Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Rapid Development System for Humanoid Vision-based Behaviors with Real-Virtual Common Interface

Kei Okada*[1], Yasuyuki Kino*[1], Fumio Kanehiro*[2],
Yasuo Kuniyoshi*[1], Masayuki Inaba*[1], Hirochika Inoue*[1]

*[1] *Department of Mechano-Informatics, The University of Tokyo,*
*[2] *National Institute of Advanced Industrial Science and Technology, METI*

Abstract

This paper describes a rapid development system for vision-based behaviors of humanoid which consist of both real robot and virtual robot in simulation environment. Vision based behavior of a humanoid is very complex and difficult to develop, therefore simulation environment to develop it rapidly and efficiently, and to verify or to evaluate it is required.

Previous simulator in robotics is limited to verify dynamics motions such as walking, or to plan or learn in simple environment, however our system is able to simulate vision based behavior, i.e. motion, perception, behaviors of a robot as a whole, since it can simulate both dynamics and collision in an environment, and motors and sensors including a view of a robot.

We regard realtime-ness of simulation as a important so that so that users can develop vision based behaviors rapidly and efficiently. It has Common Interface API to share behavior software between real and virtual robot, Moreover, visual processing functions such as color extraction, depth map generation are available.

As a result, vision based behavior consists of local map generation, planning, navigation of the humanoid in simulation is presented. We also show that software developed in the simulation environment, is able to apply to the real robot.

1 Introduction

Current complex systems such as a circuit or a car, require a simulation environment to develop efficiently or to verify and evaluate the developed system. On the other hand, the usage of a simulation in robotics researches is limited to precise dynamics simulators to verify dynamic motions such as walking or balancing of biped robots[1] or simple grid environment simulators for high level reasoning such as a planning, a navigation and a learning[3, 2].

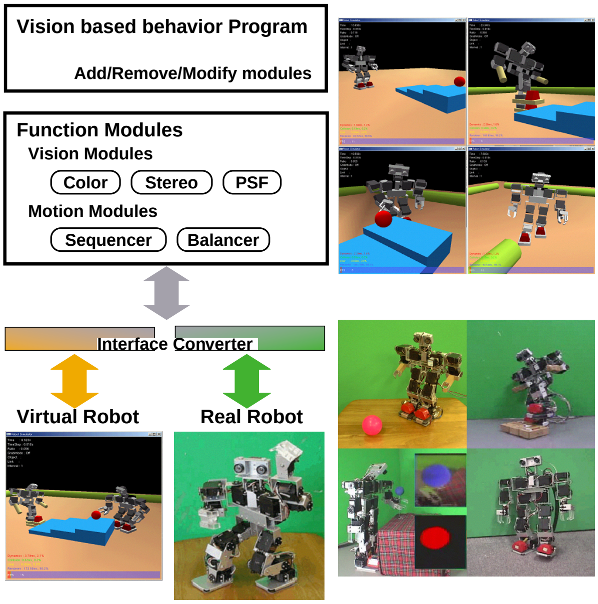


Figure 1: *Robot system for vision-based behavior*

It is needless to say, one of the essential aspects of robotics is that vast area of software must be dealt with, includes servo control, motion control, behavior control, sensor signal processing, perception, understanding, and so on. A simulation in robotics deal with one aspect of the behaviors of robots.

Recently, some researchers have developed a system which integrate vision system and behavior system in order to realize vision based behaviors of robots in real-world[4, 5, 6].

However it is difficult to develop such behaviors using real robots, since it has the risk of damaging robots or it is not easy to prepare, verify and evaluate the experiment.

To develop vision-based behaviors rapidly and efficiently, a simulation environment is required. In this paper, we describe a rapid development system for vision-based behaviors of humanoid which consist of both real robot and virtual robot in simulation environment. The simulation environment presented in this paper has following features:

1. It can simulate whole aspect of vision-based behaviors, dynamics and collision in environment, and motors and sensors including a view of robots, where previous simulation environment is limited to precise physical dynamics to verify dynamics motions such as walking, or simple grid environment for high level reasoning such as a planning or learning.
2. It can simulate vision-based behaviors in realtime in order to develop them rapidly and efficiently, where previous one is very slow which regards accuracy as important rather than realtime-ness. Since, even very accurate simulation is realized, it is differ from real environment. We believe that behavior-level of robot software must deal with such difference with error recovery of robust sensor feedback.
3. It can share the behavior programs between real robots and virtual robots, by using Common Interface API of a robot hardware.
4. It can simulate vision processing functions such as color extraction and depth map generation, which is also apply to a view of real robot.

This robot system enables users to develop and verify vision based behaviors of a humanoid robot in simulation environment without using a real robot at a risk of damaging it. The overview of developed system is shown in *Figure 1.*

MITI's HRP(Humanoid Robot Project) also developed a system with real robot and simulation environment[7, 8]. Both have dynamics and collision simulation of an environment, motor and sensors simulation includes a view of vision, interface architecture to share codes between real and virtual robot.

However, our system deal with visual processing in addition to view simulation to develop vision based behavior of a robot, such as finding the red ball by color extraction, generate depth map of a view using stereo vision, detecting floor regions from by planner surface detection based on stereo vision, visual tracking, map building from visual information, path planning, catch the object and so on, while HRP system deal with ZMP based walking experiment or collision check experiment.

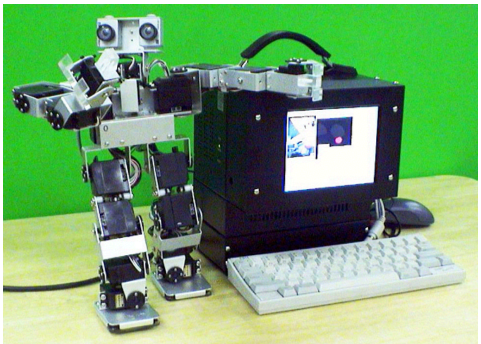


Figure 2: *Stereo vision equipped 18 DOF Humanoid Robot : Kaz*

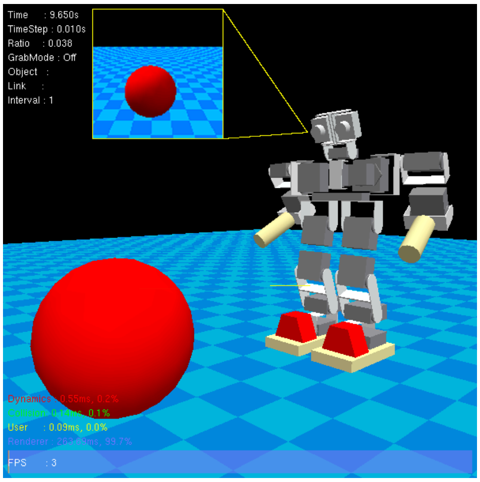
This paper is organized as follows. Section 2 overviews our real humanoid and virtual humanoid environment. Section 3 presents the mechanism to share vision based behavior programs between real and virtual robot. Section 4 shows vision based map building and navigation behavior s an example of a simulation environment, vision based a ball catch behavior also presented to demonstrate that software developed in simulation environment is able to apply to a real robot.

2 Humanoid Robots

2.1 Desktop Humanoid for Vision-based Behavior

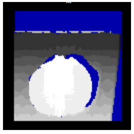
We have developed 18 DOF humanoid robot with stereo vision sensor named 'Kaz' as shown in *Figure 2*. The 18 DOFs consist of 4 DOFs at each leg, 3 DOFs at each arm, 1 DOF at each hand and 2 DOF for the head. The height of the robot is about 34 cm and weight is about 1.6 kg. This robot was developed based on the approach of Remote-Brained Robot[9].

Servo module and motor driver. Joints of this robot are servo modules (S9204, Futaba Corporation) for R/C model toys, which has a geared motor. Original servo module has proportional analog circuit inside with 4.8 V power input. However, we developed original motor driver circuit (JSK-D04) with FETs which can input up to 12 V power supply. The size of this circuit is 36.5 mm × 16.5 mm, therefore the circuit is able to install inside a servo module. The circuit has two inputs and one output signal. One input is binary signal which indicate a direction of a rotation, the other one is pwm signal to control speed of rotation. Output signal is potentio data. From our experiment, measured maximum torque of original circuit is 0.97 Nm at 4.8 V and that of developed circuit is 2.16 Nm at 10 V.

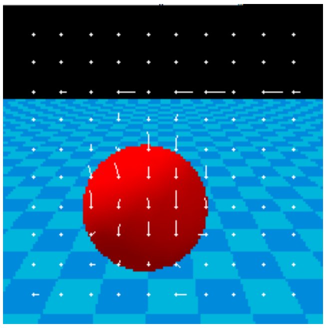


(A) Robot/View simulator

(B) Color extraction



(C) Depth generation



(D) Optical Flow

Figure 3: *Simulation environment for vision based behavior*

Onbody Microprocessor for servo control. The robot has microprocessor on body for control motors. The processor is Hitachi H8/2128, an 8-bit micro processor. This processor control low-level servo process. Target angle, gain of servo modules are controlled through serial connection from a outside PC to the processor.

Stereo vision camera. This robot has two cameras with wide view conversion at the head to perform stereo vision processing. The camera is CK-200 of Keyence Corporation.

2.2 Virtual Robot and Simulation Environment

Figure 3 shows a snapshot of our simulation environment 'FAST'[10] when the red ball is falling. This software is able to simulate collision, dynamics and the field of vision of the virtual robot, therefore the ball falls and bounces at the ground, generate the view, find ball using visual functions such as a color extraction, control joints of the robot to track the ball.

The main feature of FAST software is its speed. It is able to perform dynamics simulation and collision check of the 18 DOF humanoid robot in 0.40 msec and 0.07 msec respectively using PentiumIII 1.13 Ghz machine running Windows 2000 OS. This real-time simulation enables us to simulate vision based behavior of a robot.

Figure 3(A) is simulation environment and simulated view of vision (B) is the color extraction result of simulated view. RGB values of a simulated view image are converted to YIQ values to extracted red color. (C) is the result of depth map generation which takes a stereo simulated view as an input. In depth map generation, block matching approach is adopted. (D) is the result of optical flow generation.

Dynamics Simulation. FAST utilize Vortex toolkit[11] which is physics-based rigid-body dynamics, collision detection simulation developed by Critical Mass Labs. Vortex was developed for real-time application such as video games, therefore the speed of dynamics is fast, though the accurate of dynamics simulation is not so accurate as dynamics simulator used for robotics research so far, as DADS or ADAMS etc.

View Simulation. In FAST software, the view simulation of the robots is simple performance with lighting configuration and diffusion effects of object faces compares to other view simulation of robots. However we believe that in vision based behavior level simulation of robots, real-time view simulation is more important than realistic view simulation, and similarity between the results of visual processing in real world and simulation world is more important than the view of the robot in real and simulation world.

Our experiment up to present convinced us that the results of visual processing using the simulated field of vision of the robot is enough for developing vision based behavior of robots.

3 Interface API to Share Behavior Software between Virtual And Real Robot

To develop a vision based behavior software efficiently, it is important that the software developed in simulation environment can be applied to a real robot without porting, that is the same software applied to both a virtual robot and a real robot. To achieve this end, Interface API, an abstraction of robot controller and sensors, has been proposed[12]. In this section, we present our extension of Interface API to visual information.

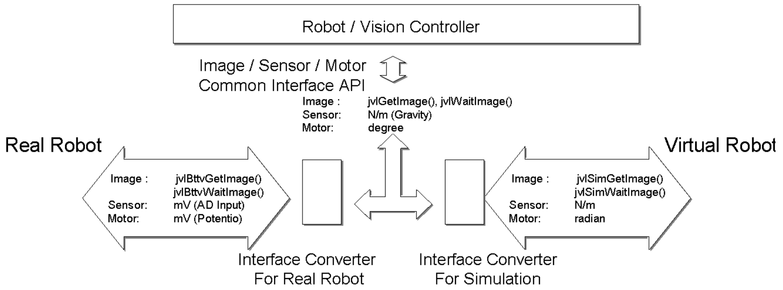


Figure 4: *Interface API to Share Behavior Software between Virtual And Real Robot*

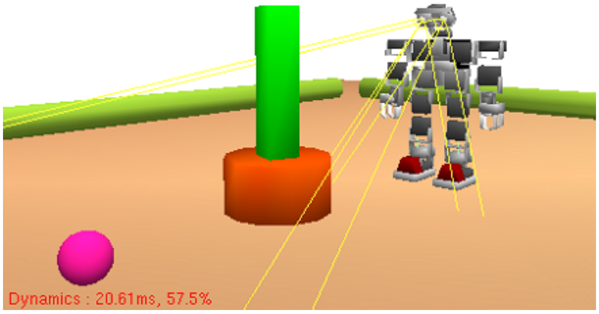


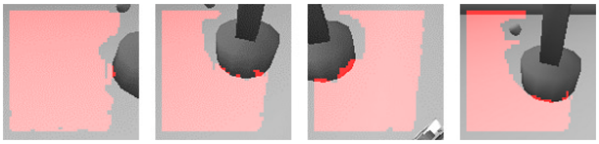
Figure 5: *Experimental Setup: Task of the robot is to approach the ball and catch it*

This architecture, illustrated in *Figure 4*, a behavior software uses Interface API to control a robot or receive sensor data including visual information. The system has Interface Converter for real robot and virtual robot which convert Interface API to hardware information of real robot or virtual robot.

4 Developed Vision based Behavior in Simulation Environment

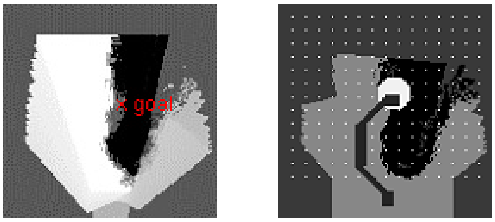
This section describes an experiment of vision based behavior of a humanoid robot as an example of this system. In this experiment shown in *Figure 5*, the task of the robot is to catch the ball. The robot has to walk toward the goal with obstacle avoidance and executes a ball catch behavior. To achieve this, path planning based on map with floor and obstacle information required.

In this experiment, the robot generate its surrounding map information by floor detection function using three-dimensional visual processing, then path planning technique is applied to generate path information from the current position to the goal position.



(A) Plane Segment detection results

(B) Floor recognition results using PSF



(C) Map generation result (D) Path planning result

Figure 6: *Visual processing and local map representation of the experiment*

4.1 Map Generation using 3D Vision

Floor Detection from a View of robot using PSF algorithm. To realize walking behaviors of a robot with obstacle avoidance, path planning using map information is required. Moreover, the robot required to generate this map using visual processing function to adapt to unknown environment. For a map of walking robot, recognition of a floor in environment is important. We apply our Plane Segment Finder(PSF) algorithm[13] which detect three-dimensional planner surfaces from visual input to recognize a floor.

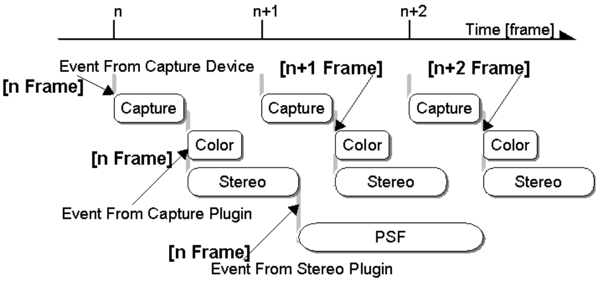


Figure 7: *Frame Information Messaging in Visual Processing Modules*

The result of PSF is shown in *Figure 6*(A), Red color regions in input images are the planer surface detected by PSF. From this result, PSF is able to detect floor regions.

Conversion of Coordinates to Generate Local Map. Floor detection results from a view of vision is represented in view coordinates whose origin is at eye. To generate local map information, the map should be represented in foot coordinates whose origin is at the sole of the foot. To convert the coordinates, it requires posture information of the robot when the view which is used for floor detection is captured.

Images in *Figure 6*(A) is captured while the robot looked around. Then floor detection results of each image are converted to foot coordinates representation. Floor detection results represented in foot coordinates are shown *Figure 6*(B). The origin of a map is bottom of center of the image. White regions are floor region, black regions are obstacles, gray regions are the out of the sight of the robot.

Frame Information Messaging in Visual Processing Modules. To convert coordination from a view to a foot, it requires posture information of the robot at a point of the time the view which is used for floor detection is captured. Generally, visual processing takes time. Hence, when the robot generate local map by looking around, the posture information when the PSF detect the floor is not as same as the posture when the view which is used for floor detection is captured.

Figure 7 shows how to deal with this problem. First capture plug-in module stores the frame information and posture information at present. When capture plug-in ends, it broadcast a signal to next plug-ins, such as stereo plug-in to generate depth map information and color plug-in to extract interest color regions. Frame information stored also broadcast to other plug-ins. When PSF plugin finish floor detection process, it uses broadcasted frame information to refer posture information and converts view coordination to foot coordination.

Map Merging to Generate Local Map for Path Planning. Since a view of a robot is limited, detected floor region at each time is also limited. Hence, An algorithm to merge map at each time to generate local map is required. *Figure 6*(C) is local map which is the result of merging each map in (A) or (B).

Black regions are obstacles, gray regions are the out of the sight of the robot. White regions are floor region, and brighter regions indicate high confidence of floor.

For path planning, the map is grided and each grid has information whether the position is floor or obstacle. Path planning results shown in *Figure 6*(D)

4.2 Behavior Control based on Vision and Map Information

This section describes how to control behaviors of the robot according to visual information and map information.

go-forward go forward certain distance by walking.

turn-right turn right 45 degree.

turn-left turn left 45 degree.

catch-ball catch a ball on ground by hand. the ball is assumed to be located on 2cm ahead of a robot.

search-ball look around to find a ball

track-ball track a ball by controlling its head

First, the robot generate local map and plan the path from current position to the goal position. Then according to the planned path, the robot execute `go-forward`, `turn-right` or `turn-left`. Before execute `go-forward`, robot examine if there are obstacles in front of it by using visual floor detection processing. If the front of robot is not floor, the robot re-generate local map and plan the path again.

4.3 Developed Behavior in Simulation Environment and Apply to Real Robot

Figure 8 shows another developed vision based behavior. First, the robot execute `search-ball` until it find the ball, then execute `track-ball`. By using a knowledge of the radius of the ball and the size of the ball in the robot's view, the distance from robot to the ball is calculated. Then execute `catch-ball` to catch the ball. In this experiment, users first program the behavior in simulation environment and then apply to the real robot without any change in the code.

5 Conclusion

In this paper, we described our humanoid robot system which consists of both real robot and simulation environment. It can simulate dynamics and collision in environment, and motors and sensors including a view of robots in real-time, so that users are able to develop vision based behaviors rapidly and efficiently, without at a risk of damaging real robot, or bothering with prepare, verify and evaluate the experiment.

Our system can share the behavior programs between real robots and virtual robots, by using Common Interface API of a robot hardware. A program developed in simulation environment is able to apply to a real robot without porting.

Finally, we present vision based behaviors experiments using our developed humanoid system. By using simulation environment for development, we are able to concentrate on the behaviors itself without bothering experimental setup or a risk of damaging the robots. We realized vision based behaviors that includes vision based local map generation, planning, navigation, walking and catching the ball, in simulation world. This complex behavior is difficult to develop only using real robot from scratch. We also show that the developed behavior software in simulation environment is able to apply to real robot through ball-catch behavior.

References

[1] J. Pratt, G. Pratt: Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation. Proceedings of the International Conference on Climbing and Walking Robots (CLAWAR99), 1999.

[2] T. Ishida and R. E. Korf: Moving Target Search, IJCAI-91, pp. 204–210, (1991).

[3] J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue: Motion Planning for Humanoid Robots under Obstacle and Dynamic Balance Constraints. Proc. IEEE Int'l Conf. on Robotics and Automation pp. 692–698 (2001).

[4] M. Inaba, F. Kanehiro, S. Kagami and H. Inoue: Vision-Equipped Apelike Robot Based on the Remote-Brained Approach, Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 2193–2198, (1995).

[5] M. Fujita and K. Kageyama: An Open Architecture for Robot Entertainment, Autonomous Agents 97, pp. 434–442, (1997).

[6] C. Breazeal and B. Scassellati: A context-dependent attention system for asocial robot, Proc. of Int. Joint Conf. on Artificial Intelligence, pp. 1146–1151, (1999).

[7] Y. Nakamura, H. Hirukawa, K. Yamane, S. Kajita, K. Yokoi, K. Tanie, M. G. Fujie, A. Takanishi, K. Fujiwara, F. Kanehiro, T. Suehiro, N. Kita, Y. Kita, S. Hirai, F. Nagashima, Y. Murase, M. Inaba and H. Inoue : V-HRP: Virtual Humanoid Robot Platform, In Proc. of The First IEEE-RAS International Conference on Humanoid Robots , (2000)

[8] F. Kanehiro, N. Miyata, S. Kajita, K. Fujiwara, H. Hirukawa, Y. Nakamura, K. Yamane, I. Kohara, Y. Kawamura, Y. Sankai: Virtual Humanoid Robot Platform to Develop Controllers of Real Humanoid Robots without Porting, In Proc. of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1093–1099, (2000)

[9] M. Inaba: Remote-brained humanoid project, Advanced Robotics, Vol. 11, No. 6, 605/620(1997)

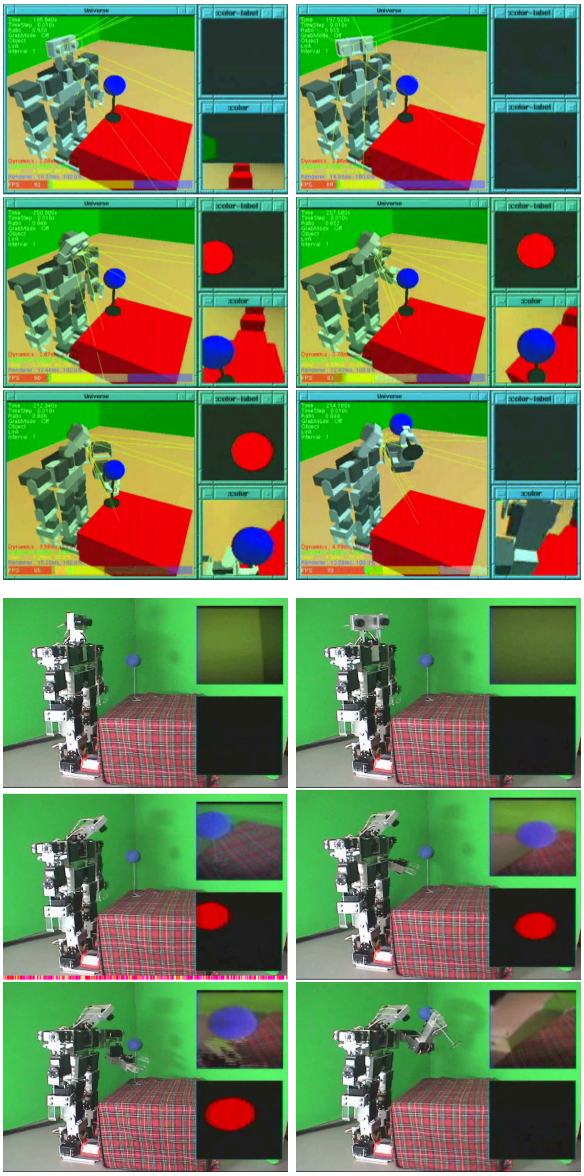


Figure 8: *Vision based behavior program in Simulation World and Real World*

[10] F. Kanehiro, M. Inaba, H. Inoue: Virtual Robot Body driven by Fast Dynamics Simulation Toolkit for Game Development, Proc. of 2001 Annual Symposium of Robotics-Mechatronics, pp 591-592 (2001).

[11] http://www.cm-labs.com, Vortex, Critical Mass Labs.

[12] F. Kanehiro, M. Inaba, H. Inoue, H. Hirukawa, and S. Hirai: Developmental Software Environment that is applicable to Small-size Humanoids and Life-size Humoanids. In Proc. of the 2001 IEEE International Conference on Robotics & Automation, pp. 4084-4089 (2001).

[13] K. Okada, S. Kagami, M. Inaba and H. Inoue: Plane Segment Finder : Algorithm, Implementation and Applications, Proceedings of International Conference on Robotics and Automation (ICRA'01) pp 2120-2125 (2001).