

# Visual Input Compensation using the Crowley-Arbib Saccade Model

Fortunato Flores Ando, Alfredo Weitzenfeld Ridel

**Abstract**—The robotic implementation of Rana computatrix uses a geometric algorithm to regain visual input that is “lost” as the result of its orienting behavior. In this paper the Crowley-Arbib saccade model is implemented in order to give this problem a solution with a biological basis. Three approaches are proposed and simulated and one is used in a small four-legged robot.

**Index Terms**—Biomimetic, robot, saccade.

## I. INTRODUCTION

SACCADES are eye movements used by primates to rapidly bring into the fovea, the most sensitive area of the retina, points of the visual scene that are relevant for the individual. Saccadic movements are used in robots with the same purpose.

In this paper we present a model of the oculomotor system involved in the generation of saccades and its implementation in a real robot used to keep track of visual input lost due to reorientation.

We used this model in experimenting with prey acquisition based behaviors requiring continuous tracking of visual input. In previous work [1] we have described frog-inspired detour behavior and its corresponding robot experimentation. While experimenting with this model, the robot kept losing track of relevant visual input that was needed to generate motor commands. This was due primarily to camera visual field limitations. To solve this problem, we used a geometric compensating algorithm. In this paper we propose a biologically inspired solution using saccades.

This paper is divided as follows. In Section II the frog’s behavior model is presented along with the robot that embodies it. Section III is an explanation of the Crowley-Arbib saccade model. In Section IV three simulation methods are

This work has been supported by CONACYT grant C03-42440 entitled “Visually-guided NeuroEthological Autonomous Robots: An Adaptive Middleware Approach to Distributed Embedded Mobile Systems” and by the “Asociación Mexicana de Cultura, A.C.”.

Fortunato Flores is research assistant at CANNES Lab of ITAM. (email: natof@prodigy.net.mx)

Alfredo Weitzenfeld is a professor and director of the Robotics Lab and CANNES Lab at ITAM. (email: alfredo@itam.mx)

described with corresponding results for saccade implementation in a virtual robot environment. In Section V robotic embodiment of the model is presented along with experimental results. Section VI presents conclusions and discusses future work.

## II. RANA COMPUTATRIX, DETOUR BEHAVIOR AND ITS ROBOTIC IMPLEMENTATION

*Rana computatrix* is a biomimetic computational system that models visually guided behavior seen in the common frog, *Rana pippiens* [2]. The model includes depth perception, object-recognition, selection and other mechanisms abstracted as schemas and implemented at a lower level as neural and non-neural substrates. Rana Computatrix models have been simulated using the Neural Simulation Language (NSL) [3].

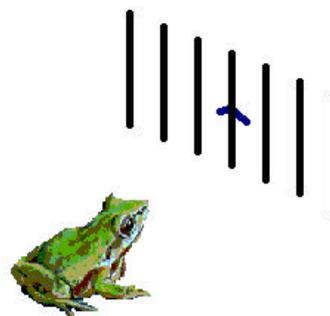


Fig. 1. Frog in front of barrier interposed on its way to the prey. The animal shows different kinds of behavior depending on how wide the barrier is and on previous frog experience.

The Rana Computatrix detour behavior model specifically deals with the behavior of the common frog when confronted to a static barrier of vertical fence posts in its way to a prey [4]. See Figure 1. In different experiments it has been seen that the real animal either shows a detour behavior, going around the barrier, or goes straight towards it, bumping several times and reorienting until it reaches the edge, where it is finally able to reach the prey. Its precise behavior depends on how wide the barrier is and on the number of times it has tried the experiment.

During simulation, visual information is processed by several schemas, creating “attractant” and “repellant” fields, i.e. excitatory or inhibitory spatial functions, that add up to create motor response. Visual input is the first step and motor action

is the last one in a series of continuous cycles that when integrated give rise to the expected frog detour behavior [4].

Rana computatrix was not only simulated in a virtual world but it was also tested in the real world by using a robot that performed the same frog experiments. Specific details can be found in [5] describing the simulation, and in [1] describing the robot implementation. Its behavior matched expected results, however the real world posed more problems than the simulation trials.

One of such problems is pictured in figure 2. When the robotic frog makes an orienting movement towards the edge of the barrier, its visual field would also change. As a result, the prey, which was previously visible, now cannot be seen and visually guided behavior cannot be calculated because there is no relevant input. This problem rises from limited visual fields in the robot's camera. During virtual simulation the size of the visual field can be adapted to avoid this problem.

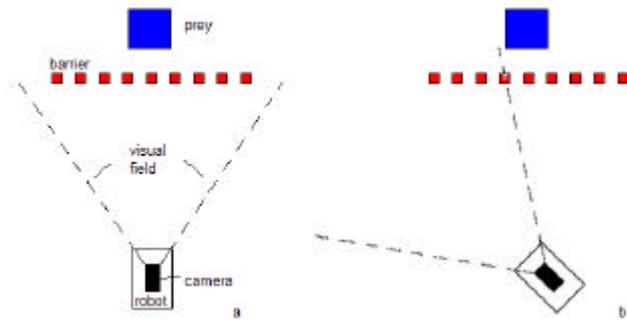


Fig. 2. Top view representation of the robot experiment and the problem that arises as it moves. a) Initial setup of the experiment, the prey and the barrier are in the visual field of the robot where new motor commands can be calculated. b) After a reorienting command, the visual field has changed and the prey cannot be seen, there is no

To bring the prey back to the frog's visual field, a motor was included in the robot enabling the camera to reorient to its original position. To control such subsystem, an algorithm, which makes geometric calculations, was designed enabling both the camera and robot to reorient back. In this way the problem was technically solved.

In this paper we investigate the use of an alternative solution, in particular, the use of saccades. This method is biologically inspired but it should be noted that we are not suggesting this mechanism actually occurs in the frog, in particular since it is based on a mammalian saccade model while toads already compensate with a larger visual field.

### III. CROWLEY-ARBIB SACCADE MODEL

The primate's retina is a highly specialized system composed of two kinds of light receptors (cones and rods) and 4 kinds of neurons (amacrine cells, ganglion cells, horizontal cells and bipolar cells). What could be considered "high resolution" vision is achieved thanks to a small region of the retina called *fovea*, having a high density of cone receptors. However the portion of the visual field that this region subtends is very small and the eye has to be constantly moved around to

"scan" outer world images in search of relevant visual features. These "scans" or saccades are very fast eye movements in response to dynamic stimuli outside the fovea. The Crowley-Arbib [6][7] saccade model, an extension of the Dominey-Arbib [8] model, tries to explain the generation of saccadic eye movements with a system that integrates the workings of cortical and subcortical structures in primates. It is successful in explaining the gap saccade and double saccade paradigms taking into account experimental recordings in live primates.

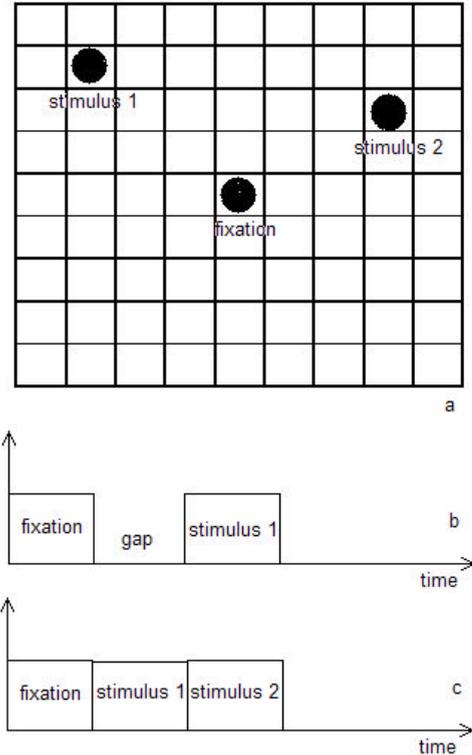


Fig. 3. a) Experimental setup for saccade paradigms. The visual field is divided in an array of elements that may be active or inactive, as seen in Figure 3a. In the gap saccade task, a fixation stimulus to the fovea is presented for a brief time. A short interval after it is turned off, another location in the visual field is activated. The eye must saccade to this second point, i.e. it must center the fovea to such location. See figure 3b. In the double saccade task, after a fixation point is presented, two targets appear in sequence. The eye must saccade to each fixation point in the same order as they appeared. The total duration of both stimuli is smaller than the time it takes to start the first saccade. See figure 3c. In this paradigm several mechanisms are involved such as memory and remapping.

Figure 3 shows the basic setup for such experiments. The visual field is divided in a 2-dimensional array of locations that are either active or inactive, i.e. lighted or obscured, as seen in Figure 3a. In the gap saccade task, a fixation stimulus to the fovea is presented for a brief time. A short interval after it is turned off, another location in the visual field is activated. The eye must saccade to this second point, i.e. it must center the fovea to such location. See figure 3b.

In the double saccade task, after a fixation point is presented, two targets appear in sequence. The eye must saccade to each fixation point in the same order as they appeared. The total duration of both stimuli is smaller than the time it takes to start the first saccade. See figure 3c. In this paradigm several mechanisms are involved such as memory and remapping.

Since the stimuli are not available during or after any of the saccades, there must be an internal spatial representation of their locations in the neural layers of the oculomotor system. When the first saccade is done the neural image representing the second location requires a “remapping” to compensate for the transformation of coordinates that happens when the eyes move. For a detailed explanation of the model refer to [6] and [7].

#### IV. ALTERNATIVES FOR SIMULATED VISUAL INPUT COMPENSATION

Saccades have been implemented in several robotic systems. In [9] saccades are used in an autonomous robot to redirect vision to potentially interesting targets. In [10] and [11] saccades are used along with other strategies in order to track selected moving targets. Here the saccadic system is proposed as a way of recovering visual information that has been lost due to the movement of the robot. While robots [9] [10][11] use saccades with non-neural, although biologically inspired algorithms, in our paper the saccadic system is a neural model where different neural layers are simulated as dynamical systems.

Three approaches to embody the Crowley-Arbib saccade model have been devised [12]. They have been tested along the Rana Computatrix model in the same virtual world used in the original experimentation. The more “suitable” of the three approaches was selected and implemented using an Aibo robot [13]. The three approaches are now explained.

The first alternative was to divide each video frame sent by the camera into a 9x9 array so that the central element could be considered as the fovea. In this computational retina no difference in resolution among fovea and periphery was considered, but the spatial location of the former was used to determine whether the prey was in the center of the visual field or not. This approach takes advantage of the higher velocity of saccades compared with detour behavior computation in order to keep the “prey” centered and focused all the time. See Figure 4. For this to work, calculation and realization of saccades have to be done in parallel to detour behavior computation so that discrete steps commanded by the motor schemas of Rana computatrix are continuously compensated.

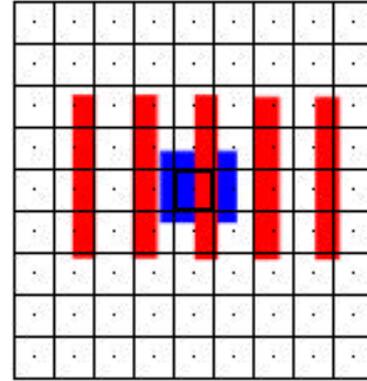


Fig. 4. The visual field is divided in a 9x9 array, with the fovea represented by the central element. Whenever the prey (dark square in background) is not foveated the system commands the camera to move in order to recenter it.

Discrimination amongst prey and barrier is done using colors, where blue represents the prey and red the barrier. The image frames that come from the camera are filtered to detect such features.

After many simulations this alternative was discarded because it imposed problems that could not be efficiently solved. Computational cost was too high because video processing was too demanding. Detour motor commands had to be divided into smaller subcommands in a way such that after each subcommand the prey would still be visible, yet not centered. The saccade system was continuously run in order to compensate for this.

The second alternative was to consider the camera as one element of the retina and use several image frames, each one taken from a different angle, to create the whole image of the retina. See figure 5. This approach requires that before each saccade, the system needs to make a “scan” of the visual field by rotating the camera and integrating the resulting information in order to create a greater visual input to the saccade model.

Nine different positions of the camera were used for each simulation. Each image, coming from a different angle, was filtered in search for a prey stimulus and was used as visual input to specific retina locations, as seen in figure 5.

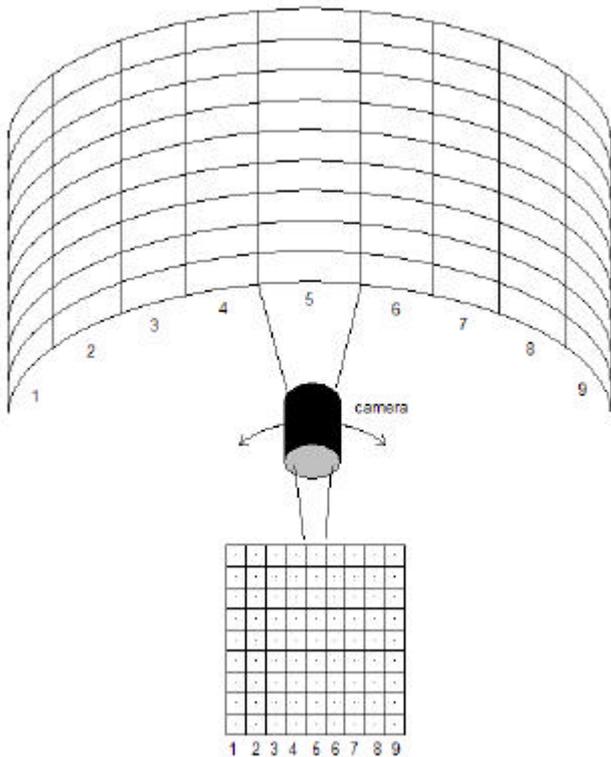


Fig. 5. The camera rotates to get 9 different images. Each image is then used to generate input to 9 locations of the retina. The resulting 9x9 locations make up the visual input for the saccade model

This approach was also discarded after many simulations because the continuous scanning of the visual field was considered artificial, inefficient and prone to errors. The amount of processing needed to create the visual input was prohibitive and to keep the robot's camera moving in all directions would just lead to a quick degradation of the system.

The third approach considered the camera as the fovea and the rest of the retina to be coarsely and virtually calculated by means of a feedback signal coming from the motor schemas of Rana computatrix. See figure 6. In this way when the prey disappears from the actual camera image, the "virtual" periphery of the retina would still carry information on where the prey is, or is supposed to be, and a saccade could be made in order to bring it again into the visual field of the camera. The state of visual input out of the foveal region is calculated using a signal that codes the direction (right or left) of the rotation generated by detour behavior. Saccade locations are computed from the computed retinal image.

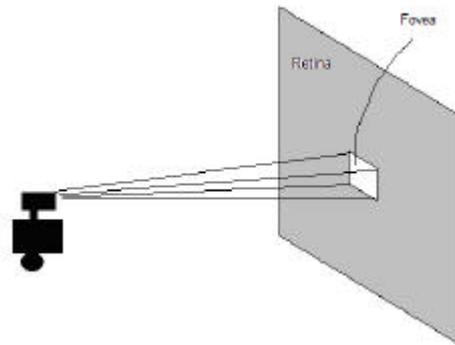


Fig. 6. What the camera sees is considered the fovea, the rest of the retina is calculated according to feedback coming from motor centers.

For this method to work the prey has to be in the visual field at the beginning of the experiment, which is also a restriction imposed by Rana computatrix. In this way by "remembering" the past location and knowing where the frog rotated to, the "virtual" retina can be built. The "virtual" visual input is created in a very rough way. If the frog rotates left, the rightmost location of the visual input is set to some arbitrary value. In this way a saccade would be made to that location. If the rotation is to the right then the leftmost location is set and a saccade is elicited in that direction.

The activation of the fovea is determined by the presence of a prey stimulus in the visual field of the camera so the image is filtered in search of such relevant information. If such a stimulus exists then the fovea is activated to an arbitrary level. No additional image processing is needed since everything else in the retina is calculated according to feedback signals.

Figure 7 shows the simulation at different times. Figure 7a shows the initial setup. The virtual robot is able to see the prey. Figure 7b and 7c show what the frog sees after 25 degrees rotation to the right. This rotation simulated a detour behavior model command. The prey is not visible any more but the "virtual" visual field shows activity in the leftmost center element, as can be seen in figures 7d through 7f. This visual input will elicit a saccade to that location. At the end of the saccade the prey is again in the visual field so rotation has been compensated, as shown in Figures 7g through 7i.

The third approach was considered the most feasible of the three because it required considerably less video processing and hence was faster.

## V. ROBOTIC EMBODIMENT

For the reasons stated above, the last approach was chosen for implementation in the real robot. We used Sony's Aibo as the robotic platform for the experiments since it is very stable and relatively easy to use. The Aibo robot we used is the ERS-210A having 20 degrees of freedom: 3 for the head, 3 for each leg, 1 for each ear, 1 for the mouth and 2 for the tail. It has a camera with a field of view of 57.6° degrees wide and 47.8° high with a resolution of 88x72 pixels.

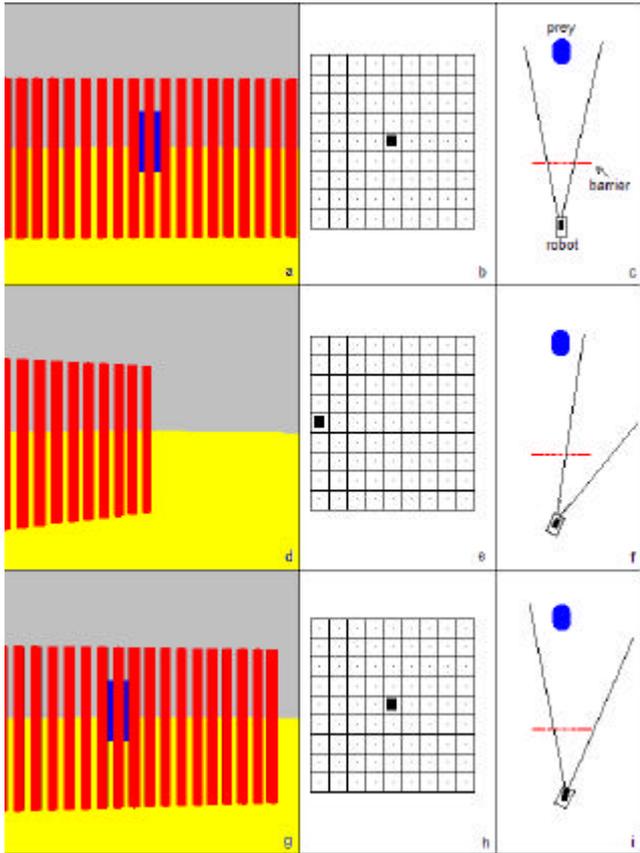


Fig. 7. a) Initial setup. The robot sees the barrier and the prey. b) The fovea is active since the prey is seen. c) Simplified top view showing the spatial accommodation of each component. d) Image seen by the frog after the rotation commanded by Rana computatrix has been done. The prey can no longer be seen. e) As a result, the fovea is inactive. Visual input now carries relevant information on the location of the prey in the visual scene. f) Top view showing rotation of the robot. g) The saccade has ended, the prey is again in the visual field. H) The fovea is active. i) Top view showing the reorientation of the camera

With so many degrees of freedom it could be thought that the electromechanical control problems are the most demanding for the user, however we used Tekkotsu [15], which is a software framework for the development of new Aibo robot applications that takes care of low level control routines, letting the modeler focus on the behavioral system. Within this framework there is a monitoring utility called *TekkotsuMon* that lets the user access the state of the robot and control it in real time. The fact that the Crowley-Arbib model is implemented in the Java language version of NSL and that *TekkotsuMon* has a Java graphical user interface (GUI) lets us construct an interface between the two applications with relative simplicity. In this way image frames sent by the robot were processed and introduced to the model and eye motor commands were sent back to the robot by the saccadic system through the GUI.

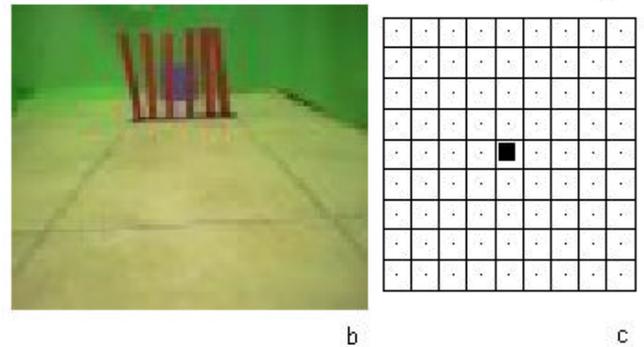
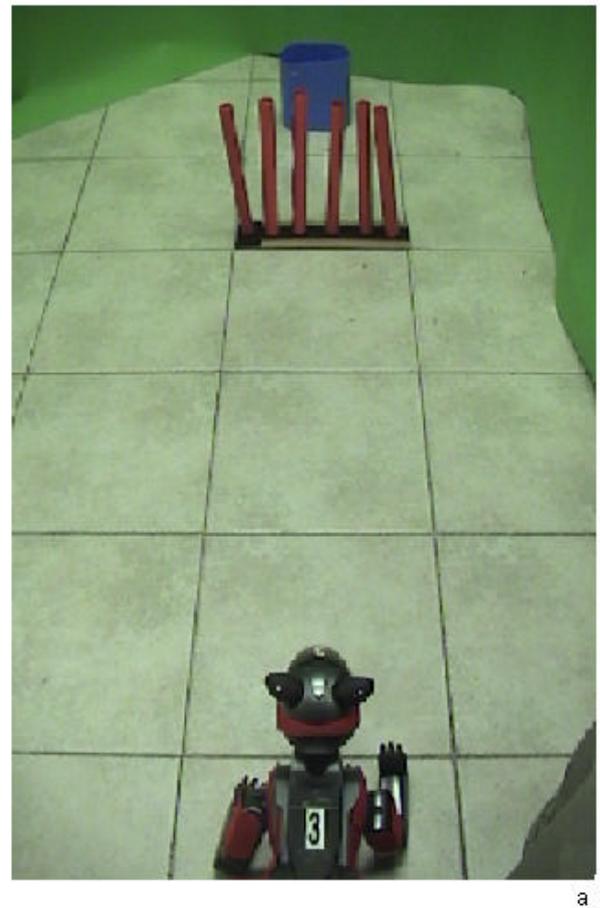


Fig. 8. a) Initial setup. The robot is in front and about a meter away from barrier. b) Robot visual field. It is able to recognize both prey and barrier. c) The fovea is active since the visual field recognizes a prey object

Video frames sent to the saccade system are 88x72 RGB images that are processed in the visual input layer of the model in order to detect a prey. Prey and barrier are identified by color. Blue objects are considered to be preys while red objects are considered to be obstacles. We use a very simple color detection algorithm based on differences of RGB values and empirical thresholds. The color detection algorithm is relatively sensitive to noise so the robot's environment is highly structured in order to minimize conflicts. Figure 8 shows the basic experimental setup. Images are filtered in search for prey objects where a mean "activation" value is obtained by:

$$act = \frac{\sum_{i=0}^{87} \sum_{j=0}^{71} filter(i, j)}{88 \times 72} \quad (1)$$

where  $filter(i, j)$  assigns the pixel in location  $(i, j)$  a value of “1” if it is considered blue or “0” for something else. If such value is greater than an empirically specified threshold then the fovea is set active to an arbitrary constant value. The virtual periphery of the fovea is refreshed depending on the orientation movements of the robot.

In a first attempt the coarse coding scheme proposed in the third simulated alternative was used, i.e. visual input is computed just left or right of the fovea as explained in the preceding section. To overcome this coarseness we take advantage of the nine possible horizontal positions for visual input. Depending on detour movement amplitudes the horizontal position of the visual input activation is calculated according to:

$$horPos(k) = \alpha \Delta t - horPos(k - 1) \quad (2)$$

where  $horPos(k) \in [-1, 1]$  is the horizontal position of visual input of the  $k$ th motor command corresponding to detour behavior;  $\alpha$  is an empirically derived constant and  $\Delta t$  is the required time to execute the motor command.

The  $horPos$  value is then mapped to integer elements in the interval  $[0, 8]$  to make it correspond to the index of the visual input array. Figures 9 and 10 show two instants of the experiment done using this scheme. The initial setup is that of Fig. 8 where it can be seen that the fovea is active, corresponding to the prey seen inside the visual field. Several orienting movements are commanded in sequence, after each one the saccadic model works in order to compensate for this and the head is moved according to the model’s behavior. After each saccade the head goes back to its original position and the saccade model is reset. The fixation mechanism of the saccade model prevents any movements when the prey is visible.

Figure 9a shows the robot just after an orientation movement has been done to the left. In figure 9b the visual input to the saccade model can be seen, the fovea is not active but an element to the right is, hence a saccade is done in that direction as can be seen in figure 9c. In Figure 10a, the visual input is seen at a later instant. As shown in Fig. 10b, an orientation movement has just finished. The fovea is active, meaning the prey is visible so no saccade is elicited. Figure 11 illustrates the path followed by the robot when simulating Rana computatrix behavior. The video showing the whole experiment of these figures can be found in [16].

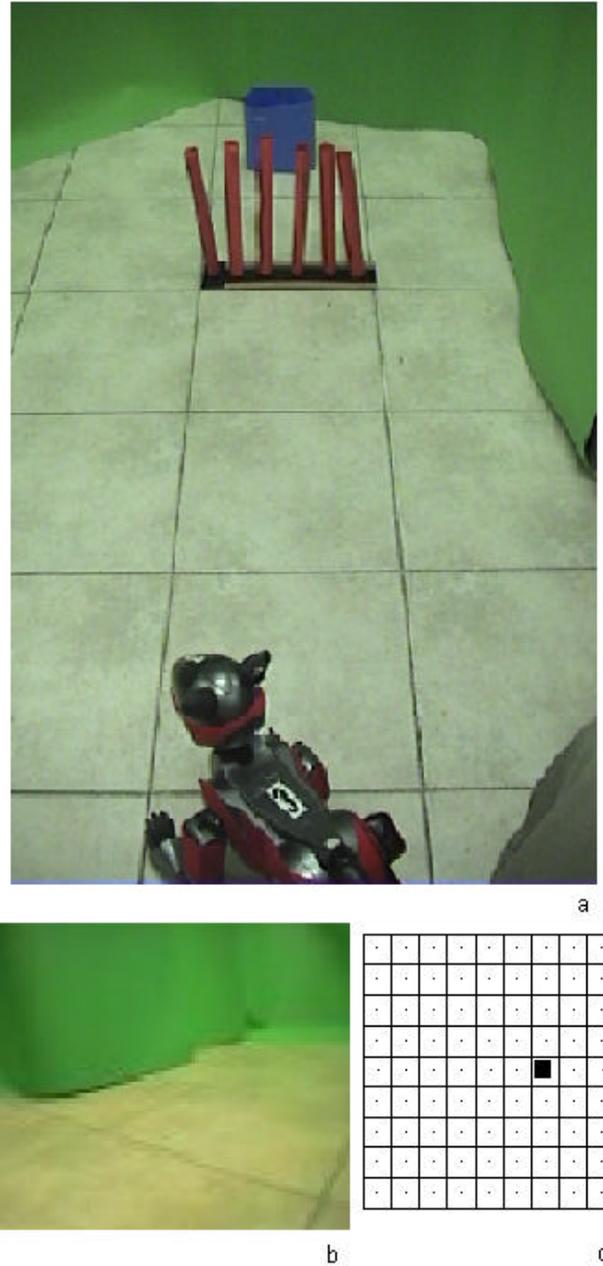


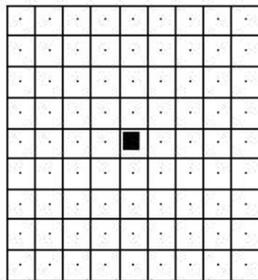
Fig. 9. a) The robot has done a detour rotation movement. b) The prey cannot be seen. c) This reflects on an inactive fovea with prey activation in some other region of the virtual visual field.



a



b



c

Fig. 10. a) The robot has followed a path corresponding to Rana computatrix behavior. A motor command has just been finished. b) In this case the prey can be seen. c) As a result, the fovea is active and no saccade will be started.

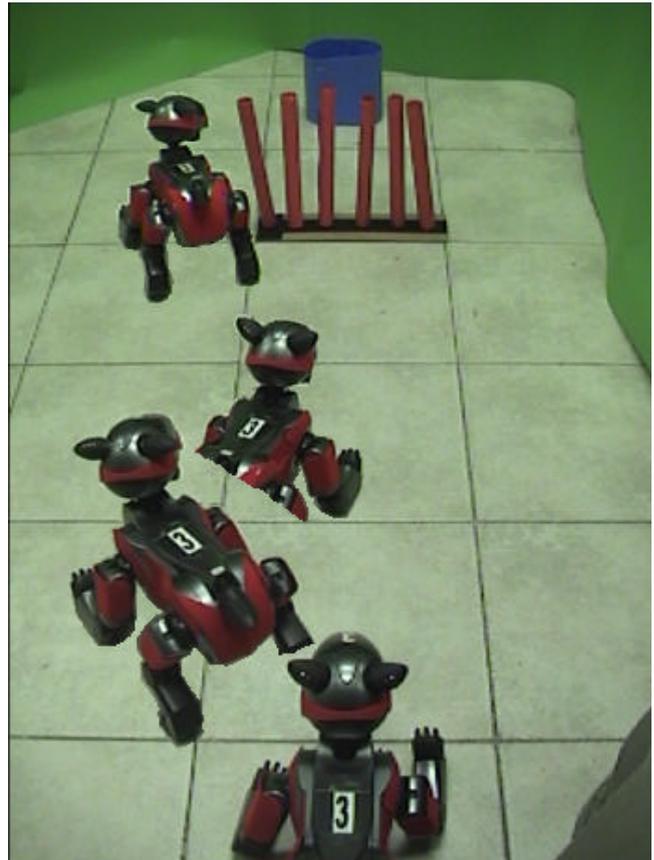


Fig. 11. The path followed by the robot in simulating detour behavior.

## VI. CONCLUSIONS

Three schemes were proposed in using the Crowley-Arbib Saccade model as a visual input error-correction mechanism in the robotic implementation of the visually guided detour behavior of Rana computatrix.

Simulations were done in order to qualitatively test the performance of each correction proposal. The first alternative worked well but it had two drawbacks. The first is related to expensive video processing required to generate full visual input. The second and most important is that it does not completely solve the problem because it performs saccades when the prey is not centered but is still inside the visual field. It was proposed that by exploiting the relatively higher processing speed of the oculomotor system, in contrast to Rana computatrix, this method could be used. However it may be difficult to coordinate both systems to achieve smooth processing.

The second alternative proved to work well but it requires more video processing than the other methods. The constant movement of the camera may be something difficult to coordinate and too artificial to work.

In the third proposal the camera is seen as the fovea and the rest of the retinal image is virtually generated with a simple algorithm. The simulation results showed that the “virtual”

visual input that was calculated had some error. It estimated the direction correctly but the amplitude incorrectly, i.e. the prey was closer to the fovea than it was actually computed. However this scheme was considered more efficient and was taken as a basis to implement the saccade model in an Aibo robot.

The setting in which the robot was tested was constrained in order to minimize noise in the visual input path. The “prey” reconnaissance algorithm implemented in the robot is fairly simple and based in color recognition. It is the same used in Rana computatrix. For applications in other robots and depending on the task a much more complex algorithm would have to be construed. The saccade target in this case was solely determined by the existence of a prey. Saccade movement amplitudes were added in the real robot model. In the future, mechanisms like the remapping circuit of the saccade model should be investigated in order to achieve this same task by letting the robot learn how to correctly compensate for its own movements. An effort to find such a subsystem would be interesting from the neural modeling point of view because it could lead to biological plausible results and engineering novel ways. It could also be useful in the development of new models regarding the orienting behavior shown by real frogs, in models trying to explain remapping of saccadic targets when not just the eyes but the whole body is moving or in explaining how such remappings are done when visual information is missing or is not the only one that has to be taken account of.

The results of the simulations and the robotic implementation show that the use of the saccadic model may be useful.

The embodiment of the model lets us detect oversimplifications and unreal working conditions that we had overseen in the simulations.

It should be noted that in the experiments we concentrated on the saccade model and have not integrated the detour behavior model. The orientation commands in the presented experiments were not issued by the detour model but were created in order to simulate Rana computatrix behavior. We are already working in achieving a full integration. Along with the difficulties of adapting a saccade model that was created to explain very specific experimental results obtained through the course of very specific experimental protocols, new challenges arise, as it is necessary to integrate it with Rana Computatrix, not just glue it together as this would have no biological relevance [1][17]. The effort of integrating both models may prove to be the most interesting and fructiferous task because the models are based in very different neural systems. These differences may suggest what and how strategies to solve specific problems evolved as the neural systems developed higher capacities.

#### REFERENCES

- [1] A. Weitzenfeld, “Embedded Mobile Systems: From Brain Theory To Neural-based Robots”, Proceedings of 11<sup>th</sup> Mediterranean Conference on Control and Automation, Rhodes, Greece, July, 2003.
- [2] M. Arbib, *The metaphorical brain 2*. New York: John Wiley and Sons, 1989.
- [3] A. Weitzenfeld, M. Arbib and A. Alexander, *The Neural Simulation Language: A System for Brain Modeling*, MIT Press, 2002.
- [4] Corbacho, F., and Arbib M. Learning to Detour, *Adaptive Behavior*, Volume 3, Number 4, pp 419-468, 1995.
- [5] F.J. Corbacho and A. Weitzenfeld, “Learning to Detour”, in *The Neural Simulation Language: A System for Brain Modeling*, Aut. A. Weitzenfeld, M. Arbib and A. Alexander, MIT Press, 2002, pp. 319–342.
- [6] M. G. Crowley, *Modelling Saccadic Motor Control: Normal function, Sensory Remapping and Basal Ganglia Dysfunction*. Ph.D. dissertation, Fac. Grad. School, University of Southern California, California, 1997.
- [7] M. Crowley, E. Oztop and S. Marmol, “Crowley-Arbib Saccade model”, in *The Neural Simulation Language: A System for Brain Modeling*, Aut. A. Weitzenfeld, M. Arbib and A. Alexander, MIT Press, 2002, pp. 283-304.
- [8] P.F. Dominey and M.Arbib, “A cortico-subcortical model for the generation of spatially accurate sequential saccades”, *Cerebral Cortex* vol. 2, pp 153-175, 1992.
- [9] M.R. Blackburn and H.G.Nguyen, “Autonomous Visual control of a Mobile Robot”, Presented at the 1994 Arpa image understanding workshop, 1994, Monterey, CA.
- [10] Ch. Balkenius and L. Kopp, “Visual Tracking and Target Selection for Mobile Robots” in *Proceedings of EUROBOT '96*.
- [11] H. Kim, B. Lau and J. Triesch, “Adaptive Object Tracking with an Antropomorphic Robot Head” in *Proceedings of SAB'04*, CA, USA, July 2004.
- [12] F. Flores and A. Weitzenfeld, “Alternatives for the Implementation of the Crowley-Arbib Saccade Model in a Robotic System”, Proc. of 1<sup>st</sup> Latin American Robotics Symposium LARS 2004, IEEE-RAS, Mexico City, October 28-29, 2004.
- [13] Aibo robot web site, <http://www.sony.net/Products/aibo/>
- [14] D.L. Sparks, “The Brainstem Control of Saccadic Eye Movements”, *Nature*, vol. 3, pp. 952-964, 2002.
- [15] Tekkotsu web site, <http://tekkotsu.no-ip.org/>
- [16] Saccade video, <ftp://ftp.itam.mx/pub/alfredo/videos/saccadeRobot.avi>
- [17] B. Webb, “Can Robots make good models of biological behaviour”, *Behavioral and Brain Sciences* 24(6), 2001.