The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

Autonomous Walking with use of Quadruped Virtual Robot

Kouki MIYOSHI, Norihiro ABE, Yoshihiro TABUCHI
Kyusyu Institute of Technology
680-4 Kawazu, iizuka, Fukuoka 820-8502, Japan
Email:miyosi@sein.mse.kyutech.ac.jp

Hirokazu TAKI
Wakayama University
930 Sakaedani, Wakayama-shi
Wakayama 680-8510, Japan
Email: taki@sys.wakayama-u.ac.jp

Shoujie He
VuCOMP
Richardson, Texas, USA
Email: hesj@computer.org

Abstract: In the development of a robot, it needs much cost and time to verify a robot's motion with use of real machinery. Especially it is difficult to validate a robot's behavior at the unsafe place. Then developers have paid attention to virtual debugging system. It makes verification of a machine's behavior makes more efficient and easy by using a program validated in VR space.

In this research, we have a virtual robot walk on a road autonomously with the images which are captured by cameras on the virtual robot.
Keywords: *VR, Virtual Robots, Virtual Space*

1. INTRODUCTION

In recent years, many robots have been developed. It is easily to expect that a robot becomes a member of the general public in the near future. But the development of robots takes much cost and time. One of the reasons is to validate with real machinery in real space. If a real robot should be damaged, it will force us to pay much time and expense for fixing the robot. This will increase necessary expense and the length of a period for robot development, so no one deploys any robots in dangerous environment for fear of damage owing to violent fall or collision.

Then a virtual debugging system has gotten attention [1][2][3]. Developers are allowed to design the same machinery as the real one in VR space, and to verify the robot motion by using it. It saves speed of developments by verifying whether the virtual robot works well or not in the virtual debugging system. Additionally, developers enable to create various testing environments and test the robot's behavior on these environments.

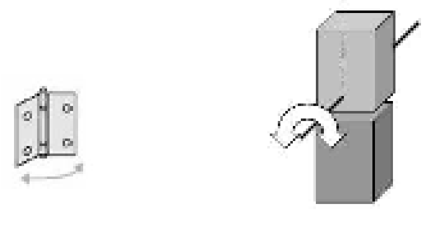
In this research, we make a virtual four legged robot built in virtual space based on physics, and have it move autonomously. By analyzing images captured with two cameras installed on the four legged robot, we have it track a line and walk on a road by using the algorithm that tracks a line.

2. Construction of Virtual Reality Space

In the simulation, it is necessary to construct the virtual reality space based on a physical rule because of making virtual environment close to real one as far as possible [4]. So we use the rigid physics calculation library, Vortex (developed by CMLabs Simulations, Inc [5][6].) to build the virtual reality space. The Vortex provides the function that creates fundamental objects like plane, box, corn, sphere, and cylinder. We enable to use a constant restraint between objects as a joint. We are able to combine different basic objects to create a composite object. Combination of composite objects and joints enables to express a complex object like robot and car.

3. Expression of Virtual Robot

In this research, servomotors of virtual quadruped robot are expressed with a hinge joint between solids two boxes and two rectangular as shown in Fig.1 [7].



(a) Hinge joint (b) Servomotor

Fig.1. A Virtual Servomotor

A virtual quadruped robot consists of 13 composite objects and 12 hinges. Each part of body, shoulders, upper legs, and lower legs is composed of boxes and cylinders. These parts are linked with a hinge joint. Fig.2 shows an appearance of robot.

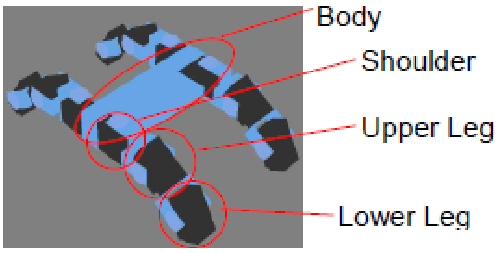


Fig.2. Quadruped Robot

4. Autonomous Walk

4.1. Image Buffer

In this research, images of the environment that robot sees are used to control the robot, but the position of the image buffer is not provided with Vortex but with Open GL. On the other hand, both Open GL and Vortex manage any rigid object with the transformational matrix, then the robot can acquire its transformational matrix from the result of image processing and infer it location based on the images observed with two virtual cameras.

4.2. Virtual Gradient Sensor

Images captured with cameras will decline (shown as Fig.3) because the four-legged robot declines when it goes ahead. So in addition to cameras, a virtual gradient sensor is installed on the robot. The images are rotated by the angle detected with a gradient sensor to make images processing easy.

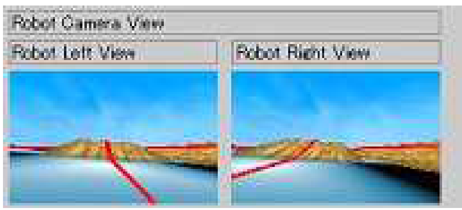


Fig.3. Images from two virtual cameras

A virtual gradient sensor calculates the gradient angle of camera. The gradient θ is represented as the expression (1) by using two coordinates (x_1, y_1) and (x_2, y_2) shown in fig.4.

$$\theta = \sin^{-1}\left(\frac{x_1 - x_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}\right) \qquad (1)$$

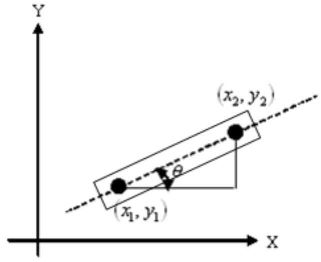


Fig.4. The Virtual Gradient Sensor

4.3. Autonomous walk along a line

Before correcting an image captured with a virtual camera, the image is binaried and linearly expanded. The binalization makes tracking a line easier and the linear expansion prevents a line from being segmented when the image is rotated.

There may be several lines except for the line the robot should trace in the captured image as shown in fig.5, when the robot walk along the line. After rotating images, system searches a line from a point close to the robot in the image to find the line the robot should follow.

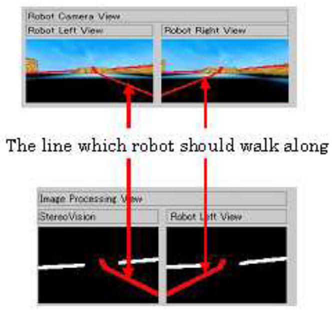


Fig.5. Extracting Line the Robot should Follow

To make a walking command, the left image is used as a base image. The first step is to find a red pixel that is a part of red line from the left image. The second step is to find the rod pixel that has same y-coordinate value as one of the red point found in left image from the right image as shown in Fig.6. This process is run over until a red point is found in the both images. The average number of x-coordinate of the right and the left images decides that the robot is on line or deviates from side to side.

If the robot is on the line, the average x-coordinate is equal to the half size of the image width. If the robot deviates to the right side, the average x-coordinate will be less than the half size of the image width. On the other hand, in case the robot deviates to the left side, the x-coordinate will be more than the half size of the image width (shown as Fig.6). Even if a robot is going straight, as it will shake from side to side, the x-coordinate dose not precisely coincide to the half size of the image width. So the decision of whether the robot is on the line or not is relaxed. If a robot is judged to be on the left side, it will be given a command to move to the right direction, and vice versa.

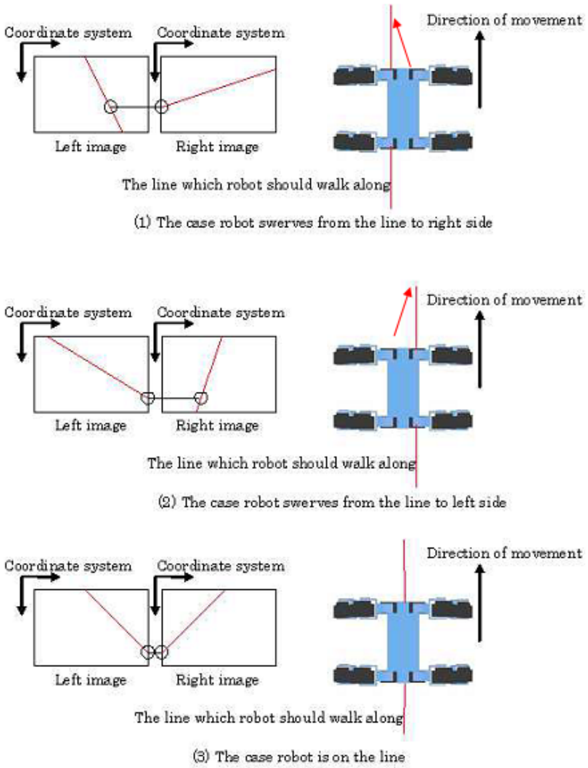


Fig.6. The Decision of a Walk Order

The robot must turnaround only when a draft exceeds a threshold value, because four legged robot always slips during a walk to some extent from side to side. If the line is found in just one of images, say right one, it must turn to the left because it is on the right of the line to track.

4.4. Autonomous walk on a road

When the robot walks on a road, system calculates an imaginary center line passing through the center between two white lines (shown as Fig.7) white lines which are the edges of the road. The robot follows the line with the algorithm shown above.

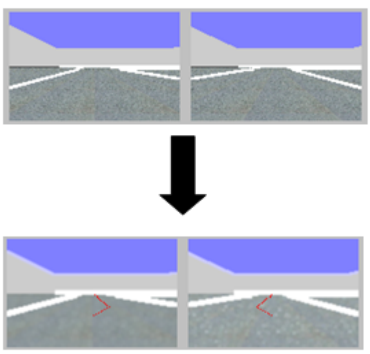
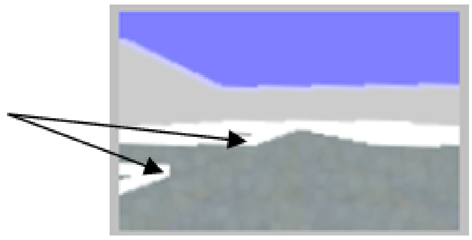
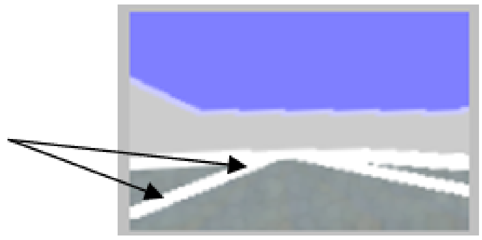


Fig.7.Plot the Imaginary Center Line

Finding a branch is necessary to correspond to a road that has branches. A white line on a road on which there is a branch is disconnected, for example, as shown in Fig.8 (a), in which there is a corner toward left, at the two points marked with arrows a white line is disconnected. If there is no branch at all, a white line remains as it is as shown in Fig 8 (b).



(a)→A road with divergence



(b)→A road without divergence

Fig.8. Branch Point on White Line

Detection of a branch point is done based on the fact me ntioned above. Which direction a branching road extend s to is determined by finding portions where a white line is disconnected. As the robot is always swaying toward right or left while walking, disconnection on a white li ne is not always observable from the robot, that is, even if it has been observed at the previous frame, it may not be observable at the current frame but it may be observe d again at the next frame. This will cause the robot to mi sunderstand there are two disconnection on the same wh ite line.

It is necessary to give a map to a robot in advance because it must determine the route leading it to the goal from the start. The map exploited by a robot is a topological one without distance information. This map allows a robot to infer its current location with respect to branching points. Assuming that distance between two successive branches on a white line is enough long, it is considered that such intermittent observation of disconnection means that the robot staying in a branch observes the same branch repeatedly. Consequently, if a robot encounters the sequence as shown in Fig.9, it can recognize that there is one branch in the interval B and D respectively, where 1 corresponds to the discovery of one disconnection.

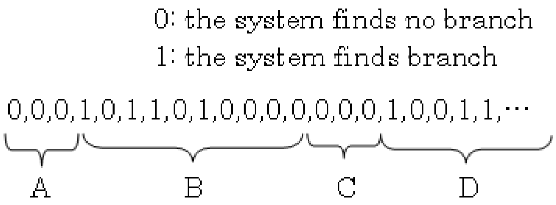
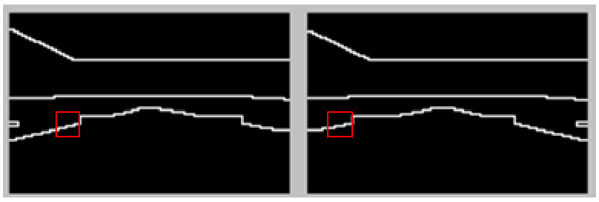


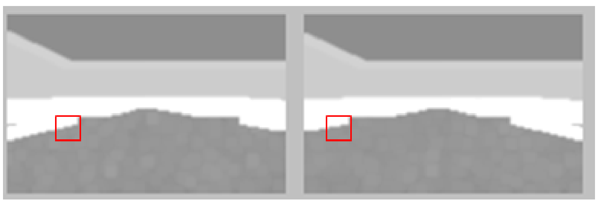
Fig.9 Determine Whether the Robot is at Branch Point

As s robot is subject to non holonomic constraints, it does neither move sideways nor turn without translation. This means that a robot must be given a reparatory of trajectories in the same way as a car-like non holonomic robot. At present, a robot is not given such a reparatory but it must generate a trajectory to turn a corner being close at hand using a distance measurement with a stereo vision. There are, however, the cases it goes over a white line at a corner if it curves to near right angle. To avoid this problem, a bird's eye view of the region observed by the robot using a stereo vision is necessary to guide the robot behavior.

First, correspondence between a right and left image is calculated for each edge of white lines based on an edge based stereo vision. Next, correlation based on the correspondence between pixels successfully corresponded is calculated, then based on the result three dimensional coordinates of points are obtained. A robot judges where to start rotation to turn a corner being close at hand.



(a) Edge Images



(b) Gray Scale Images

Fig.10 Corresponding Point of Stereo Vision

5. Conclusions and Future Work

This research's aim is to make a virtual robot autonomously walk in the Virtual space and we successfully implement simulation. Now the robot is able to walk autonomously along without a branch. In the case that the road has diverging, system finds the branch point and detects how many branch points the robot passed.

Now, except for up hills or down hills, the system successfully simulates the behavior of a robot on a flat plane with a constant homogeneous friction coefficient. Next we would like to simulate the behavior on rolled ground with variable friction coefficients. To have a robot walk autonomously along a road with several forks, the robot must locate itself refereeing to the map information calculated using given start and goal points to the robot. The robot must recalculate the route, if the robot judges it impossible to go ahead in motion for the reason that the road is occupied with obstacles or the road is too narrow for the robot to go through.

6. Acknowledgment

We greatly appreciate the aid of the Grant-in-Aid for Scientific Research (S) and (A).

7. Reference

[1] Yi PAN, Norihiro Abe, AND Kazuaki Tanaka (2003): The Virtual Debugging System for Embedded Software Development, *SCI2003*, 78-84.

[2] Yi Pan, Norihiro ABE, Kazuaki Tanaka (2003): The Virtual Debugging System for Embedded Software Development, *VRAI2003*.

[3] Yi Pan, Norihiro Abe, Kazuaki Tanaka, Hirokazu Taki (2004): The Virtual Debugging System for Developing Embedded Software Using Virtual Machinery. *Embedded and Ubiquitous Computing, International Conference (EUC 2004),* Aizu, Japan, 85-95.

[4] Norihiro Abe, Yoshinori Takamura, Kazuaki Tanaka, Hirokazu Taki (2003): Construction Of Physics Simulation Environment for Education Using Vortex, In *proceedings of the SCI2003*, 301- 307.

[5] VORTEX Developer Guide, CMLabs Simulation, Inc, 2002

[6] VORTEX Viewer Developer Guide, CMLabs Simulation, Inc, 2001

[7] Toshiaki Oomori, Norihiro Abe, Kazuaki Tanaka, Hirokazu TAKI, Shoujie HE (2005): Concurrent development of Virtual Robots and Real Robots Based on Physical Low, *IMEKO / IFAC / IFIP Workshop on Advanced Robot Systems and Virtual Reality. ISMCR2005*.