Sonifying Robotic Trajectories with a Spherical Omnidirectional Vision System in the AURAL Environment

Artemis Moroni[1], Sidney Cunha[1], Josué Ramos[1], and Jônatas Manzolli[2]

[1]Robotics and Computer Vision Division, Renato Archer Information Technology Center
Rod. D. Pedro I km 143.6, Campinas, São Paulo, 13069-901, Brazil
[2]Interdisciplinary Nucleus for Sound Studies, University of Campinas
CP 6166, Campinas, São Paulo, 13083-872, Brazil
{Artemis.Moroni, Sidney.Cunha, Josué.Ramos}@cti.gov.br; jonatas@nics.unicamp.br

Abstract. This paper describes the OmniEye, an omnidirectional vision system developed to track mobile robots in AURAL, i. e., in a computational structured environment. AURAL aims to control the interaction between visual, sound and robotic information in a research for automatic and semi-automatic processes of artistic production. Different convex mirrors can be used to achieve an omnidirectional system. The use of a spherical mirror in this case introduces distortions in the image. A toolbox for the calibration of central omnidirectional cameras was used to obtain a first estimation for the imaging function. Then, a genetic algorithm was applied to adjust the coefficients of the imaging function. Experimental results and the application of the OmniEye for translating robotic paths into sound events in the AURAL environment are described.

Keywords: Omnidirectional vision, robotics, algorithmic composition.

1 Introduction

This development is part of the AURAL project, where a user draws a path in an interactive interface and transmits it to a mobile robot. An omnidirectional vision system, the OmniEye, occupies an important role in the AURAL, for it is the "observer" being used to feedback the robot localization [1]. Like others [2, 3, 4], AURAL belongs to that kind of systems that combine the behavior of mobile robots with sound events. In the AURAL, a robot tries to travel along the path in a follow up area, but it can be disturbed by other robots or obstacles while traversing it. The interaction of physical parameters and the presence of the mechanical bodies of the robots are potentially able to generate a complex sequence of interactive events. These events will be used to modify the performance controls of JaVOX, an evolutionary environment applied to sound production [5].

Different convex mirrors can be used to achieve an omnidirectional system. Parabolic, hyperbolic, spherical mirrors or even pre-designed surfaces with specific

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 354-365

desired properties can be applied. Yagi [6] compiled a literature review which shows the application of various types of omnidirectional visual systems. Such systems can be assembled, according to various models, using multiple cameras, which point to different directions, or even using a single free camera which rotates around a fixed axis [7]. From the possible ways of building an omnidirectional system, it was decided to assemble a spherical mirror. In spite of the fact that the spherical mirror does not present any special property [8], it is relatively easy to be built and can also be used in robotic navigation and tele-operation, having a low cost compared to hyperbolic mirrors.

The use of a spherical mirror, with no single effective viewpoint in the omnidirectional system, introduces distortions in the captured image. The processing of this kind of system can be carried out in two different ways: through an initial rectification of the image and further application of concerning techniques, or the handling of the omnidirectional image. The former approach is useful when the final result of the process is oriented towards the human interpretation of the image, while the latter avoids the need of a rectification and can be used for the automatic processing of the image by means of computational systems.

This paper belongs to the second group. Here, the development of a geometrical formulation for images, aims to determine a relation between the coordinates of the physical world and the coordinates of the pixels of a corresponding omnidirectional image. For this, given a captured image and the corresponding scene of the world, the modeling of the geometrical projection of this image is necessary to relate some measurements of interest.

Next, in sections 2 and 3, a description of the geometrical modeling for the development and calibration of the omnidirectional camera will be made. In section 4 the genetic approach to optimize the calibration function will be discussed. In section 5 some results will be shown and section 6 deals with the translation of trajectories into sound events. Finally, in section 7, a conclusion is presented.

2 Geometrical Modeling for the OmniEye

Specific algorithms to process images obtained with the omnidirectional system invariably require geometrical parameters from the optical system which is being applied. The analysis of the radial distortion introduced by the system camera-mirror in relation to a world scene is of fundamental interest to the spherical mirror, as well as the determination of the intrinsic and the extrinsic parameters of the mathematical model applied to the camera.

In this development, the catadioptric omnidirectional system is made up of a camera, a spherical convex mirror and a conical weight, assembled in a pendulum mount, which gives a vertical direction having good accuracy to align the camera and stabilize the set up. To cause a minimal obstruction in the image captured or, in other words, to obtain areas with minimal occlusions, nylon threads were used to fix the system. The optical axis of the camera was aligned with the optical center of the mirror, which was hung from the ceiling of the room, minimizing occlusions. The whole environment was captured in a single image. The set up is shown in Figure 1.

Fig. 1. The OmniEye with the spherical mirror, the camera fixed with nylon threads and the conical mass that make up the omnidirectional system.

The extrinsic parameters are the entries of vector **T** and rotation matrix **R,** totalizing 6 parameters. The intrinsic parameters are those necessary to determine the optical, geometrical and digital characteristics of the visualization provided by the camera. These parameters can be described by: (1) the geometric projection (characterized by the focal distance **f** of the lens and the pixel size); (2) the transformation of the coordinates of the camera-to-image reference systems and (3) the geometric distortion introduced by the optical system during the process. In spite of the geometric distortion, we have:

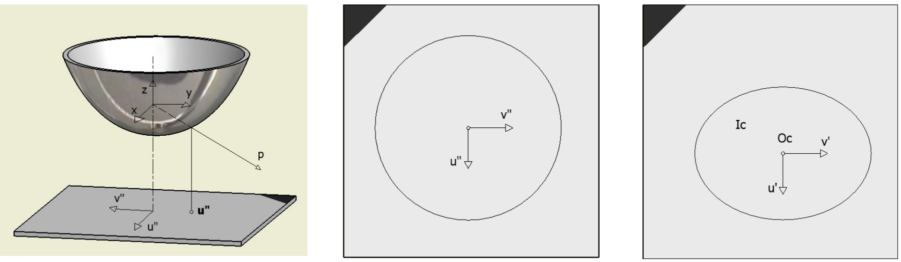
$$x_c = -(x_{im} - o_x)s_x \,;\, y_c = -(y_{im} - o_y)s_y \tag{1}$$

where (x_c, y_c) and (x_{im}, y_{im}) are the coordinates of the image point of the camera-image reference systems, respectively; (o_x, o_y) are the coordinates of the center of the image and (s_x, s_y) are the actual size of the pixel (in millimeters) in the horizontal and vertical directions, respectively.

The modeling of the catadioptric omnidirectional perspective camera system permits us to relate the coordinates of the image (in pixels with respect to the axis **u** and **v**) with the three-dimensional vector **p** whose origin is in the single center of projection of the mirror and whose end is at the point of reference in space. This projection model is based on Scaramuzza et al. [9].

In spite of the spherical mirror, it is assumed that there is a single center of projection (origin of vector **p**) which is also the center of the radial symmetry of the mirror with its optical axis. This approximation is assumed because only the central part of the image is actually used in the experiment. The localization of the objects will be made in a restricted area of the environment.

A system with a single center of projection is such that beams of light from the camera, reflected by the surface of the mirror, intersect each other at a single point (virtual point, origin of vector **p** as shown in Figure 2). Systems without a single center of projection, in contrast, are those in which the intersection between the described beams do not occur at a single point.



(a) (b) (c)

Fig. 2. (a) The coordinate system in the catadioptric case. (b) Sensor plane, in metric coordinates. (c) Camera image plane, in pixel coordinates. Pictures (b) and (c) are related by an affine transformation.

The construction of omnidirectional catadioptric systems employing lenses with hyperbolic, parabolic or elliptic mirrors assures the property of the single center of projection. For spherical mirrors this property can only be approached locally in the central area of the image. Thus, coordinate vector **p =** (x, y, z) and its projection **u =** (u, v) described in Figure 2 can be related as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \lambda \begin{bmatrix} u'' \\ v'' \end{bmatrix} \qquad (2)$$
$$\lambda \geq 0$$

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda \begin{bmatrix} \alpha \cdot u' \\ \alpha \cdot v' \\ f(\alpha.\rho') \end{bmatrix} \qquad (3)$$
$$\lambda, \alpha \rangle 0$$

Since **p** is a vector, a constant α can be included in f(u, v), since this latter function depends only on the radial distance $\rho^2 = u^2 + v^2$ of point **p** to the optical axis.

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \lambda.\alpha \begin{bmatrix} u' \\ v' \\ a_0 + \ldots + a_n \rho'^{n} \end{bmatrix} \qquad (4)$$
$$\lambda, \alpha \rangle 0$$

Therefore, the process of calibration consists of determining the coefficients of the polynomial expression, the intrinsic parameters given below, as well as the extrinsic parameters.

$$f(u'') = a_0 + a_1\rho'' + a_2\rho''^2 + a_3\rho''^3 + a_4\rho''^4 + \ldots \qquad (5)$$

By applying a spherical coordinate system, we obtain:

$$\begin{aligned} u &= \rho'' \cdot \sin(\theta) \cdot \cos(\varphi) \\ v &= \rho'' \cdot \sin(\theta) \cdot \sin(\varphi) \\ z &= f(\rho'') = \rho'' \cdot \cos(\theta) \end{aligned} \qquad (6)$$

If u and v are known, φ and θ can easily be found. u and v are extracted from the pixels of the image. Since φ and θ are known, the coordinates x and y, associated with the u and v pixel coordinates, can be evaluated for any desired plane z. In this case, the plane is the floor of the room, the (x, y) world coordinates of an image can be calculated, and consequently the path of a mobile robot.

3 The OmniEye Calibration

The toolbox [10] allows the calibration of any central omnidirectional camera or, in other words, cameras having a single center of projection. The calibration is accomplished in two different stages: initially, a set of images containing a chess pattern is captured from different positions and orientations in space. Then, the corners of the pattern are manually determined using the toolbox. The calibration is then automatically calculated by using the obtained data, with the help of a corner detector to improve the accuracy of the data.

Through the camera calibration the relationship between the pixels of the image and the 3D vector can be determined, as well as the origin in the single projection center and the end in the space points projected on the image, as shown in Figure 2.

However, even if the property of a single projection center is not exactly verified, the toolbox still provides good results using the calibration. The spherical mirror furnishes the possibility of a good estimation of a hyperbolic mirror in a restricted area of space, in the central part of the image. During the calibration with the toolbox, the degree of the polynomial used to map the pixels of the image, with the corresponding 3D points of the world, is requested. Experience has shown that polynomials of degree 4 are enough to describe the image-world mapping resulting from the optimization of the SSD function [9]. Once the coefficients in Equation 5 are determined, a spherical coordinate system was used to find any vector emanating from the omnidirectional image to the world.

The error incurred was of 8% in the lowest “maximum distance” although the result was very consistent. Aiming to obtain a better approximation, a genetic algorithm was applied to optimize the coefficients.

4 Image Function Estimation with a Genetic Algorithm

All evolutionary approaches share many features. They are all based on the general framework provided by J. H. Holland's original genetic algorithm (GA) [11]. In nearly every case, new populations of potential solutions to problems are created (here, the problem is of image function estimation), generation after generation, through three main processes: 1. By making sure that better solutions to the problem will prevail over time, more copies of currently better solutions are placed into the next generation. 2. By introducing new solutions into the population, that is, a low level of mutation operates on all acts of reproduction, so that some offsprings will have randomly changed characteristics. 3. By employing sexual crossover to combine good components between solutions, that is, the "genes" of the parents are mixed to form offsprings having aspects of both. With these three processes taking place, the evolutionary loop can efficiently explore many points of the solution space in parallel, and good solutions can often be found quite quickly.

In the genetic algorithm applied to estimate the calibration function, the chromosome of each individual of the population is coded in an array of length 8, where the first five elements contain the coefficients of the polynomial defined by Equation 5 being estimated. The sixth element of the array corresponds to the distance of the focus of the mirror. The seventh and eighth elements contain the coordinates x and y of the center of the image. It is worth to point out that the spherical mirror does not have a well defined focus. The values which are being investigated are those which better estimate the mapping from pixels of the image with the points of the world, whose coordinates are known.

4.1 The reproduction cycle

In the first experiment using the genetic algorithm, eight points P_i, i = 1 ... 8, were marked on the floor of the room for fitness evaluation. An image of the scene was captured using the omnidirectional system and the pixel coordinates (u_i, v_i) corresponding to each of the eight points were obtained from this image.

A tournament selection was applied to choose the parents for the next generation. The values obtained for each individual from Equation 6 were applied as parameters in Equation 7, used to estimate position P_i' for each point P_i. The distance d_i between each pair (P_i, P_i') was evaluated and D was assigned with the greatest d_i. The fitness F for each individual was evaluated as:

$$F = 1/D \qquad (7)$$

Therefore, what was investigated was the shortest “maximum image-world distance”. An arithmetic crossover was applied to the pairs of parents, followed by the Gaussian mutation [12]. The best individuals of the previous generation were included in the new one.

4.2 The results

Since the first experiment using the genetic algorithm, the distance has been 3% of the shortest maximum distance. It is worth to note here that there are some sources of errors. For example, the alignment of the camera with the center of the mirror in the vertical plane is very error prone. Aiming to obtain more points to improve the imaging function, a grid of points was drawn on the floor. Figure 3 shows the image that was used to obtain the pixels of the points of the grid and Figure 4 depicts the mapping.

The function fitness was evaluated considering 24 points. A better result was obtained with an error of 1% relative to the maximum distance; 5000 individuals; mutation rate = 15%; crossover rate = 30%. In this application, the result was satisfactory since the robot used in the experiment was a Nomad 200, having a 45 cm diameter and 85cm high. The height of the mirror from the floor was 2.9 m.

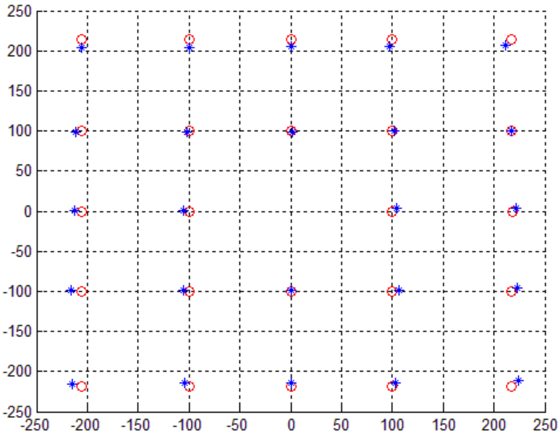


Fig. 4. The map that was obtained using the genetic algorithm with 24 points to evaluate the fitness. The circles are the points of the world; the crosses are the estimated points.

5 Tracking the robot

Functions of the OpenCV library are being used to track the robot. Initially, the vision system captures an image of the environment that will be used as a background image. This image is subtracted from all the other images which were obtained in real time. If no modification of brightness in the environment occurs, the result of the subtraction is a black image.

A high intensity light source is mounted on the robot. The lamp is lit after the background image has been captured. Each image captured with the omnidirectional vision system is subtracted from the background image and a thresholding function is applied. The result is a binary image (black and white), according to the threshold level applied. Operations of mathematical morphology (Top Hat, Opening, Closing) are then applied in the binary image. Next, a routine to find contours is used to obtain the location (pixels) in the image of the mark associated with the lamp of the robot. The coordinates of the pixels of the contour are then used to calculate the centroid of the mark. Next, the mapping function is used to evaluate the coordinates of the world associated with the coordinates of the centroid. These coordinates of the world are used as feedback to the robot concerning its position in the environment.

Figures 5 (a) and (b) show the results of the morphological operations with the omnidirectional images, pointing out the path traversed by two robots, the Nomad and a Roomba. In (a), a trajectory was sent to be traversed by the Nomad robot. In (b), the way traversed by a Roomba robot is shown. Note that the spiral performed by the Roomba, when turned on, is easily recognized in the upper part of Figure 5 (b).

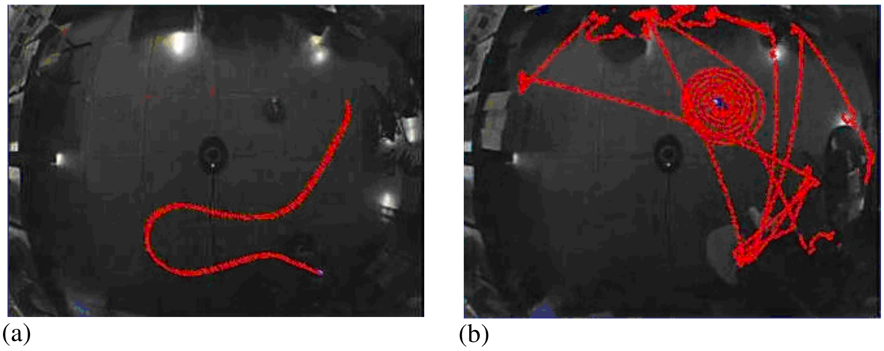

(a) (b)

Fig. 5. On the left, the path traversed by the Nomad robot, observed by the omnidirectional system. On the right, the path traversed by the Roomba robot.

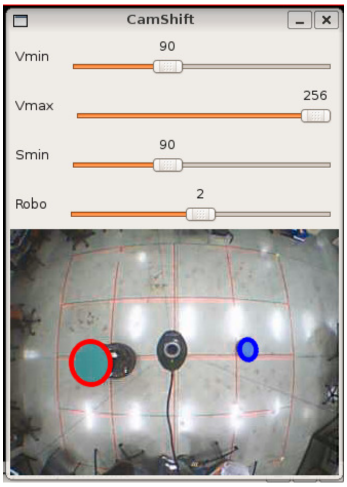
The developed code allowed to accomplish all the mentioned operations in real time recording video images at 30 fps, with all the robot path logs and localizations in the world referential, as seen in Figure 5. But the use of a light source for tracking the robot presupposes that there is not a great light variation in the environment. To overcome this limitation, another approach based on colors was applied. On each robot, a strongly-colored panel was fixed and a variation of the Camshift demo from OpenCV samples was applied. In short, once the program is launched, a rectangle on the panel to be tracked is selected with the mouse, in the image captured with the OmniEye. A color histogram is created to represent the panel. Next, the "panel probability" for each pixel in the incoming video frames is calculated. The location of the panel rectangle in each video frame is shifted. The new location is found by starting at the previous location and computing the center of mass of the panel-probability values within a rectangle. The rectangle is then shifted to its right over the center of mass. CamShift stands for "Continuously Adaptive Mean Shift" and is the based on Mean Shift algorithm [13]. The algorithm is called *continuously adaptive* and not just a *mean shift* because it also adjusts the size and angle of the panel rectangle each time it shifts it. It does this by selecting the scale and orientation that are the best fit to the panel-probability pixels inside the new rectangle location.

Figure 6 shows the CamShift interface, and the robots Nomad (red circle) and Roomba (blue circle). The coordinates of the center of mass of the circles in the image are evaluated and applied in the equation system 6 to evaluate the position of each robot in the area.

6 Translating Paths into Sound Events

The system that translates paths into sound events is also based on Evolutionary Computation. In this context, the MIDI protocol representation was used to code the genotype, like in the original development of VOX POPULI [14, 15]. This environment, initially developed in Visual Basic, was translated into Java, resulting in

JaVOX. The features described in this paper are codified both in JaVOX and VOX POPULI.



Vmin 90

Vmax 256

Smin 90

Robo 2

Fig. 6. On the left, the Nomad robot, inside the red circle; on the right, the Roomba inside the blue circle, both are tracked with the CamShift interface.

In both environments, a control area (pad) of the interactive interface enables the user to draw curves in a phase space, associating to each one of them a trajectory that guides the sound production. Figure 7 shows the curves drawn by the user in the graphic interface of VOX POPULI and the resulting sound sequence.

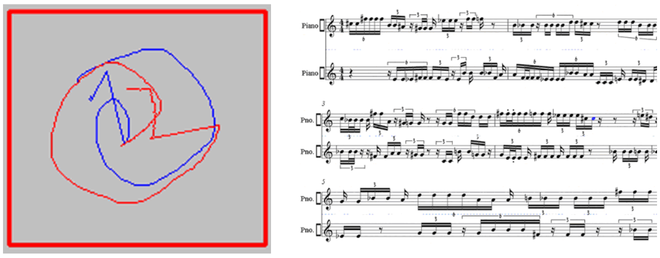
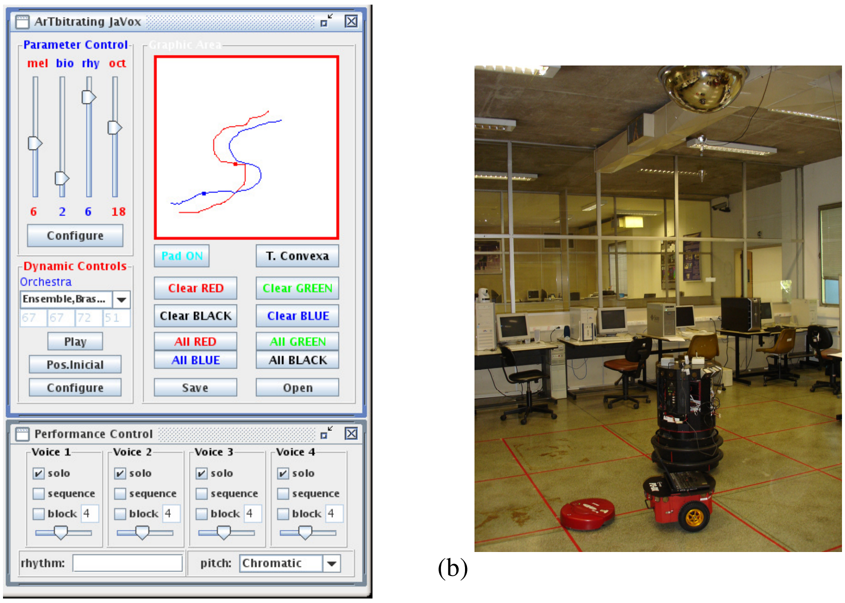



Fig. 7. Sound sequence resulting from the curves at the left, drawn in the interactive pad of VOX POPULI environment.

Similarly, in AURAL, the paths are drawn and transmitted to a mobile robot. The mobile robot traverses a structured area which is associated, through a bi-dimensional projection, with the area in the graphic interface that is approximated with MIDI events. The robot is observed by the OmniEye, that estimates the location of the robot in the area and sends it to JaVOX. The corresponding position is plotted in the interactive pad. The sequence of points describes the approximated path traversed by the robot.

Figure 8 shows the interface of JaVOX environment, where the lines drawn by the user direct the sound production in real time. Like in VOX POPULI, JaVOX links each line with the interface parameter controls. The red curves are associated with the melodic parameter (mel), in the x-component, and octave parameter (oct), or voice interval, in the y-component. These parameters guide the evolutionary sound production. In figure 8, the red line was drawn by the user and sent to the robot as a trajectory to be traversed. The blue line represents the path traversed, observed by the OmniEye. Both are used to control the sound production.



(b)

(a)

Fig. 8. The JaVOX interface. Below, the performance controls that are associated with events decurring from the interaction between the robots. On the right, the OmniEye with three mobile robots: a Roomba, a Pioneer and Nomad.

Besides the trajectories, JaVOX has other possibilities to control the sound production in real time. See, in the lower part of Figure 8.a, the Performance Control interface is shown. For each one of the four voices there are three controls named 1) solo; 2) sequence; and 3) block. These three modes of sound performing generate significant variations in the sound result and can be applied as a compositional strategy. The interaction of these controls with the dynamic behavior of the mobile robot, the OmniEye and eventually, the presence of other robots in the area, can generate a complex sound organization. The link process between the behavior of the robots in the structured area and the translation into sound was developed aiming to verify the capability of AURAL to create self organized sound textures departing from simple interactions between the mobile robots. A supervisor module TrajeCt (for *traject control*) receives the sequence of trajectory points from JaVOX and sends it to the Nomad robot. Communication between each part of the system is made by means of an Inter-process communication protocol (IPC). The path traversed by the mobile robots is captured by the OmniEye that provides the coordinates (Equation 1) and the criteria of behavior for performance control in JaVOX.

The interaction between the free navigation of the Roomba(s) and the path traversed by the Nomad generates a collective behavior (between the robots) that is used as a performance control in JaVOX. Figure 10 shows the omnidirectional system and three mobile robots. There may be four robots in the environment at the most, each one associated with a voice in JaVOX, but other robots can be linked using other interface controls.

7 Conclusion

Different convex mirrors can be used to achieve an omnidirectional system. Among the possible ways of building an omnidirectional system, a spherical mirror was selected because of its availability and low cost and also because it can be used in robotic navigation and tele-operations. A significant gain in precision was obtained by applying a genetic algorithm to refine the coefficients of the perspective projection function. The use of the previous model, originally developed for a hyperbolic mirror, was very convenient. Populations of different size were tried, and the convergence was quick.

This technique does not use any specific model of the omnidirectional sensor. The resulting device is easily reproducible and of low cost. The application of the OmniEye in the AURAL environment, besides the feedback, allows to record sessions to study the behavior of the robots. In the AURAL context, the OmniEye can be considered not only as a support for creative explorations, but also as a device to learn about “automatic aesthetics”. In either case, it helps the user and the computer to work together interactively in a new way to produce results that can not be produced individually.

Acknowledgements

We wish to thank the students Lucas Soares, Igor Dias, Igor Martins and Eduardo Camargo, who worked in the development of the OmniEye. We also thank the students Thiago Spina, Felipe Augusto and Gustavo de Paula, who worked with the robots Nomad and Roomba. We thank the researchers Rubens Machado and Helio Azevedo for their useful suggestions. We are also thankful to the technical support of Jonnas Peressinotto and Douglas Figueiredo. We thank the Scientific Initiation Program of the National Research Council (PIBIC/CNPq), CenPRA and the Interdisciplinary Nucleous for Sound Studies of the State University of Campinas (NICS/UNICAMP) for making this research possible. This research work is part of the AURAL project, supported by the Foundation for the Research in São Paulo State (FAPESP) process 05/56186-9.

References

1. Moroni, A., Cunha, S.: OmniEye: A Spherical Omnidirectional System for Tracking Robots in the AURAL Environment. In: 11th Computer Graphics and Artificial Intelligence Conference, pp. 109–118, Athens, (2008)
2. Manzolli, J., Verschure, P. F. M. J.: Roboser: a Real-world Musical Composition System. Computer Music Journal 3, 55–74 (2005)
3. Wassermann, K. C., Eng, K., Verschure, P. F. M. J., Manzolli, J.: Live Soundscape Composition Based on Synthetic Emotions. IEEE Multimedia 4, 82–90, (2003)
4. Murray, J., Wermter, S., Erwin, H.: Auditory robotic tracking of sound sources using hybrid cross-correlation and recurrent networks. In: IROS2005 - International Conference on Intelligent Robots and Systems, pp. 3554–3559 (2005)
5. Moroni, A. S., Manzolli, J., Von Zuben, F.: ArTbitrating JaVox: Evolution Applied to Visual and Sound Composition. In: Ibero-American Symposium in Computer Graphics, pp. 97–108 (2006)
6. Yagi, Y.: Omnidirectional sensing and its applications. IEICE Transactions on Information and Systems 3, 568–579 (1999)
7. Mei, C., Rives, P.: Single View Point Omnidirectional Camera Calibration from Planar Grids. In: IEEE International Conference on Robotics and Automation, pp. 3945–3950 (2007)
8. Svoboda, T., Pajdla, T.: Epipolar Geometry for Central Catadioptric Cameras. International Journal of Computer Vision. 49 (1), 23–37 (2002)
9. Scaramuzza, D., Martinelli, A. and Siegwart, R.: A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In: 4th IEEE International Conference on Computer Vision Systems ICVS'06, p. 45 (2006)
10. Omnidirectional Camera Calibration Toolbox for Matlab. http://asl.epfl.ch/~scaramuz/research/Davide_Scaramuzza_files/Research/OcamCalib_Tutorial.htm
11. Holland, J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
12. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin (1996)
13. Comaniciu, D., Meer, P.: Robust Analysis of Feature Spaces: Color Image Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 750–755 (1997)
14. Moroni, A., Manzolli, J., Von Zuben, F. J., Gudwin, R.: Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. Leonardo Music Journal, 10, 49–54, 2000.
15. Moroni, A., Manzolli, J., Von Zuben, F.J., Gudwin, R.: Vox Populi: Evolutionary Computation for Music Evolution. In: Bentley, P., Corne, D. (eds.) Creative Evolutionary Systems, pp. 205–221. Morgan Kaufmann, San Francisco, 2002.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 354-365