Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Obstacle Avoidance and Path Planning for Humanoid Robots using Stereo Vision

Kohtaro Sabe*, Masaki Fukuchi**, Jens-Steffen Gutmann**, Takeshi Ohashi**
Kenta Kawamoto*, and Takayuki Yoshigahara***

* Life Dynamics Laboratory Preparatory Office
** Networked CE Development Laboratories
*** Broadband Applications Laboratories
Sony Corporation, Tokyo, Japan
Email: {sabe, fukuchi, steffen, takeshi, kenta}@pdp.crl.sony.co.jp, takayuki.yoshigahara@jp.sony.com

Abstract— **This paper presents methods for path planning and obstacle avoidance for the humanoid robot QRIO, allowing the robot to autonomously walk around in a home environment. For an autonomous robot, obstacle detection and localization as well as representing them in a map are crucial tasks for the success of the robot. Our approach is based on plane extraction from data captured by a stereo-vision system that has been developed specifically for QRIO. We briefly overview the general software architecture composed of perception, short and long term memory, behavior control, and motion control, and emphasize on our methods for obstacle detection by plane extraction, occupancy grid mapping, and path planning. Experimental results complete the description of our system.**

Keywords-component; Humanoid Robot; Obstacle Avoidance; Stereo Vision; Plane Extraction

I. INTRODUCTION

Recently, we proposed *Robot Entertainment* as a new application for autonomous robots [1]. The aim of this proposition is to establish a new industry involving autonomous robots and artificial intelligence. We previously reported the development of a quadruped robot, AIBO, for pet-type applications [2], and a biped robot SDR-3X as a prototype for motion entertainment with dancing performance [3][4]. We further developed a biped robot SDR-4X as a prototype for a partner robot [5] where the emphasis is more on the capability of interaction with humans than it was in SDR-3X. One of the significant technical achievements of SDR-4X is its integration of many advanced technical features, such as real-time motion control, face recognition, speech recognition and others. This year we gave SDR-4X a nickname: QRIO, which stands for "quest for curiosity". A main technological target of QRIO is to autonomously explore and wander around in home environments as well as to communicate with humans.

Solving the problem of obstacle avoidance for a humanoid robot in an unstructured environment is a big challenge, because the robot can easily lose its stability or fall down if it hits or steps on an obstacle. For a wheeled robot, many solutions on this subject have been presented in the literature using ultrasonic sensors or laser range finders and they mainly detect walls and relatively large obstacles around the robot. In contrast, our strategy focuses on floor estimation, because in our view information about the floor is most important for a humanoid robot while walking. For this purpose, we developed a stereo-vision system and detect the floor plane using a randomized version of the Hough transform.

This paper describes our obstacle avoidance architecture allowing QRIO to walk safely around in a home environment. The following sections present an overview of QRIO and the general hardware and software architecture. Section IV described our approach to stereo matching. We then present our method of plane extraction from stereo data (Section V), the occupancy-grid map generation (Section VI), and our path planning method (Section VII). Experimental results are described in Section VIII and we conclude in Section IX.

II. OVERVIEW OF QRIO

The development of this entertainment robot is a unique challenge of uniting hardware and software technology together. QRIO is equipped with a system called "Real-time whole body stabilizing motion control" composed of 38 joints of the intelligent servo actuator (ISA: a uniquely developed actuator). In real time, it enables QRIO to walk adaptively on inclined and irregular terrain and allows the robot to re-stabilize immediately even when external forces affect its balance. Furthermore, a sub-system for real time step pattern generation realizes various walking patterns ranging from active and stable biped walking to dance performances.

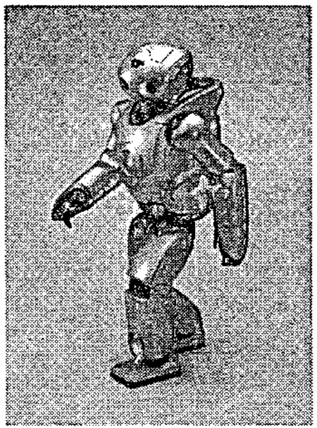


Figure 1. Entertainment biped robot QRIO

From the recognition point of view, QRIO is able to perceive its environment in three dimensions utilizing its stereo-vision hardware system. The stereo-vision system consists of 2 cameras as the robot's eyes and a FPGA module for stereo processing in the robot's head. Using its stereo camera QRIO can compute distance to objects, extract a floor plane and generate a path for walking around obstacles. Furthermore, ball and landmark detection using color segmentation as well as face recognition helps QRIO to understand the outer world. Last but not least, speech recognition and sound localization utilizing seven microphones as well as speech synthesis play a big role in the communication capabilities of QRIO. Audio and visual recognition results are memorized in a system called *short-term memory* that reflects the current environment. By utilizing its build-in wireless LAN, QRIO is also able to communicate with network computers and use them as a *remote brain.*

III. System Architecture

A. *Hardware*

Our stereo-vision system has been designed specifically for QRIO. It consists of 2 color CCD cameras (each 352x288 pixels), a FPGA module for stereo processing, two 8Mbyte SDRAM units, a flash ROM and an 8-bit micro processor (see Fig.2). The stereo-vision module receives a pair of images from the left and the right cameras, and computes disparity between them using block matching. The resulting disparity image is send to the main CPU board as a digital YUV video signal. Camera control parameters such as AGC, AWB, and stereo control parameters can be set through a special serial communication link (OPEN-R Bus [1]) between the 8bit CPU on board and the main CPU.

When designing a stereo-vision system, the base line between cameras is an important factor, because it affects precision of depth estimation, and the hardware exterior design. In consideration of the height of QRIO and the main applications for our stereo system, we chose a base line of 5cm. This allows for reliable floor estimation up to a range of 2m and reliable distance estimation of other objects in the range of 15cm to 3m.

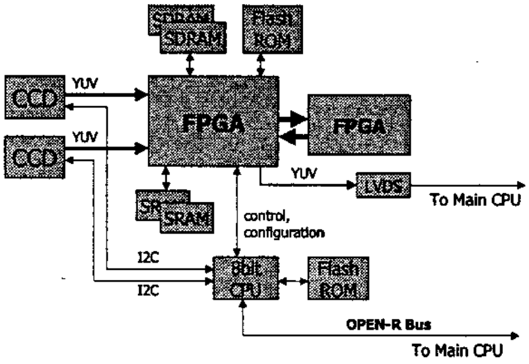


Figure 2. Small stereo vision hardware

B. *Software*

Fig.3 shows our software architecture for obstacle avoidance consisting of various software modules and their data flow. The system is built on top of OPEN-R, a system where software modules run in parallel and execution is driven by passing message between modules. We briefly describe each module below.

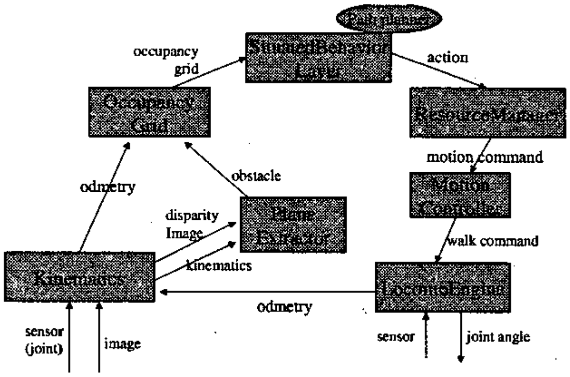


Figure 3. Software configuration

The *Kinematics* module receives image data from the stereo-vision system and joint angle sensor data from each actuator. All input is time-stamped by a frame number giving precise information about when the data has been taken. The module synchronizes vision and joint angle data for obtaining pairs with equal frame numbers. From the joint angles the kinematic transformation from the foot sole to the body center (fTb) and from the left camera to the body center (cTb) is computed. The module delivers disparity images with the corresponding kinematic transformations to the *Plane Extractor.* Synchronizing the inputs is important because of the delay in data acquisition and transfer. If frame numbers do not match, significant problems arise when results from image recognition are interpreted using the wrong values of the angular joints, especially while the head is shaking or when walking. In addition, the module computes and provides odometry information computed by integrating the fTb transformation for each step taken by the robot.

In the *Plane Extractor* module, the disparity image obtained from stereo-vision is first converted into 3D range data using parameters from stereo calibration. Then a Hough transformation is applied for finding planes in the 3D data. The most likely plane (the one with the highest score in the Hough transform) is selected and if it approximately corresponds to the information from kinematics, is considered as the floor plane. Finally all 3D measurements are converted to the floor coordinate system. The accuracy of floor estimation by this plane extraction method is considerably better than when directly using the joint kinematics, especially on a soft carpet or when some of the joint sensors are poorly calibrated. Section V describes more details about our method for plane extraction.

The *Occupancy Grid* module is invoked either when receiving new 3D information from the *Plane Extractor* or on odometry information from the Kinematics module. It integrates information into a 2D occupancy grid of size 4x4m around the robot. Each cell in the grid holds a probability that the cell is covered by an obstacle and the time the cell was last updated. Section VI presents more details of the underlying probabilistic methods.

The *Situated Behavior Layer* (SBL) [10] is a control mechanism where the behavior of the robot is determined autonomously according to internal states and external observations. Part of SBL is a *path-planning engine* that, given a goal point, computes a collision-free path leading to the destination. Our path-planning approach employs an A* search on potential fields imposed on the occupancy grid and is described in more detail in Section VII. Usually path planning only focuses on how to find an optimal (e.g. shortest) path to the destination, but in a real and unknown environment, it is also important to decide on where to explore and look in order to accomplish finding a path to the goal. Therefore, when following a path, we adaptively change the head motion of the robot according to the situation present in the occupancy grid and the current best path to the destination. The path planning system then generates walk and head motion commands which are send to the *Resource Manager* which in turn send them to the *Motion Controller* and the *Locomotion Engine.*

IV. STEREO COMPUTATION

Our stereo-vision system receives a left and a right image from the two CCD cameras in the robot head. By calibrating the cameras, image distortion is removed and camera parameters such as focal length and the principal point are obtained. These parameters are useful later for computing 3D range data.

From the undistorted images, disparity is calculated for each pixel in the left image by searching for the corresponding pixel in the right image using block matching along the epipolar line. This algorithm easily causes incorrect matches when there is either no texture or a repetition of similar texture patterns. For this reason, an additional reliability image is calculated following criteria to reject results on above conditions. After block matching has been carried out, the matching score (the normalized correlation) is calculated by interpolating scores near the highest peak. The sharpness of this peak corresponds to the available texture around this pixel and thus can be used as a reliability value for the disparity calculation. Furthermore, the matching score is compared with neighboring scores. If there are other peaks with similar matching scores then the disparity computation is ambiguous (repeating pattern) and the reliability is set to a low value.

Fig.4 shows an example. On top is the left input image, lower left shows the disparity image, and lower right contains the reliability image. In the reliability image, bright pixels have high and dark pixels have low reliability. Note that the textureless white paper at the lower left part of the image gives low reliability. The dark brown floor carpet gives higher reliability near the camera where texture is more visible.

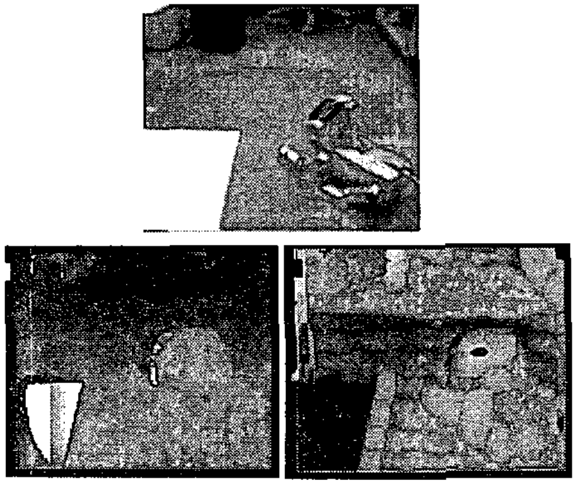


Figure 4. Input image (top), and disparity and reliability image (bottom)

V. PLANE EXTRACTION

A. Hough transform

For finding the floor plane, disparity is first converted into 3D range data using the parameters from camera calibration and then a Hough transformation is applied to all data points. In general, the Hough transformation is a method for estimating the parameters of a surface from a set of measured samples. If the number of sample points and parameters to be estimated is large, the direct application of the Hough transformation might be computationally expensive. In this case utilizing the randomized Hough transformation can be advantageous. This method randomly selects sets of data points from which the surface parameters can be directly computed and records the result in a table. If many data sets yield the same parameters, a high score for these parameters is obtained.

Plane extraction using randomized Hough transformation has previously been proposed by Okada *et al.* [11]. Our method differs to theirs in the following ways.

- All 3 points on the estimated plane are randomly selected so that parameter statistics can be obtained quickly.

- The parameter space (θ, $\phi \cos \theta$, d) is used instead of (θ, ϕ, d) to ensure fairness of voting.

- Reliability values are used as weights for voting.

- The peak is computed as a weighted mean for obtaining higher precision.

- A multi-layer image is used for coarse to fine estimation and a criteria for stopping voting is introduced at each layer.

Fig.5 shows a flow chart of our plane extraction method.

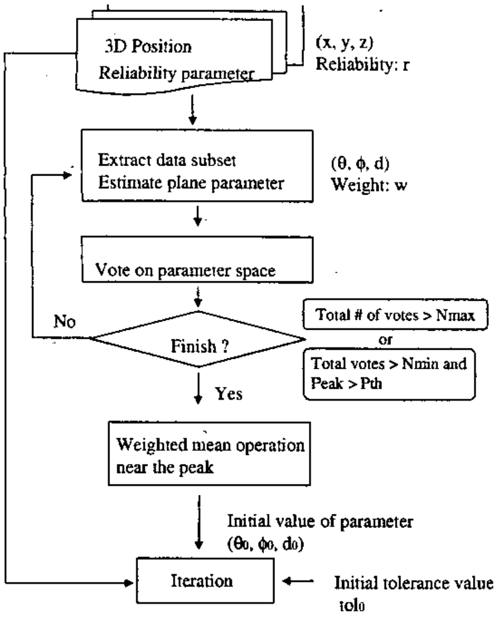


Figure 5. Flow chart of plane extraction

B. Results

Fig.6, left, shows the image of QRIO's left camera in a standing posture with the head looking down about 30 degrees. On the right all pixels with a distance less than 10mm to the detected plane are shown. Pixels with a range of 2 m or more are omitted. Note that some distant pixels fail to lie on the plane, however, most pixels successfully match the floor.

Fig.7 contains 3 histograms of the floor-distance of pixels depending on how far the data points are from the robot: 30cm to 50cm, 50cm to 100cm, and above 100cm. As the histograms show, the floor plane near the feet can be estimated within an error of +/-10mm. For the plane farther than 1m, estimation stays within the range of +/-30mm. Theoretically, the resolution of depth measurements at a distance of 1.5m is over 80mm, however, given the angle θ of the optical axis with the floor plane, the distance from the plane is proportional to $\Delta d \cdot sin\theta$, which for small values of θ reduces the error to the plane significantly. For the task of floor extraction this is advantageous since the larger the depth is, the more acute θ is.

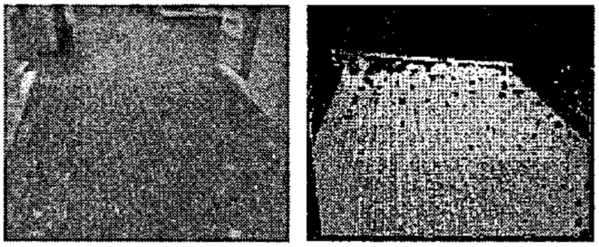


Figure 6. Floor image and extracted result

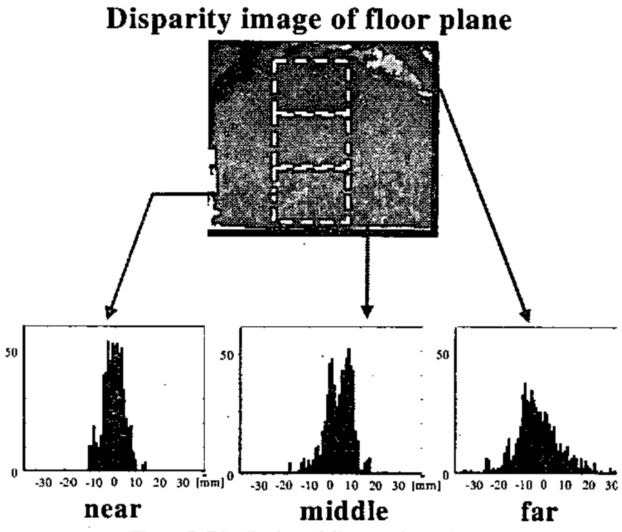


Figure 7. Distribution of distance from plane

VI. Environment Map

After applying floor detection explained in the previous section, obstacles and regions the robot can walk on can be found. However, in general it is difficult to decide from a single observation with a limited field of view, the action the robot should carry out next. Humans have the ability to memorize former observations and to perceive an environment by integrating observations over time. We follow this notion and introduce an environmental map where all observations and motions are integrated.

The environmental map is a 2-dimensional occupancy grid centered on the current position of the robot (egocentric coordinate system). We maintain the (global) robot orientation and a small relative (x y) location of the robot within the cell at the center of the grid. Each grid cell contains the probability that an obstacle covers the cell and the time the cell was last updated. Initially, all cells are set to a probability of 0.5 and time 0.

A. Observation model

Various methods for updating an occupancy grid from observations are compared by Wijk [6]. Similar to Wijk, we also compared the Bayesian update method, the Dempster Shafer method, and the Borenstein method [9]. It turned out that Bayes and Dempster Shafer gave very similar results. The method of Borenstein seems to be advantageous when obstacles are moving in the environment. In the end, we decided to employ the Bayesian method since most of the obstacles (walls, etc.) are stationary and Bayes' method gave superior results to Borenstein's for static obstacles.

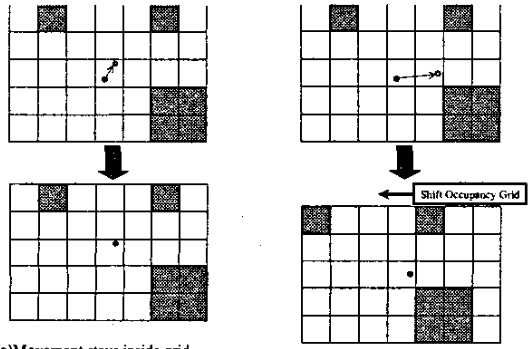
B. Motion model

The coordinate system of the occupancy grid and the robot position and orientation within the grid are updated on robot motion. Robot motion is defined as a coordinate transformation from one foot to the other whenever the robot performs a step. From this transformation a displacement and a change in orientation can be derived and applied (added) to the position and orientation of the robot in the grid. If the robot stays within the cell at the center of the grid, no further actions have to be performed (Fig.8 (a)). However, if the robot crosses the cell's boundary, we shift the grid by the number of cells the robot passed in each direction (Fig.8 (b)). In our implementation shifting the grid is actually performed by changing an offset into the grid data array so that no data has to be moved physically. After shifting the coordinate system of the occupancy grid, new cells at the border are initialized.

Given the cell size *CS*, the robot position (*Bx*, *By*) inside the grid, and the displacement (*dx*, *dy*), the amount of shift (*Sx*, *Sy*) can be computed as:

$$(Sx, Sy) = \left(\left\langle \frac{Bx + dx}{CS} \right\rangle, \left\langle \frac{By + dy}{CS} \right\rangle \right), \qquad (1)$$

where <*v*> returns the largest integral value not greater than *v*. The new robot position inside the grid then becomes:



(a)Movement stays inside grid

(b)Movement crossed grid boundary

Figure 8. Update method of Occupancy Grids (Shift operation)

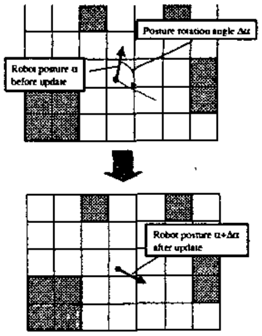

Posture rotation angle Δα
Robot posture α before update
Robot posture α+Δα after update

Figure 9. Update method of Occupancy Grids (Rotate operation)

$$(Bx, By) \leftarrow (dx - CS \cdot Sx, dy - CS \cdot Sy). \qquad (2)$$

Finally, the change in orientation *Δα* is simply added to the global orientation of the robot (Fig. 9).

VII. Path Planning

As the occupancy grid reflects the environment around the robot, it can be used to find a path to a goal point on which the robot can safely reach its destination.

We define each cell of the occupancy grid as a node connecting to neighboring cells and define the path planning problem as a search problem on this graph. Graph search is realized using the A* search algorithm. A potential field calculated from the obstacle probabilities is used as an evaluation function. The potential *u(n,n')* at node *n* to an obstacle *n'* is calculated as a repulsive potential from the obstacle probability *P(n')*:

$$u(n, n') = \frac{d_0 P(n')}{\max(d(n, n'), d_0)}, \qquad (3)$$

where *d(n,n')* is the Euclidean distance between *n* and *n'* and d_0 is a safety margin to ensure the robot does not hit any obstacle. The potential *U(n)* of a node *n* is then chosen as the maximum of *u(n,n')* over all nodes *n'* close to node *n*:

$$U(n) = \max_{n' \in N(n)} u(n, n') \qquad (4)$$

Our evaluation function for the A* search is:

$$f(n) = g(n) + \alpha \cdot h(n) \qquad (5)$$

$$g(n) = \sum_{n_s}^{n} U(n) + L(n_s, n) \qquad (6)$$

$$h(n) = \hat{L}(n_g, n) = d(n_g, n), \qquad (7)$$

where *g(n)* resembles the path cost from start node n_s to the current node *n*, *h(n)* is the estimated path cost from the current node *n* to the goal node n_g and α is a weighting factor. $L(n_s,n)$ is the path distance between n_s and *n* and $L(n_g,n)$ the path distance between *n* and n_g.

The evaluation function *f(n)* reflects the optimal estimated cost for passing through node *n*. By expanding nodes *n* with minimal *f(n)* value, the optimal solution can be found. Finally, by searching for shortcuts along the found path, a smooth walking trajectory is realized.

VIII. Example of Execution

Fig.10 shows a situation of QRIO avoiding obstacles on a stage presented to the public at the ROBODEX exhibition in Yokohama, Japan in 2002. In this demonstration, QRIO is starting on the left side and walking to a goal point at the right end of the stage.

Fig.11 displays snapshots of the occupancy grid calculated on-line at the positions corresponding to the scenes of Fig.10. The size of each occupancy grid is 4m x 4m with a cell size of 4cm making the grid 100x100 cells large. White cells correspond to the floor while dark ones refer to obstacles. Blue areas indicate unobserved terrain and an icon in the center of the grid shows the robot pose.

The size of the stage is 1m x 2m and gives a rough idea of the shape and positions of obstacle in the occupancy-grid snapshots. As these results suggest, the environment around the robot is well recognized and a safe, collision-free path around the obstacles and on the stage is found.

IX. Conclusion

By the development of a series of components and techniques, namely a small stereo vision system, the recognition of floor and obstacles using plane extraction, the representation of an environmental map based on these observations and robot motion, and the generation of a walking path on the environmental map, the autonomous mobility for the biped robot QRIO in the home environment is realized.

The environment is represented in a robot centric coordinate system without making any structural assumptions about the surrounding world. We therefore believe, our approach is well suited for many different a home environment where no *a priori* information about the environment is given. The limitation of our system is that the environment has to contain enough texture in order to obtain reliable stereo data.

In our future work, this local navigation system is combined with global map information so that QRIO can act intentionally using the knowledge of the environment.

References

[1] M. Fujita and K. Kageyama, "An Open Architecture for Robot Entertainment", in *Proc. Inernational Conference on Autonomous Agents*, pp. 435-440, 1997.

[2] M. Fujita and H. Kitano, "Development of an Quadruped Robot for Robot Entertainment", *Autonomous Robots*, vol. 5, pp.7-8, Kluwer Academic, 1998

[3] T. Ishida, Y. Kuroki, J. Yamaguchi, M. Fujita and T.T. Doi, "Motion Entertainment by a Small Humanoid Robot Based on OPEN-R", in *Int. Conf. on Intelligent Robots and Systems (IROS-01)*, pp.1079-1086, 2001.

[4] Y. Kuroki, T. Ishida, J. Yamaguchi, M. Fujita and T.T. Doi, "A Small Biped Entertainment Robot", in *Proc. Int. Conf. on Humanoid Robots*, pp.181-186, 2001.

[5] M. Fujita, Y. Kuroki, T. Ishida and T.T. Doi, "A Small Humanoid Robot SDR-4X for Entertainment Applications", in *International Conference on Advanced Intelligent Mechatronics*, 2003.

[6] Ole Wijk, "Triangulation Based Fusion of Sonar Data with Application in Mobile Robot Mapping and Localization", *PhD thesis*, Royal Institute of Technology, Stockholm, Sweden, 2001.

[7] A. Elfes, "A Probabilistic Framework for Robot Perception and Navigation", *PhD thesis*, Carnegie-Mellon University, 1989

[8] D. Pagac, E. Nebot and H. Durrant-Whyte, "An evidential approach to map-building for autonomous vehicles", *IEEE Transactions on Robotics and Automation*, 14(4), 623-629, 1998.

[9] J. Borenstein and Y. Koren, "Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance", *IEEE Transactions on Robotics and Automation*, 11(1), 132-138, 1995.

[10] M. Fujita, Y. Kuroki, T. Ishida, T.T. Doi, "Autonomous behavior control architecture for entertainment robot SDR-4X", in *Int. Conf. on Intelligent Robots and Systems (IROS-03)*, Las Vegas, 2003.

[11] K. Okada, S. Kagami, M. Inaba, H.Inoue, "Plane Segment Finder : Algorithm, Implementation and Applications", in *Int. Conf. on Robotics and Automation (ICRA-01)*, 2001.

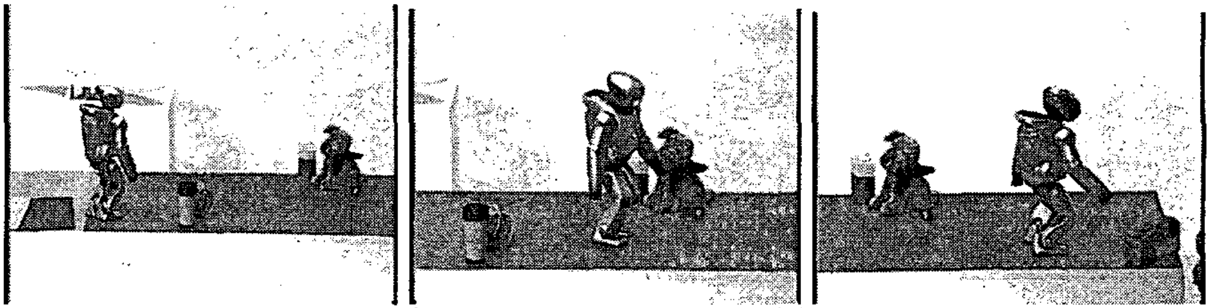


Figure 10. QRIO walking around obstacles on the stage of RoboDEX 2002

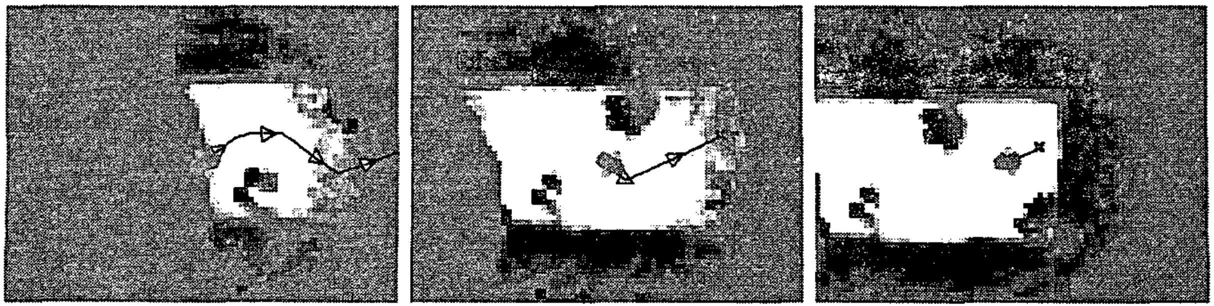


Figure 11. Snapshots of occupancy grid and planned path (indicated by arrows) of the scenes in Fig.10