VisiON NEXTA: Fully autonomous humanoid robot

Nobuo Yamato[1], Yohei Akazawa[2], Hiroshi Ishiguro[3], Tomotaka Takahashi[4], Takeshi Maeda[1], Takuro Imagawa[1], Hitoshi Takayama[3], Noriaki Mitsunaga[5], and Takahiro Miyashita[5]

[1]VSTONE, 4-2-7 Shimaya Konohana-ku,
Osaka, 554-0024, JAPAN
[2]Systec Akazawa, 6-2-47 Noda Fukushima-ku,
Osaka, 553-0005, JAPAN
[3]Osaka University, 2-1 Yamadaoka Suita,
Osaka, 565-0871, JAPAN
[4]Robo Garage, Yoshidahonmachi Sakyo-ku,
Kyoto, 606-8501, JAPAN
[5]ATR Intelligent Robotics and Communications Laboratories
2-2-2 Hikaridai Keihanna Science City, Kyoto, 619-0288, JAPAN

Abstract. This paper describes the construction and functions of VisiON NEXTA, the humanoid robot that we have developed. It has 23 degrees of freedom as well as omnidirectional image, 3-axis acceleration, and 3-axis gyro sensors that detect angular velocity. Moreover, this robot has the following new functions in comparison with VisiON, which is the humanoid robot we developed last year:

1. Voice communication (speaking and listening)
2. Wireless network communication
3. USB device ports
4. Computation based on inverse kinematics and motion control by acceleration and gyro sensors.

This robot was developed as part of a platform to study such research issues as coordination among robots, teleoperation by human operators, and robotic systems integrated with sensor networks.

1. Introduction

VisiON NEXTA is a fully autonomous humanoid robot developed by TeamOsaka, which was established in 2003 to develop robot technologies in Osaka; actually, TeamOsaka won the RoboCup 2004 in Lisbon.

Companies, universities, and cities are collaborating in TeamOsaka. Members include Osaka City, VSTONE Co. Ltd., Osaka University, Systec Akazawa, Robo Garage, and ATR Intelligent Robotics and Communications Laboratories.

2. Specifications of VisiON NEXTA

Table 1 shows the general specifications of VisiON NEXTA (Fig. 1). Its significant features are as follows:

1. A fully autonomous robot based on sensory information.
2. An omnidirectional camera.

Because of their limited visual field, robots using a normal camera often miss targets, for example, a ball and a goal. Such a robot needs to move its head or rotate its body, which wastes time. Our idea solves this problem by using an omnidirectional camera instead of a normal one.

Table 1 General construction of VisiON NEXTA

Height [mm]	465
Width [mm]	260
Depth [mm]	160
Weight [kg]	3.2

Fig. 1 VisiON NEXTA

2.1 Mechanical Specifications

As mentioned above, VisiON NEXTA has a total of 23 degrees of freedom. **Table 2** shows the arrangement and types of actuators. **Table 3** shows the specifications of the actuators used for VisiON NEXTA. The actuator has a microcontroller and communicates with the host CPU through a RS485 serial network. Therefore, the host CPU can receive current angular positions of the joints, speed, voltage, and temperature of each actuator.

Table 2 Motor types and rotation axes

Parts	Rotation Axis	Servomotor
Head	Pitch	DX-113
Body	Pitch, Yaw	DX-117, DX116
Shoulders	Roll, Pitch	DX-113, DX116
Arms	Pitch, Yaw	DX113, DX113
Hips	Roll, Pitch, Yaw	DX117, DX117, DX116
Knees	Pitch	DX-117
Ankles	Roll, Pitch	DX117, DX117
	Total	23

Table 3 Specifications of servomotors

Type	DX-113	DX-116	DX-117
Size [mm × mm × mm]	46 × 31 ×37	46 × 31 ×37	46 × 31 ×37
Torque [kg cm]	10.2	21	28
Speed [rpm]	0.15	0.125	0.167
Voltage [V]	15	15	15
Weight [g]	60	66	66
Motor	Coreless	Maxon RE-max	Maxon RE-max

2.2 Electrical Specifications

VisiON NEXTA has two CPUs. The main CPU processes image data and controls all behaviors of the robot; the sub-CPU controls the motors. **Table 4** shows the specification of these CPU boards.

Table 4 Specifications of CPU boards

	Main	Sub
CPU	GEODE 400 MHz	SH2-7054 40 MHz
ROM	4 GB	384 KB+64 KB
RAM	256 MB	
Interface	USB1.1+RS232+VBA	RS232+RS485

Sensor

VisiON NEXTA has three types of sensors. First is the omnidirectional camera, which captures images in 360 degrees. Therefore, the robot does not need to move its head and turn around to find targets. Second is the 3-axis acceleration sensor, which senses the sum of gravity and external forces. So the robot can recognize whether it is standing or falling down. Third is the 3-axis gyro sensor, which senses angular velocity. With these sensors, the robot recognizes its own orientation.

Motor control

The sub-CPU communicates with servomotors through RS485. The data of servomotors are updated every 16.6 ms.

2.3 Software Specifications

Robotic software consists of several modules.

Motion control module

Robotic movement is implemented in two ways: inverse kinematics and predefined motion patterns. This is because mathematical motion patterns, for example, walking and turning, are easier to implement with inverse kinematics than predefined motion patterns, while such motions as bowing and waving a hand, are easier to implement as predefined motion patterns.

Image processing module

The image processing algorithm is as follows:

1. Capture images to transmit to the host PC.
2. Beforehand, a human operator makes a color index table to recognize the target object.
3. Match the image data with the color index data.
4. Make a histogram with the specified color index, according to the target object.
5. Compute the median point and the color of the target region by selecting the maximum peek in the histogram.

Autonomous control module

The module autonomously controls the robot by using the motion control and image processing modules; it also determines the robot’s actions and decides such behavior during competitions as walking and freestyle movements.

3. Research approaches with VisiON NEXTA

As mentioned before, we developed VisiON NEXTA as part of a research platform and are studying new autonomous humanoid robot technologies. This section introduces one research approach.

3.1 Behavior selection and environment recognition methods based on sensor history

In this subsection, we propose a method that recognizes environment and selects appropriate behavior for humanoid robots.

3.1.1 Outline of proposed method

Humanoid robots have difficulty moving in such daily environments as a family's house because the viscous friction or elasticity of each floor, which directly influences the robot's motion and are difficult to immediately measure, are different from each place. Therefore, we propose a method to recognize the features of environments and select appropriate behavior based on the histories of simple sensor outputs to achieve a humanoid robot able to move around a house.

Our method's key idea uses long sensor history to find the features of the environment. To measure such features, almost all previous research proposed methods that used several kinds of sensors with large amount of calculations to quickly process outputs. However, such approaches are unreasonable because the robot lacks sufficient space on its body for the attached sensors and processors. Hence we propose using sensor history to measure them because there are close relationships between sensor histories, motions, and environments. When the robot performs specific motions in specific environments, we can see those features in the sensor history that describes the motion and the environment. Furthermore, such features as viscous friction or floor elasticity do not change quickly. Thus we can use the sensor history for a long time to measure them.

The outline of the method is as follows:

A-1 [preparation 1] In advance, the robot's user makes basic motions appropriate to the environment.

A-2 [preparation 2] For each basic motion and environment, the robot records the features of the time series data of its sensors when it follows the motions.

A-3 [preparation 3] For each basic motion, the robot builds decision trees to recognize the environments based on recorded data by using C4.5 [1]. It calculates recognition rates of decision trees by using cross-validation of the recorded data.

B-1 [recognition 1] The robot selects the motion that corresponds to the decision tree that has the highest recognition rate. It moves along the selected motion and records the features of the time series data of the sensors.

B-2 [recognition 2] The robot calculates the recognition reliability of each environment based on the decision tree and the recorded data. Then it selects the environments that have reliability greater than a threshold as candidates of the current environment. The threshold is decided by preliminary experiments.

B-3 [recognition 3] The robot again builds decision trees based on the data recorded at the process (A-2) that correspond to the selected candidates of the current environment. Go to (B-1).

By iterating these steps, the robot recognizes the current environment and selects appropriate motions.

3.1.2 Robot's motions and features of the environment

Figure 2 shows the motions that the robot has in advance. In our method, there are two kinds of motion, basic and those depending on environments. The basic motions are comprised of a set of motions that can be done in each environment without changing the loci of joints, such as waking up, lying down, etc. All robotic motions appropriate to each environment are generated in advance by the user. By utilizing our method, once the environment is recognized, the robot can select the suitable motions for it.

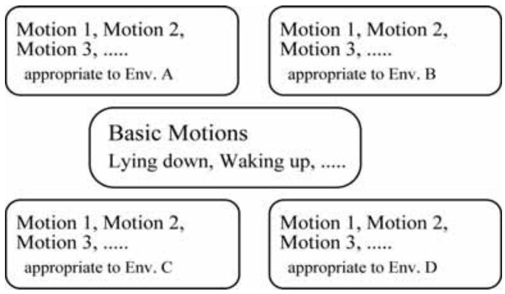


Fig. 2 Robots have two kinds of motion, basic and those that depend on environment. They are generated by users in advance.

In this paper, we use averages and standard deviations of the time series data of the sensors and averages and standard deviations of velocities and accelerations of changes in the sensor outputs as the features of the environment. For example, the relationships between motion, environments, and sensor histories are shown in **Table 5**.

Table 5 Relationships between basic motions, environments, and features of sensor history. $s_n(t)$ denotes time series data of sensor s_n

basic motions	label of environment	Ave. of $s_n(t)$	Std. dev. of $s_n(t)$	Ave. of $ds_n(t)/dt$	Std. dev. of $ds_n(t)/dt$	Ave. of $d^2s_n(t)/dt^2$	Std. dev. of $d^2s_n(t)/dt^2$
Lying down	Tiled floor	136.19	21.429	131.13	6.1985	157.83	11.292
Waking up	Tiled floor	149.15	25.64	128.84	6.2903	132.89	13.554
….	….	….	….	….	….	....	....

3.1.3 Decision tree based on relationships between basic motions, sensor histories, and environments

A decision tree to recognize the environment is made by C4.5 [1], which is a program for inducing classification rules in the form of decision trees from a set of given examples. We use the relationships described in **Table 5** as examples and make decision trees for each basic motion by using knowledge analysis software WEKA [2] that can deal with C4.5. **Figure 3** shows an example of a decision tree for the lying down motion.

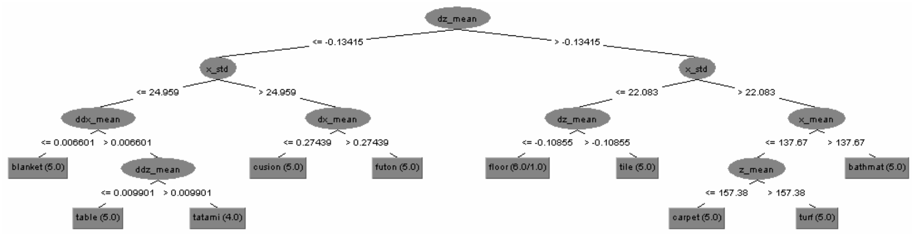


Fig. 3 Decision trees recognize environments based on relationships between lying down motion, environments, and sensor histories. Circles denote features of sensor history. Rectangles denote environments

We can also determine the recognition rate of a decision tree of each basic motion and the reliabilities of the recognition results by cross-validation by calculating the number of correctly classified instances over all instances. Reliability is the same as the recognition and misrecognition rates of each environment by specific decision trees. For example, when the recognition results are the *wooden table* environment, the reliability of the wooden table environment is the same as the recognition rate, and the reliability of other environments is the same as the misrecognition rate.

3.1.4 Experiments for verification of proposed method

To verify the validity of the proposed method, we conducted a preliminary experiment with our small humanoid robot, Robovie-M. As shown in **Figure 4**, we attached two 2-axial acceleration sensors to the robot to acquire acceleration values along three orthogonal axes as features of the environment. **Table 6** and **Figure 5** show environments for recognition and the basic motions in the experiments. We recorded the time series data of the sensor outputs five times at each environment and for each motion.

For instance, we introduced the recognition process of the *futon* (a Japanese mattress) environment. First, the robot selected the *stepping on both legs* motion because the motion's decision tree has the highest recognition rate. All recognition rates are described in **Figure 6**. Second, the robot obtained the sensor history while doing the *stepping on both legs* motion and classified it by using the motion's decision tree. The result of classification was the *blanket* environment. The reliabilities of each environment were obtained, as shown in **Table 7**. This time the reliability threshold was 0.2. Then the selected candidates of the current environment were *tatami*, *futon*, *artificial turf*, and *blanket*. Next, the robot made decision trees for each basic motion based on the data of the candidates. By calculating their recognition rates, as shown in **Figure 7**, the robot selected the *stepping on one leg* motion. As a result of performing the selected motion, the robot classified the data to the *futon* environment and obtained *artificial turf* and *futon* as candidates, as shown in **Table 8**. The robot selected the *lying down* motion from the recognition rates based on the candidate's data shown in **Figure 8**. Finally, the robot obtained the data while lying down and recognized the current environment as the *futon* environment shown in **Table 9**. We verified that the robot recognized all environments by using our method, as shown in Table 6.

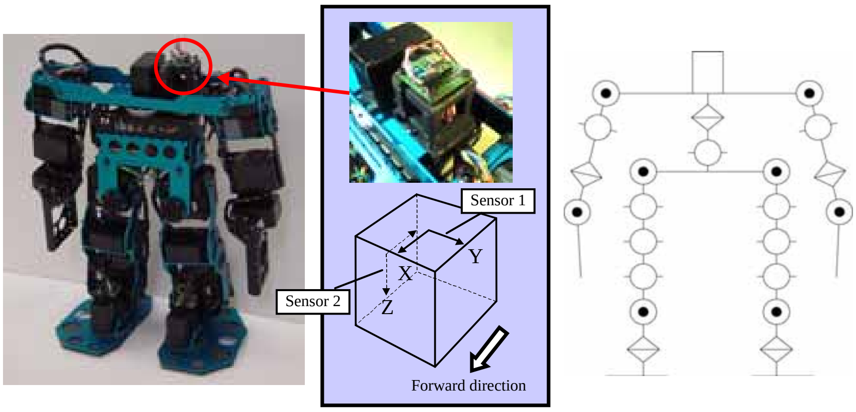


Fig. 4 Image on the left shows humanoid robot Robovie-M, and center images indicate sensor arrangement. On the robot's left shoulder, two 2-axial acceleration sensors are attached orthogonally to acquire acceleration values along three axes that describe horizontal and vertical motions. Image on the right shows an arrangement of robot's degrees of freedom.

Table 6 Left column describes environments used in the experiment. Right column describes the basic motions. Environments are selected from a typical Japanese house.

Environments		Basic motions
Ceramic tiled floor	Linoleum floor	Lying down
Wooden table	Tatami mat	Waking up
Cushion	Futon	Tossing and turning
Carpet	Bathmat	Stepping on one leg
Blanket	Artificial turf	Stepping on both legs

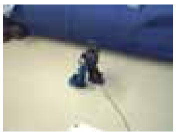
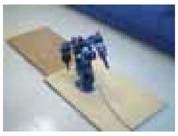
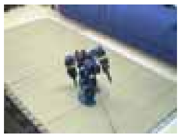
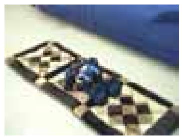
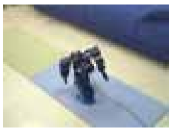
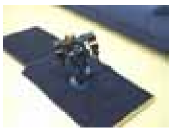
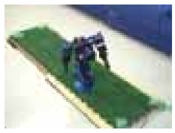
Ceramic tiled floor	Linoleum floor	Wooden table	Tatami mat	Cushion
				
Futon	Carpet	Bathmat	Blanket	Artificial turf
				

Fig. 5 Pictures of environments in experiments.

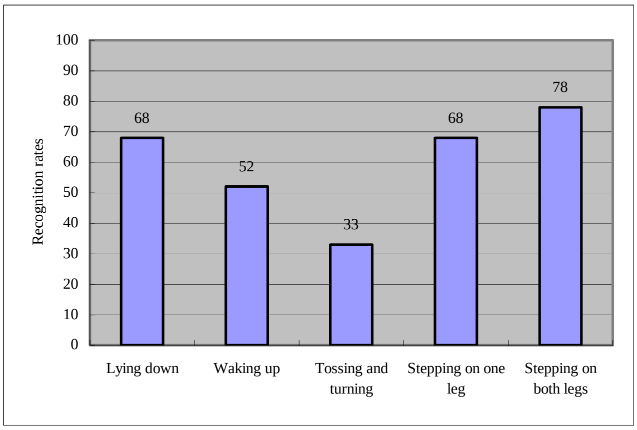


Fig. 6 Recognition rates of decision trees of each motion based on all data. Highest rate is the *stepping on both legs* motion.

Table 7 Reliabilities of each environment when decision tree of *stepping on both legs* motion classifies data to *blanket* environment

Environment	Ceramic tiled floor	Linoleum floor	Wooden table	Tatami mat	Cushion	Futon	Carpet	Bathmat	Blanket	Artificial turn
Reliability	0.0	0.0	0.0	0.2	0.0	0.2	0.0	0.0	0.4	0.2

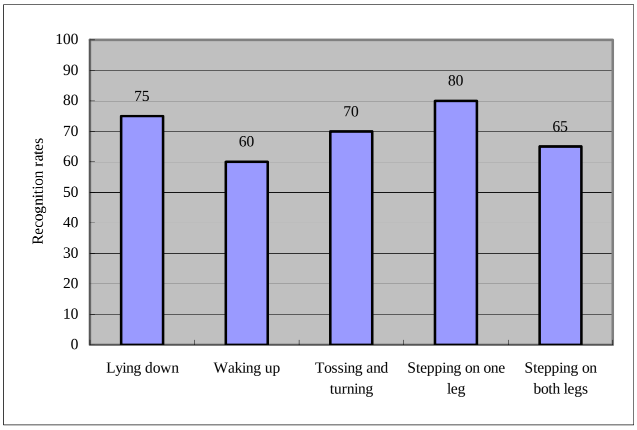


Fig. 7 Recognition rates of decision trees of each motion based on data that correspond to *tatami*, *futon*, *artificial turf*, and *blanket*

Table 8 Reliabilities of each environment when decision tree of *stepping on one leg* motion classifies data to *futon* environment

Environment	Ceramic tiled floor	Linoleum floor	Wooden table	Tatami mat	Cushion	Futon	Carpet	Bathmat	Blanket	Artificial turn
Reliability	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.2

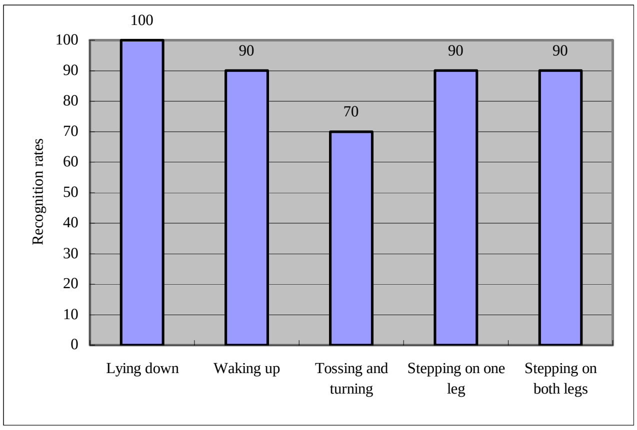


Fig. 8 Recognition rates of decision trees of each motion based on data that correspond to *futon* and *artificial turf.*

Table 9 Reliabilities of each environment when decision tree of *lying down* motion classifies data to the *futon* environment.

Environment	Ceramic tiled floor	Linoleum floor	Wooden table	Tatami mat	Cushion	Futon	Carpet	Bathmat	Blanket	Artificial turn
Reliability	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

Conclusion

In this paper, we introduced our humanoid robot named VisiON NEXTA that has two CPUs, 23 degrees of freedom, and several kinds of sensors to autonomously generate various motions not only for the RoboCup but also for a platform of several research issues. We also proposed a method to recognize environment and select appropriate behavior for humanoid robots based on sensor histories. The results of experiments indicated that the robot recognized ten different environments and selected appropriate behaviors.

References

1. Quinlan J. R., C4.5: Programs for Machine Learning, Morgan Kaufman, 1993.
2. I. Witten and E. Frank, "Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations -," Morgan Kaufman, 2000.