macarthur, A., Preston, D., Olmstead, D., Flint, B., Sullivan, C.: Exploiting real-time FPGA based adaptative systems technology for real-time sensor fusion in next generation automotive safety systems. In: Proceedings of Design Automation and Test in Europe (2005).

[3] www.spark.diees.unict.it and www.spark2.diees.unict.it

[4] Kass, M.,Witkin, A., Terzopoulos, D.: Snakes: Active Contours Models. International Journal on Computer Vision **1**, 321–331 (1988).

[5] Vilarino, D.L., Brea, V.M., Cabello, D., Pardo, J.M.: Discrete-Time CNN for Image Segmentation by Active Contours. Pattern Recognition Letters 19(8), 721–734 (1998).

[6] Arena, P., De Fiore, S., Fortuna, L., Nicolosi, L., Patan´e, L., Vagliasindi, G.: Visual Homing: experimental results on an autonomous robot. In: 18th European Conference on Circut Theory and Design (ECCTD 07), Seville, Spain (2007).