Câmbio: A Realistic Simulated Robot for Vision Algorithms

CLAUS C. ARANHA[1], SCHUBERT R. CARVALHO[1], LUIZ M. G. GONÇALVES[2]

[1]Universidade Estadual de Campinas
Caixa Postal 6176, 13083-970, Campinas, SP, Brasil
{claus.aranha,schubert.carvalho}@ic.unicamp.br
[2]Universidade Federal do Rio Grande do Norte
DCA-Centro de Tecnologia-UFRN, 59.072-970 - Natal, RN, Brasil
lmarcos@dca.ufrn.br

Abstract: We introduce concepts and algorithms for control of visual motor commands and realistic simulation of basic abilities as visual servoing and perception for robotics vision simulation. We use the tools to build part of a humanoid (arms and head) robot, affectively named "Câmbio". We describe Câmbio's design, providing an overview on the most used feature extraction techniques for perception, discussing implementation issues. We show the usefulness of a simulated platform as an inexpensive alternative for testing and developing computer vision algorithms in real-time robotics applications.

1 INTRODUCTION

In this work, we present a realistic, simulated device for computer vision, robotics, and humanoid research named *Câmbio*. We provide a dynamic simulation of a stereo head and its image processing capability and of two integrated arms for object manipulation. We provide some tools for feature abstraction to be used as the basis for high level processes like, for instance, visual attention: use of Gaussian derivative operators for feature extraction, use of Multi Resolution Images (MR) for data reduction, use of Stereo Disparity and Motion features, and use of statistical moments. The main goal is to allow researchers in Computer and Robotics Vision to perform experiments at reduced costs, running different tests with different robotic setups, using different testing algorithms, and in different situations putting the robot to its limits without the fear of (or even intending) breaking it up. By using efficient and realistic simulation tools, we can work on several aspects of cognition, attention, sensorimotor development, computer vision, and other research fields, without the immediate need of a real robot. There are some associated costs not only related to running the world, but to the task of defining rules for such world, and to simulating it as well as the real world the robot will face later. We propose, however, that we could reach a reasonable amount of similarity to the "real world" if we put the simulated domain within certain bounds, and slowly stretch these bounds to upper limits.

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

2 Related Work

Several robot simulators can be found currently [1]. Mainly there are very simple simulators of high-level robot capabilities as robot mission and path planning, or low-level skills as robot motion and motors simulation. In general, they lack low-level sensing simulation, which is directly provided in form of geometrical primitives as depth or a map model of the environment. The 3D simulator model proposed in this work, Câmbio, was carried out from the needs of developmental improvements to be done in another simulated robot environment, "Roger-The-Crab". Roger is a bidimensional-world simulator imitating a crab which was originally implemented in 1991 to be used as a tool for high-level vision research. Roger has been used to develop tools and algorithms for attention control [2], pattern categorization [3], and new models for computer animation [4]. Some of these works were the test-bed for real-time algorithms running in a stereo head [3]. The main improvements we introduce are the addition of the third dimension to the world model and to have a more realistic perception module. When dealing with perception, some main issues are what kind of features should be extracted from the environment and how to obtain them, how to define a feature, how to process acquired images, how much time to

look for a region in order to allow the system to get the desired features, and how to do all of this efficiently. Some works on visual perception [5, 8, 9] use stationary images as the target of their visual algorithms, not considering temporal aspects like motion, stereo vision, and behaviour. A good technique inspired by biological evidences is proposed in [7] using operators based on derivatives of the gaussian distribution for the extraction of image features. In [6], they try to build a "fovea" region in which to put the zones of interest. They use multiple scales of a same image, with the intent of centering the region of interest in the scale with higher resolution. The work was developed in an online robotic platform requiring a dedicated image processing device. In our work, besides in simulation, we provide a model with a more accurate representation of the "real world" than the above ones for systems that will later be implemented in online robots.

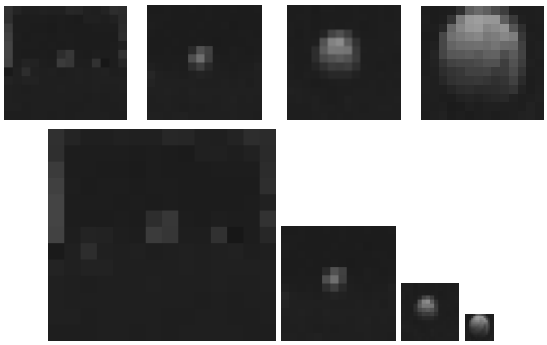

Figure 1: Top: Multi-resolution image of a sphere, $16 \times 16 = 256$ pixels each. Botton: Images re-scaled to represent the region that they cover on the original image.

3 Theoretical Background

In practice, to extract features from an acquired image it is generally necessary to perform operations as "convolution" (filtering). This yields a very bad efficiency, specially when required to apply several operators in a sequence, in a given image. The use of multi-resolution (often used multi-scale, pyramidal, retinal or foveal) images consists of using several smaller images downsampled from the same original image in different resolution levels. The pyramidal approach is still expensive as it uses the whole image in its highest resolution level. In the current work, we use a retinal approach carrying out an image representation somewhat similar to the one present in the human eye. The world is not sensed in a homogeneous manner. There is an image level containing the central region of the visual field in full resolution for detection of details (the fovea). Other image levels contain larger areas, even the whole visual field, in decreasing resolution. In practice all image levels have the same size of the fovea image, consequently with smaller resolution. To transpose this idea to a computer, we apply a number of "averaging" filters with different kernel sizes through differently sized portions of the original image. We use 4 images of the same size with different resolutions as seen in top of Figure 1. The original represented portions that they really cover on the original image can be seen in botton of Figure 1, where the MR images are re-scaled. Now we can apply other desired operators for feature extraction at a very reduced cost.

Currently, we provide up to the (second order) Laplacian of Gaussian operator for feature extraction. Equation 1 represents the basic formula for the used gaussian distribution. Equation 2 represents the gaussian convolution function and the derivative kernels used are given by Equations 3, 4, and 5, in two orthogonal directions each. One can note we modified the kernels of the derivatives, without loss of generality. We yet provide stereo disparity features, that is, perceived difference between the projections of a world point in two images generated by a pair of cameras set apart a certain distance. A simple correlation approach is used over the above second order Gaussian derivative, considering estimations given by previous levels. Motion features are also computed from two consecutive MR image frames.

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{1}$$

$$G_{d=x,y}^{(k=0,1,2)} = g_d^{(k)} * I_t \tag{2}$$

$$\left.\begin{array}{l} g_x^{(0)}(x,y) = \lambda e^{ax^2} \\ g_y^{(0)}(x,y) = \lambda e^{ay^2} \end{array}\right\} \tag{3}$$

$$\left.\begin{array}{l} g_x^{(1)}(x,y) = 2a\lambda e^{a(x^2+y^2)}x \\ g_y^{(1)}(x,y) = 2a\lambda e^{a(x^2+y^2)}y \end{array}\right\} \tag{4}$$

$$\left.\begin{array}{l} g_x^{(2)}(x,y) = 2a\lambda e^{a(x^2+y^2)}(2ax^2+1) \\ g_y^{(2)}(x,y) = 2a\lambda e^{a(x^2+y^2)}(2ay^2+1) \end{array}\right\} \tag{5}$$

$\forall (x,y) \in ([-s,+s],[-s,+s])$; where $a = \frac{-1}{2\sigma^2}$, $\lambda = \frac{1}{\sigma\sqrt{2\pi}}$, $\sigma = 1.7$, and $s = 3$.

4 Implementation

Câmbio's architecture has four modules (Figure 2): World Model Module (WMM), Sensory Servo Module (SSM), Decision Making Module (DMM) and Control Module (CM). The world is modeled in a simplified fashion, objects are stored as a set of their parameters including position and orientation in the world. A special kind of world objects are Câmbios parts themselves. The WMM alter the objects parameters according to the Control Module's requests. Another role of the WMM is to produce sensory information as requested by the SSM. The WMM uses the eye's

position and orientation to build visual sensory information (image). Actually, this is done by generating an OpenGL scene of the world and using built-in libraries to render the image. Arm arm sensory data is simulated by running a first round of a "bubble" algorithm to find out which objects are candidate for "touching", and then geometrically testing each of these candidates against the end effector.

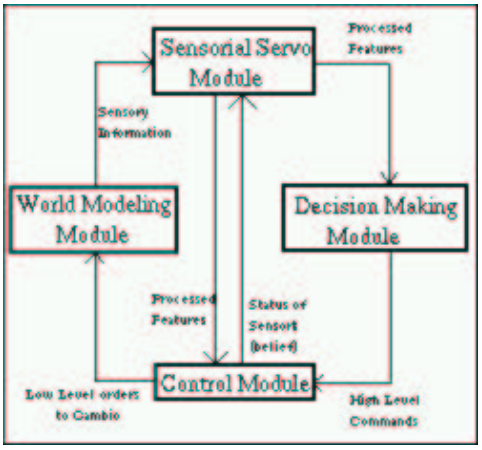

Figure 2: System modules.

The SSM must process information sent by the WMM, according to requests of the DMM. The SSM is currently an information server which processes images and arm data, building feature maps as needed for the high level processes. The arm sensory data is, currently, directly forwarded to the DMM without any processing. For the eyes, the SSM first generates an MR image as seen above. Then, it applies any required convolution operators over this MR producing the final 4 (16×16) images for each desired feature, in each eye. Actually, we provide the partial derivatives of the Gaussian Operator, with σ values of 1.4, 2.7 and 4. The use of the Gaussian function with other parameters, or even completely different convolution masks is possible with very little code changing. Also, it is offered the disparity and motion operators.

The DMM is the most "user managed" module. It applies some sort of user defined algorithm in the information provided by the SSM, which results in actions to be forwarded to the Control Module. The DMM may contain subsystems for learning, attention, and pattern categorization based on the services provided by the other modules. Currently, we have implemented the upper level controller for the vergence algorithm, and the standard-input controls for moving each part of the robot and requesting information from the SSM. Other high-level tasks can be part of any high-level, coming algorithms.

The role of the CM is to receive high level actions from the DMM, and to turn them into lower level control directives for the robot actuators. The CM stores the same low level information about the "robot objects" as the world does, and alters them according to its actions. This redundancy is to provide a certain degree of error to Câmbio. We currently added controled Gaussian errors in the motion controllers. So, we approximate our device to a real platform, and, differently of the real platform, we have ways to check where exactly a given resource is by using the information redundancy.

5 Experiments and demonstrations

In order to test Câmbio's architecture, we have implemented a high-level Vergence Controller. The idea is to keep both eyes "verged" in a place in the world while the robot searches for and/or moves to a new focus of interest, as in [2]. A very basic algorithm for changing attention moves the "dominant eye", given by some attentional process, so as the vergence point gets as close as possible to the center of its image. Concurrently, the other ("non-dominant") eye's position is adjusted in a loop which tries to minimize the disparity in the space around the vergence point. If any of the eyes motion requires a horizontal movement greater than 30 degrees, the neck has to move. While doing all of that, the system also takes care for the eye axis not to open more than parallel to each other.

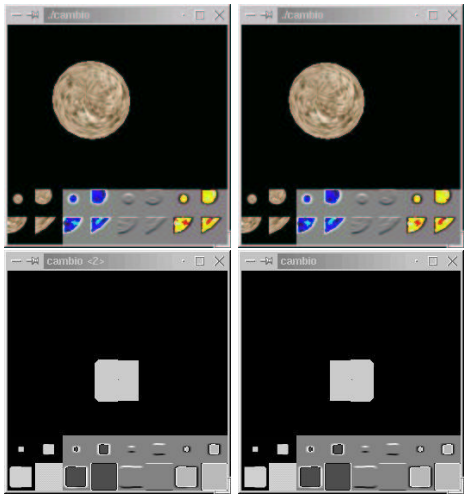

Figure 3: Letting Câmbio's eye verge on 3D objects.

In order to test disparity working in the control of vergence, we put by hand one of Câmbio's eyes still in an object point. The other eye axis is initially parallel to the first. Then we let the simulator to start its operating loop. As a result, the other eye axis goes close to the first eye one, by minimizing disparity for the eye centers, as seen in Figure 3. One can note stereoscopy by verging the eyes, one in each picture, on the textured sphere in the top pair. The same experiment with a cube (a not textured object) is seen in the bottom of

Figure 3. In this case, the algorithm could verge close to the object, but it did not perform that well due to the lack of texture.

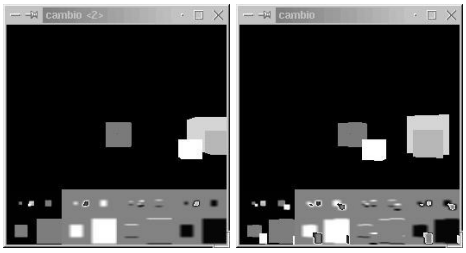

Figure 4: Câmbio gets back and verge again.

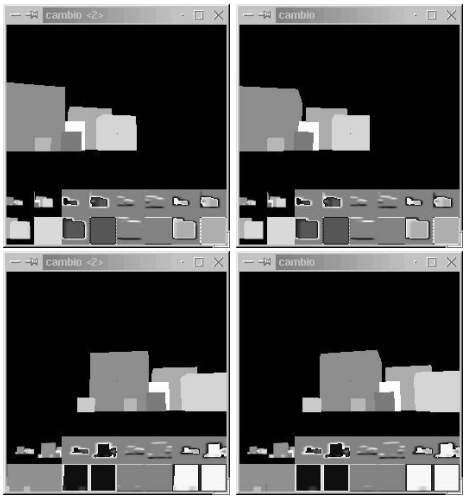

Figure 5: Câmbio executing a saccadic movement.

In another experiment, other objects enter the visual field, while keeping the same goal point in space for the right eye (the cube). It keeps verged in the same object as seen in Figure 4. The correlation approach and the sparse distribution of the objects in the environment makes the other objects interfere a little in the vergence. Besides, one can also note depth here by verging the eyes.

In other tests, with Câmbio's eyes verged in some object, we send send eye motion commands to its control system (a saccadic movement). Figure 5 shows an initial situation with both eyes (verged in a cube). An angular displacement is calculated and passed to its control software in order to move to other region (in this case to the cube on the left part of Figure 5). We could see that, while moving, its eyes could not stay so verged due to the motion algorithm used (one eye following the other). But, when the dominant eye stops the movement (reaches the target), the other eye verges immediately, now in the other cube. Final state of this sacaddic motion can also be seen in Figure 5. One can also note stereoscopy in the Figure. Some sort of attentional policies can be implemented later, based on the current features, to define targets.

6 Conclusions and perspectives

Currently, we have Gaussians, averaging, disparity and Sobel's operators implemented in Câmbio by using OpenGL ("C") directives. It is our aim to add as many different operators as possible, so that one can try different approaches for feature extraction and different policies for higher level processes. We yet intend to perform comparative studies of such operators using Câmbio's architecture. We also plan to add more complex objects to the world model and implement some more realistics physics laws to the world simulation, like the ability to bump into objects and, consequently, develop a policy to perform motion in such a populated environment. Finally we intend to implement, test and compare different attentional and feature extraction learning systems, like those presented in the references, as this is the main goal used for the development of this simulation.

References

[1] Robot Simulators on the Net. Accessed from: www.robotcafe.com/dir/Software/Simulators/ cswww.essex.ac.uk/Research/tuuv/simulators.html robotics.stanford.edu/users/sbenson/botworld.html www.cyberbotics.com/products/webots/

[2] L. M. G. Gonçalves, G. A. Giraldi, A. A. F. Oliveira, and R. A. Grupen. Learning policies for attentional control. *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99)*, November 1999.

[3] L. M. G. Gonçalves, R. A. Grupen, A. A. Oliveira, D. Wheeler, and A. Fagg. Tracing patterns and attention: Humanoid robot cognition. *The Intelligent Systems and their Applications*, 15(4):70–77, July/August 2000.

[4] L. Gonçalves and F. Silva. Control mechanisms and local perception to support automomous behavior in virtual animated agents. *Computer & Graphics Journal*, 25(6):965–982, December 2001.

[5] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Patten Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[6] M. M. H. Rajesh P.N. Rao, Gregory J. Zelinsky and D. H. Ballard. Eye movements in iconic visual search. *Vision Research*, 2002.

[7] R. P. N. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence Magazine*, 78(1-2):461–505, October 1995.

[8] I. A. Rybak, V. I. Gusakova, A. V. Golovan, L. N. Podladchikova, and N. A. Shevtsova. A model of attention-guided visual perception and recognition. *Vision Research*, 38(2):387–400, 1998.

[9] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence Magazine*, 78(1-2):507–547, October 1995.