Mechatronics xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Mechanism, control, and visual management of a jumping robot

Dong Hwan Kim *, Jae Hyun Lee, Inho Kim, Seung Hee Noh, Se Kyu Oho

School of Mechanical Design and Automation Engineering, Seoul National University of Technology, 172 Gongreung-dong, Nowon-gu, Seoul 139-743, Republic of Korea

ARTICLE INFO

Article history:
Received 4 June 2007
Accepted 22 May 2008
Available online xxxx

Keywords:
Jumping robot
Pneumatic lift-up
Automatic balance mechanism
Three-dimensional motion representation

ABSTRACT

A new robot, known as the Scoutrobot, is introduced in this paper. It is capable of moving along the ground and jumping over a surmountable obstacle. The Scoutrobot can patrol a harsh environment by rolling or jumping while delivering environmental information to the user. It requires less time and makes use of simpler mechanisms to overcome obstacles compared to conventional robots. The Scoutrobot contains several sensors, including two cameras for distance measurements and ultrasonic sensors for alignment. The jumping motion is performed by releasing a pneumatic cylinder against the ground. The jumping height is controlled with a pulse-width modulation-control mechanism on the pneumatic valve. An automatic lift-up structure is used to recover the robot's initial pose after it jumps up or down. A robot management system provides a three-dimensional virtual display image of the robot's location and pose.

1. Introduction

Robots are evolving continuously as technologies advance and users demand more sophisticated tasks. It is commonly understood that a robot must have a specific mechanism to comply with a required function. For example, wheels or caterpillar tracks are often used so that the robot can climb over or avoid an obstacle of considerable height. As a floor navigating robot with wheel mechanism a Scoutrobot is developed [1,2]. The main role of the robot is on patrolling and monitoring an environment with a fast movement and delivering the information of the environment. Scout robots are also applied to the tasks for rescue and search [3,4]. In this paper, we propose a new robot mechanism to overcome an obstacle based on human behavior: jumping over the obstacle instead of climbing over or crawling under it. Although humans can easily jump over some objects, they cannot roll along the ground, limiting their traveling speed. Thus a robot with fundamental jumping and rolling movements is particularly desirable.

We designed a jumping robot and used it to patrol a rugged environment while avoiding obstacles by rolling and jumping. The robot was required to provide information about the environment in front of it as it encountered and maneuvered around obstacles. This information was collected using sensors and cameras, and delivered to a remotely placed user. Our tests demonstrated that the robot was sufficiently strong and flexible to withstand the impact caused by jumping onto an object or down to the ground. One simple way of reducing the damage from impacts is to use a lightweight body design. This reduces the size of the robot so it can fit in a small space and maneuver through narrow channels. However, a robot that is too lightweight cannot be fully equipped with the necessary components, such as cameras, sensors, and battery. A weight limited to a range that can be easily lifted by a human with a compact packing of the robotic components appears to be a suitable compromise.

A jumping robot can be used for military purposes and security surveillance, or as a rescue device for an accident scene. Its powerful functions, such as camera vision, wireless communication, and sensor detection, allows the system to be used as a substitute for dangerous human military operations, for monitoring specific areas, and for rescuing injured people from rugged terrain or a critical situation. Our proposed jumping robot, known as the Scoutrobot, is based on an exceptional mechanism, and provides obstacle avoidance using stereo vision and a jumping control algorithm. A three-dimensional display represents live movements of the robot using a perspective transformation, which allows the user to view the scene around the robot more effectively than a simple two-dimensional display.

2. System configuration

Conventional robots that utilize a wheel or caterpillar track mechanism to crawl under, climb over, or go around small obstacles require a complicated design and can be very costly. The proposed jumping mechanism allows the robot to overcome an obstacle as long as it can be surmounted by jumping, eliminating the time wasted by a detour. Several mechanisms have been developed previously to permit a robot to jump, such as a winch wire release [5], spring compression, and pneumatic extrusion [6,7].

* Corresponding author. Tel.: +82 2 970 6362; fax: +82 2 974 8270.
E-mail address: dhkim@snut.ac.kr (D.H. Kim).

0957-4158/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mechatronics.2008.05.004

The winch wire mechanism has a plate spring used to store a compression force, which is obtained by winding the wire through a pulley. However, it requires a winding or releasing mechanism and a long plate spring. Direct compression requires a stiff spring to ensure that a sufficient force is generated over a relatively small compressed length, requiring a heavy body to support the large torque motor and an additional mechanical clutch. We adopted a pneumatic cylinder that moves fast enough to bounce back from the ground when it is released by compressed air. This requires an air tank, which we placed on top of the robot body. The robot has weight of 3 kg and the size is 250 mm in wheel diameter, 200 mm in length, and 250 mm in height. The strength of the machine is designed to be durable at impact for a free fall from 1 m height. The maximum velocity of the robot is 25 cm/s and the maximum piston rod velocity is 250 cm/s, which determines the jump height up to 30 cm. With considering the desired jumping height of 30 cm where the robot weight is a crucial factor, and installment space for sensors and actuators, the machine size is arrived at $250 \times 200 \times 250$ mm size.

Contrary to the works [6,7], the jumping height is controlled by combining the body traveling speed with jumping speed considering the jumping height, which is estimated by vision and sensor. The jumping speed is controlled by imposing PWM signal with different duty ratio on the solenoid valve. The structure of the Scoutrobot is illustrated in Fig. 1.

The robot was designed to jump without inflicting serious damage to its body and to recover its initial pose after the jump. Thus, it must be flexible and contain a self-returning mechanism. This was accomplished by installing suspensions on the wheels and a bar along the back of the robot. During a jumping maneuver, the bar is released from the robot's back, and when the robot hits the ground, the bar is deflected by a spring and returns to its original position.

The distance to and size of an object ahead of the robot is determined using two cameras and a stereo vision algorithm. The jumping maneuver is affected by the obstacle height, which is used to determine the point at which the robot will start to jump, the vertical jumping velocity required from the cylinder extrusion, and the forward traveling velocity of the robot. We do not want the robot to jump too high, which could cause unnecessary damage to the robot, or to jump too low.

A Bluetooth module is used to transmit and receive commands between the robot microcontroller and the remote computer using wireless communication. The camera images are delivered to the computer via a wireless radio frequency (RF) module. The system structure for the Scoutrobot is illustrated in Fig. 2, including the procedures for jumping over an obstacle and the means of delivering commands and image data between the robot microcontroller and the remote computer. The robot views the area in front of it while it is moving, and the CCD camera images are continuously transmitted to the PC through the wireless RF module. The image data are sent to an image grabber, where image processing is performed to determine the shape and size of any obstacles. If an obstacle is found ②, the robot must decide whether the robot to detour around it or jump over it based on its geometrical measurements, which are obtained by integrating the camera images and sensor information. When a decision about the object is made, the final command is delivered to the robot controller digital signal processor. If the obstacle is small enough to be overcome by jumping, the robot aligns itself so that its direction of motion is normal to the obstacle baseline. After estimating the height of the obstacle, the jumping velocity, which determines the jumping height, is calculated and a duty ratio for the valve that controls the pneumatic cylinder is determined. Eventually, the robot jumps up and lands on top of the obstacle, or jumps over the obstacle if it is sufficiently low ③. If the object is too high for robot to jump up on it, the robot detours around the obstacle and keeps moving afterward ④.

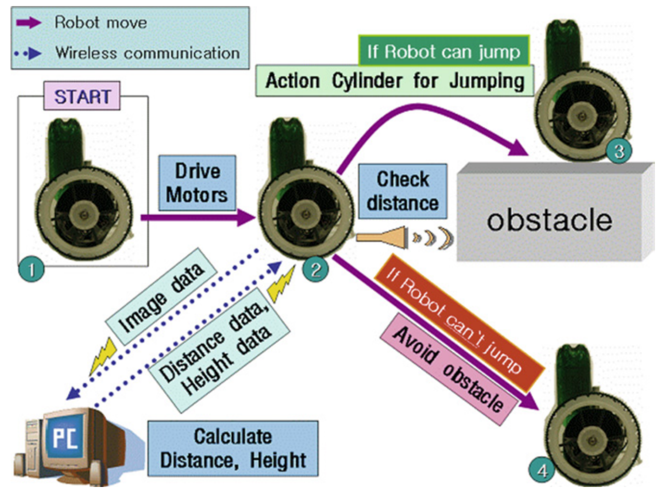

Fig. 2. System configuration of the Scoutrobot.

The movement of the robot in its environment is represented on a computer screen. A robot management system (RMS) is introduced to provide a framework for the viewable representation. This renders a view of any object using different orientations and scans the area around the robot, allowing the user to visualize the robot's environment and to navigate the robot in three-dimensional space. The three-dimensional display provides a good view of the robot's environment and can be used to determine a landing position on the ground, which is not possible with a two-dimensional display.

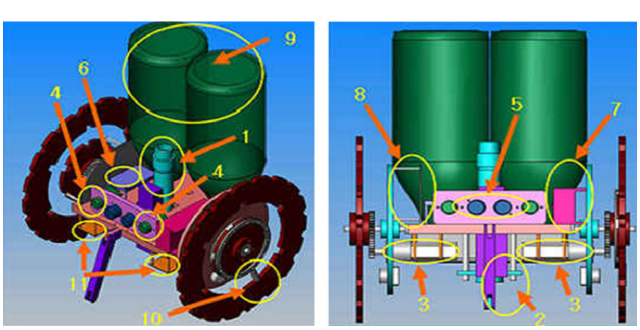

Fig. 1. Structure and components of the Scoutrobot: 1. cylinder, 2. solenoid valve, 3. motor encoder, 4. camera, 5. ultrasonic sensor, 6. blue tooth module, 7. digital signal processor controller, 8. electric power circuit, 9. air tank, 10. damper, and 11. motor driver.

3. Jumping mechanism design and verification

3.1. Mechanism design

The Scoutrobot has length of 250 mm, a wheel diameter of 200 mm, and a height of 250 mm, including its air tanks. A cylinder is used to exert a jumping force on the ground, which drives the robot up. This force is determined by a pulse-width modulation-controlled solenoid valve. The air is stored in two air tanks, which must be recharged when they run out.

After the robot jumps to a high position, it must land safely back onto the ground. To prevent any damage from occurring, a balance bar is installed on the back of the robot. A released spring force is used to move the bar back to its original position after the robot hits the ground. The wheels are designed to reduce the impact of

Please cite this article in press as: Kim DH et al., Mechanism, control, and visual management of a jumping robot, Mechatronics (2008), doi:10.1016/j.mechatronics.2008.05.004

the ground when the robot jumps. This results in a set of springs resembling umbrella ribs, as shown in Fig. 3.

The weight and size of the Scoutrobot are its most critical design factors. Lighter robots will jump higher and land safer. However, smaller weights require smaller dimensions, which limit the number of sensors, actuators, and wireless modules. A compact packing of all components is desirable to minimize the size and weight, and these components should be as small as possible.

When the height of the obstacles is determined by vision processing, the robot waits for a command from computer whether the robot jumps or goes around. The computer calculates the height of obstacle and the distance to obstacle and sends a message to the robot. The command message code for jumping or turning around is set as Table 1. The notations of "J1" means jumping with speed mode 1, and "T5" represents detouring the obstacles. The other values after "T" denotes speed modes where larger value is assigned with faster speed. The jumping speed is controlled by pneumatic valve by assigning different duty ratio.

The jumping height depends on the traveling and jumping speeds of the body. The jumping angle, a major factor in determining the jumping height, is determined from a vector sum of the two speeds (see Fig. 5). The jumping angle becomes small when the robot travels fast and the jumping speed is relatively slow. The jumping speed of the Scoutrobot is determined by the cylinder speed which is controlled by a pulse-width modulation (PWM) signal variation supplied to the pneumatic valve. For example, the cylinder speed decreases when a low duty ratio is applied to the valve, which means that the valve blocks the line connected to the cylinder more often than it allows the line to remain open. Experimental jumping heights for various duty ratios are given in a later section. In Fig. 4, the model of the cylinder speed control by PWM is illustrated.

When the robot takes a command of jumping over obstacles or jumping onto the obstacles from the result of the camera image processing, the robot is determined an appropriate traveling velocity and piston rod projection speed. When the two speeds are set on robot movement, the projectile trajectory is determined. If the height of obstacles is small enough to jump over, the traveling speed does not have to be set fast. Only piston rod speed is controlled enough to jump over. For large height jumping case of near 30 cm, the traveling speed need to be set fast enough to increase the maximum jumping height because the starting speed of jumping is a vector combination of traveling speed and piston rod speed.

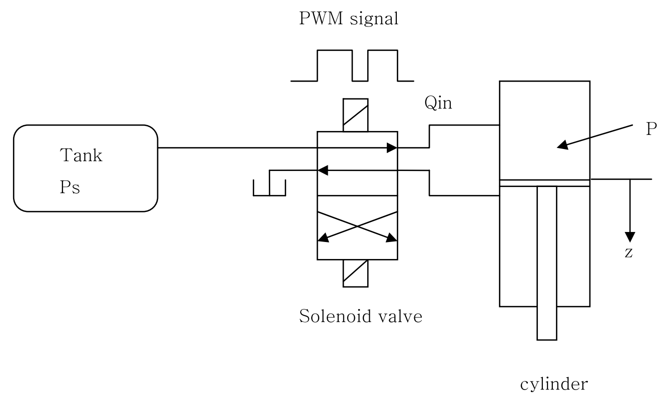

Fig. 4. Model of cylinder speed control by PWM signal.

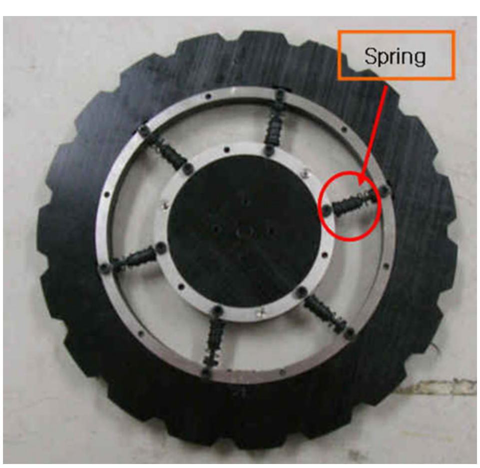

Fig. 3. Wheel design with a suspension mechanism.

Table 1
Command message code for jumping or turning around motions

Height	10–15 cm	15–20 cm	20–25 cm	25–30 cm	Over 30 cm
Command	J1	J2	J3	J4	T5

The jumping height is calculated as follows:

$$x(t) = v_{0x}t$$
$$y(t) = v_{0y}t - \frac{1}{2}gt^2, \tag{1}$$

where $x(t)$ and $y(t)$ are the horizontal and vertical positions of the robot, respectively; v_{0x} and v_{0y} are the initial traveling speed and vertical jumping speed before jumping, respectively; g is the gravitational acceleration; and t is the elapsed time after jumping. The initial jumping velocity written in vector form is

$$v_0 = v_{0x}\mathbf{i} + v_{oy}\mathbf{j} \tag{2}$$

The user varies v_{0x} by adjusting the robot's speed, while v_{0y} is selected to obtain an appropriate jumping height, which is a function of the force exerted by the cylinder that is controlled by the valve duty ratio. The jumping angle is determined from the horizontal and vertical speeds,

$$\lambda = \tan^{-1}\left(\frac{v_{0y}}{v_{0x}}\right). \tag{3}$$

If we do not take into account frictional losses and the fact that the cylinder pressure reaches approximately the supply pressure [7], the equation of the cylinder rod is given by

$$m_r\ddot{z} = A(P - P_0) - m_r g = (P_s - P_0)(A - A_0) \tag{4}$$

where m_r the cylinder rod mass, P and P_0 are the cylinder chamber inlet and outlet pressures, respectively. A and A_0 are the pressure-giving and pressure-receiving areas. We can assume that the cylinder rod moves at uniform acceleration until it reaches the cylinder stroke L. Thus, the cylinder rod velocity v_r is given as

$$2\ddot{z}L = v_r^2 \tag{5}$$

The cylinder rod velocity also varies by the change of duty ratio on the valve. In other words, at low duty ratio, the rod velocity is very slow. There is an almost linear dependency between the duty ratio and rod velocity. By (4), (5), and the duty ratio the rod velocity is given by

$$v_r = \frac{2(P_s - P_0)(A - A_0)}{m_r}L\gamma \tag{6}$$

where γ is the valve duty ratio. The required vertical jumping speed is determined by combining the two constraints. Once this is known, the corresponding duty ratio needed for the solenoid valve to cause the pneumatic cylinder rod to exert the desired force on the ground can be determined from the pneumatic pressure, mass of the cylinder rod, and mass of the robot body. By utilizing the impact principle between two bodies, the vertical jumping speed v_{0y} can be expressed by

$$v_{0y} = \frac{m_r v_r}{(M + m_r)} \tag{7}$$

where M is the robot body mass, and γ is the valve duty ratio. The maximum jumping height is determined by projectile motion

$$y_{\max} = \frac{v_{0y}^2 \sin^2(\lambda)}{2g} \tag{8}$$

The jumping angle is determined by the horizontal traveling speed v_{0x} and vertical jumping speed v_{0y} shown as (3). The vertical jumping speed is used to ensure that the robot jumps to the correct height and is based on the calculated height of the obstacle. If the vertical speed is too high, the robot can be damaged when it lands on the surface of the obstacle after jumping. If the vertical speed is too low, the robot will not make it over the leading edge of the obstacle. Therefore, a specific condition for stable jumping must be imposed. When the robot jumps onto an object, the landing spot x_h is located between the leading and far edges of the obstacle, as shown in Fig. 5. When the robot jumps over an object, x_h is located at a point beyond the obstacle. The robot decides to jump over or land on an object purely from the geometrical information, relying on its camera vision or predefined data for each obstacle. The maximum jumping height $y_{\max}$ must be greater than the maximum height of the obstacle. The maximum horizontal distance is either the far edge of the object if the robot is jumping onto an object or the landing distance if the robot is jumping over an object. Considering this basic understanding of jumping, the restraints are

$$\begin{array}{l} \begin{cases} d_1 < x_h < d_1 + d_2 & \text{for jumping onto an obstacle} \\ x_h > d_1 + d_2 & \text{for jumping over an obstacle} \end{cases}, \\ y_{\max} > h \\ t_h > t_{\max} \end{array} \tag{9}$$

where d_1 is the jumping distance to the leading edge of the obstacle, which is usually determined in advance to ensure that it is safe to jump; d_2 is the obstacle width, which can be given to the robot by the user or estimated from the robot vision process; t_h and $t_{\max}$ are the landing time and the time to reach the maximum vertical jumping height of the robot, respectively; and h is the height of the obstacle, which is measured using the robot vision system. The robot vision determines the jumping distance based on the stereo vision of the Scoutrobot, and delivers this information to the robot controller. The obstacle width can be estimated by commanding the robot to step to the side and record an image of the obstacle. However, this will slow down the robot due to the time required to measure the side of the obstacle before the robot can return to its jumping position in front of the obstacle. Without predefined information for each obstacle, the camera vision system provides the next best source of information.

The landing time t_h can be computed from (1) by substituting the obstacle height h for the vertical distance and considering that the result must be greater than $t_{\max}$:

$$t_h = \frac{v_{0y} + \sqrt{v_{0y}^2 - 2gh}}{g}. \tag{10}$$

Thus, from (9), the vertical jumping velocity v_{0y} must satisfy

$$\begin{array}{l} d_1 < v_{0x} \dfrac{v_{0y} + \sqrt{v_{0y}^2 - 2gh}}{g} < d_1 + d_2 \\ y_{\max} = \dfrac{v_{0y}^2 \sin^2(\lambda)}{2g} > h. \end{array} \tag{11}$$

Given v_{0x}, the corresponding vertical speed is determined by adopting the conditions given by (11). From a practical point of view, the landing time can be set to 20% longer than $t_{\max}$.

The required vertical jumping speed is determined by combining the two constraints. Once this is known, the corresponding duty ratio needed for the solenoid valve to cause the pneumatic cylinder rod to exert the desired force on the ground can be determined from the pneumatic pressure, mass of the cylinder rod, and mass of the robot body.

Since v_0 depends on both the horizontal and vertical velocities of the robot, the jumping height can be controlled by changing these values. Once the robot recognizes the height of an obstacle by utilizing its camera vision and sensors, it can determine the appropriate jumping velocity associated with its traveling speed, resulting in a jumping angle. Fig. 6 shows the air flow rate control by PWM signal on the valve to have different cylinder exerting speed, resulting in different jumping height.

The Scoutrobot must be sufficiently durable to withstand an impact with the ground or obstacle when jumping, which can be determined by analyzing the stress distribution during impact. We used the Nastran software package to determine the stresses on the robot body. The weakest parts were the wheels and wheel gears. The gear axis had the highest stress, but this was not above the maximum yield stress of the material, demonstrating that the structure as a whole is sound (Fig. 6). The robot is equipped with a spring suspended shock absorbers rather than standard absorbers in order to reduce wheel mass. This results in a set of springs resembling umbrella ribs which makes the total wheel weight

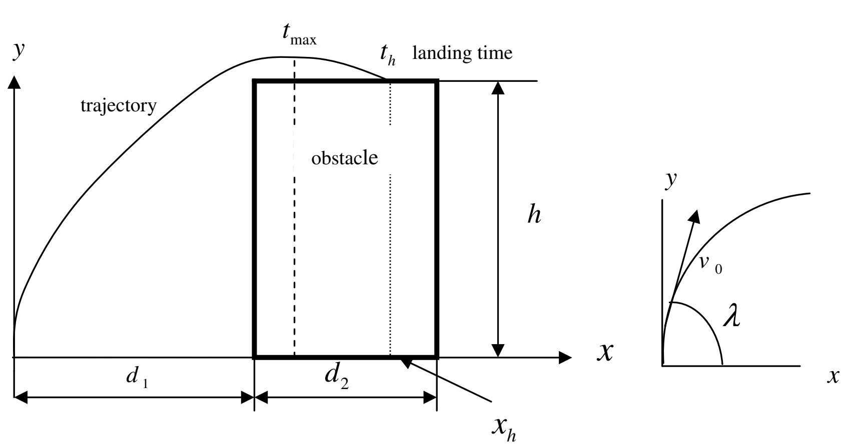

Fig. 5. Determining the jumping velocity.

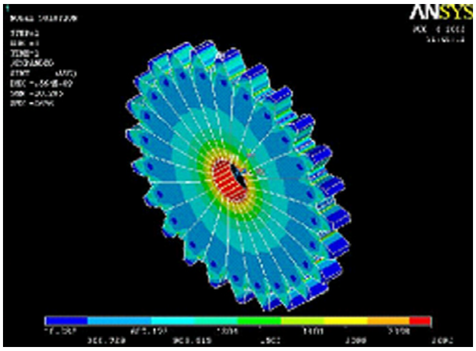

Fig. 6. Stress distribution on the gear connected with wheel.

light. Usually, standard shock absorbers are heavy not to be employed in jumping robot. In addition, the standard costs relatively higher than the umbrella ribs typed spring suspension.

3.2. Jumping performance

Several tests were performed to study the jumping performance of the robot. The dominant factors that determined the jumping height were the cylinder rod speed, cylinder pressure, and robot traveling speed. Fig. 7 shows the experimental and theoretical jumping heights under various conditions. As expected, the jumping height increased with the duty ratio and the height also increased with the traveling speed and cylinder rod speed, which also caused the jumping speed and angle to increase. For theoretical computations, the dimensions of the robot are selected with $m_r = 0.05$ kg, $M = 1.4$ kg, $A = 0.001256$ m^2, $A_0 = 0.0006406$ m^2. The cylinder rod speed was controlled by the duty ratio of the PWM-controlled valve. Fig. 8 shows sequential images of the robot jumping up onto an obstacle. For this example, the estimated distance, corresponding jumping speed, and alignment with the obstacle were all predetermined.

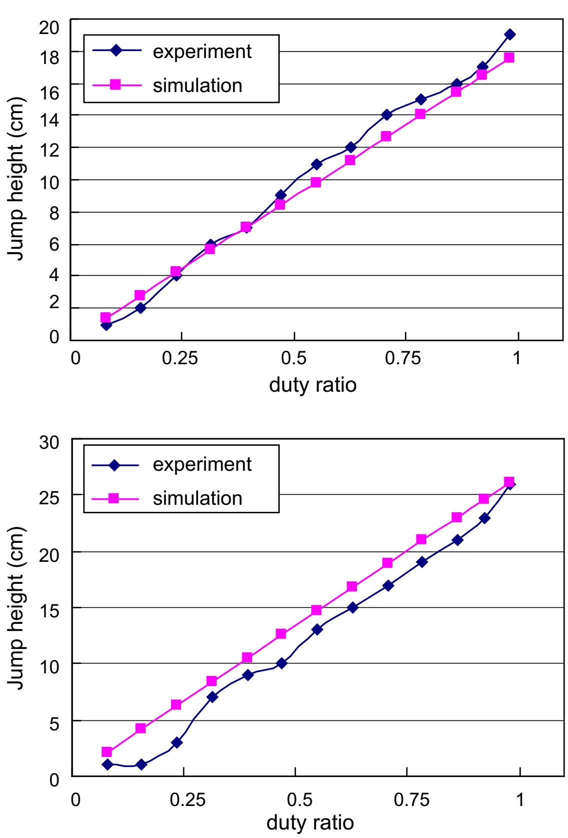

Fig. 7. Jumping height variations by theoretical computation and experiment for various duty ratio: (a) speed = 10 cm/s, pressure = 5 kg/cm^2 and (b) speed = 20 cm/s, pressure = 7 kg/cm^2.

The jumping performance for a stair is tested how effectively the robot can jump on the stair. When the robot detects a stair ahead, the robot poses a jumping on mode. Before jumping, the robot should be aligned to the front line of stair. Without alignment with the stair line, the robot tends to fall back to the floor or lower stair. The average success rate for a jump onto a stair is around 75% while jumping over obstacles comes up to 90%. The difference of success rate for each case arises from different situation. Jumping onto a stair is stricter than just jumping over obstacles because the stair jumping needs a well alignment with the stair front line not to have a falling-back to floor.

4. Recognizing obstacles and estimating their dimensions

The height and distance to an obstacle must be known or at least estimated for the Scoutrobot to successfully jump over it. If the obstacle is too large, the robot must avoid it by detouring. The robot determines whether jumping or detouring is more appropriate based on the distance to the obstacle, height of the obstacle, and shape of the obstacle. In most robots, a set of ultrasonic sensors is used to measure the distance to an object, but the signal from these sensors is sometimes poor due to the narrow view angle and unexpected noise. In addition, the dimensions of an object cannot by easily measured with ultrasonic sensors. Thus, a camera is sometimes combined with the ultrasonic sensors to obtain two-dimensional values of the vertical and horizontal lengths of the object by considering the geometric relation between the camera and image frames, which is known as a forward projection. Fig. 9 illustrates the computations used to determine a point on an object based on the focal length f and the coordinates of the point with respect to the camera frame, Z and Y.

If we use only one camera, the robot can only recognize two-dimensional features, not the depth of an object. To overcome this limitation, binocular camera vision can be adopted for a robotic vision system. By using inverse projection mapping, which identifies three-dimensional geometric information about an object from two-dimensional image coordinates and the mapping between the camera and image frames, the three geometric components of the object can be computed, including the depth of the object. An epipolar line is desirable to save time determining corresponding points between images from the two cameras. However, an epipolar line search yields undesirable noise signals due to incomplete camera setups or a camera selection based on CMOS. As an alternative, we set the two cameras parallel to each other so that the distance to the object Z can be calculated based on simple trigonometric properties between the object and the camera coordinates (see Fig. 10):

$$Z = \frac{Tf}{d}, \tag{12}$$

where T is the distance between the two cameras, f is the focal length of the cameras, and d is the horizontal image disparity $(x_l - x_r)$ between the two image coordinates of a point on the object.

Once the distance to the obstacle is calculated, the starting point for a jump can be predetermined as a fixed value located in front of the obstacle. Then the vertical speed of the robot body can be calculated according to the height of the obstacle. If the

Please cite this article in press as: Kim DH et al., Mechanism, control, and visual management of a jumping robot, Mechatronics (2008), doi:10.1016/j.mechatronics.2008.05.004

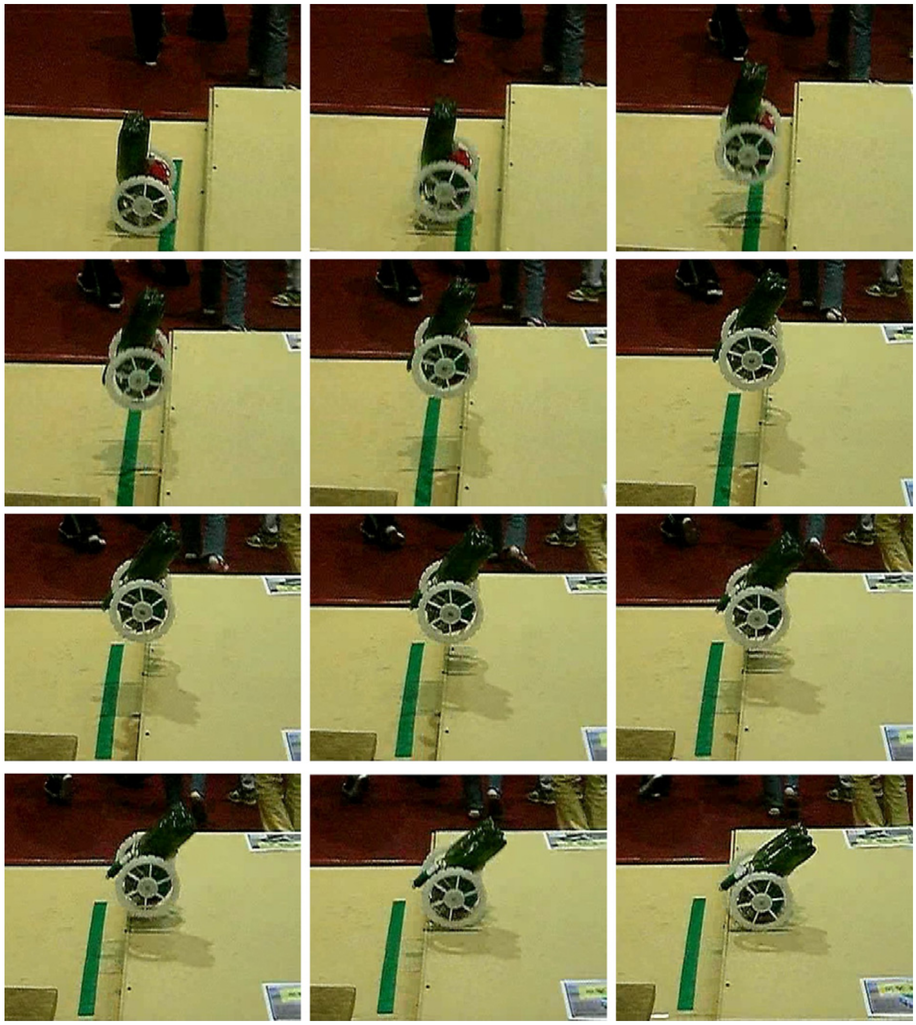

Fig. 8. Sequential images of the Scoutrobot jumping.

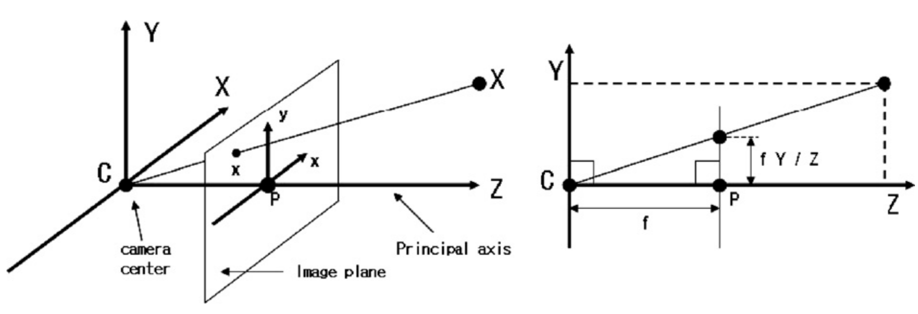

Fig. 9. Measuring the dimensions of an object using the image frame information.

obstacle has multiple levels, such as stairs, the first jump from the ground is performed by computing the travel speed and vertical jumping speed. The following jumps are performed by rotating the cylinder body to obtain an appropriate angle to jump to the next level since the robot will not be able to attain its desired traveling speed in the available space between steps. When an obstacle appears in front of the robot, it will calculate the distance to and the height of the object as it approaches the starting point for the jump. If the object is small enough to jump over, the robot can adjust its traveling and vertical jumping speeds as required. The higher the object, the higher the vertical and horizontal speed assigned by the robot controller will be, which increase the jumping velocity (v_0).

The height of the object is determined by cameras and ultra sensors. The distance to the obstacle is obtained by stereo vision. If the correspondence between a pair of image feature point is established, inverse projective-mapping holds. By this inverse mapping, the geometry of obstacles in a three-dimensional space, particularly the distance to the obstacle is obtained. Once the distance is estimated, the height of obstacle is computed from calculating the pixel image size displayed on screen. The estimated value by vision has much error as the distance is far away from

Please cite this article in press as: Kim DH et al., Mechanism, control, and visual management of a jumping robot, Mechatronics (2008), doi:10.1016/j.mechatronics.2008.05.004

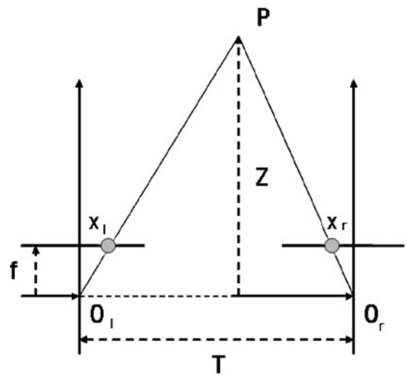

Fig. 10. Geometric properties of an object between two cameras.

the obstacle, hence a fixed distance to the obstacle is always selected before jump. The obstacle height H is determined as follows.

$$H = \frac{Zf}{d} \tag{13}$$

Once an object is captured on camera, the vision system first takes a binary process to save memory and to ease image processing. The binary process is based on the threshold value, but this varies on environment. Thus, a histrogram is introduced to determine an appropriate threshold value for captured image. Next, an edge detection technique is employed to calculate the size and shape of obstacles. In this work, Prewitt mask is used because it provides thinner edges compared with Sobel mask. In Fig. 11 two results of edge detection are shown where the left image is done by Sobel mask and the right one is done by Prewitt mask. As seen the results, even after edge detection is done, undesirable noise remains. The noise is eliminated by median filter.

Once the image is converted to edged image, the center of obstacle is computed from maximum and minimum pixel in x- and y-direction, respectively. In other words, the center of obstacle is calculated as follows.

$$x_c = \frac{x_{\max} - x_{\min}}{2}, \quad y_c = \frac{y_{\max} - y_{\min}}{2} \tag{14}$$

The size of the obstacle is computed by computing the area using mass center principle, which simply adds shaded portion in x- and y-direction. The shape of the obstacle is purely determined by vision process. When all edge information on the obstacle is obtained, the computer recognizes how it looks like by comparing pre-stored object information. When the captured object and extracted object from data base show pretty reliable agreement the computer recognizes the obstacle whether is of triangle, rectangle, or cubicle, etc.

Fig. 11. Results of edge detection by Sobel mask (left) and Prewitt mask (right).

5. Three-dimensional robot localization display

As robot technologies advance, we would like robots to become human-friendly objects that interact with their environments rather than simple machines that follow commands. A three-dimensional robot motion display including an environment image tracker and robot–human interconnection are key issues that must be developed. A three-dimensional display permits different perspective or projection views by rotating the robot viewing angle and by zooming in on the scene. For the Scoutrobot, the location or ground condition is displayed after the robot jumps, and live images are provided to the human user. Fig. 12 shows an example of the three-dimensional robot and environment display. Human users can easily view and interpret the robot's movement and corresponding environment, and advanced users can prevent the robot from moving into prohibited areas.

The RMS provides a three-dimensional view of the robot. The robot moves around the environment and encounters obstacles. The location of the robot in the defined space needs to be measured even if it does not cover any environment. The environment around the robot is visually displayed by employing the RMS and location of the robot is estimated by the information of ultra sensors and encoders. In other words, once the robot is placed on certain location the ultra sensors sends the signal and measures the distance of the robot from the wall. Encoders equipped in wheels update the traveling distance for the robot. The RMS is a system to show visually the location of the robot placed in three-dimensional space by employing 3D applications. Also, the trajectory of

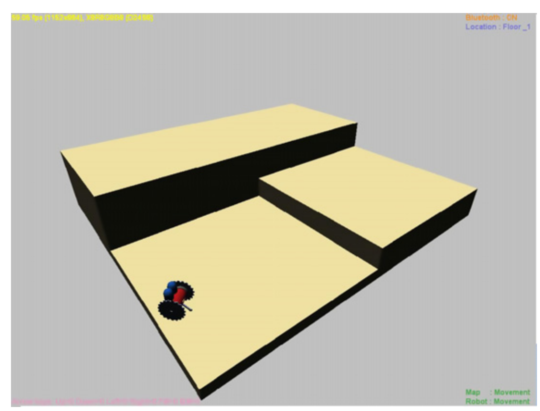

Fig. 12. Three-dimensional display of the robot and its environment.

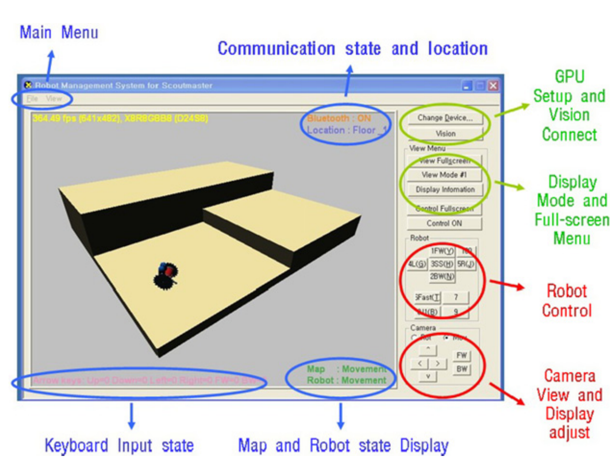

Fig. 13. Direct3D robot management system program.

the robot is displayed and information of the robot is delivered to main computer. Through the RMS operator can easily move the robot to the desired location and view the robot in various direction and location.

The Direct3D RMS, which is capable of viewing the robot and environment using three-dimensional shapes, was used for the Scoutrobot. A three-dimensional projection was included to provide a perspective view. This system provides users with a sense of realism, as if they were navigating with the robot in real time and recognizing the information detected by the robot's sensors, such as position, obstacle appearance, and route condition. The communication and vision programs were integrated so that all program segments could be incorporated into one program. The integrated program is illustrated in Fig. 13. All sub-functions are represented by sub-windows, enabling the program to generate various perspective views.

5.1. Structure of the RMS

One of the basic functions of the RMS is to provide command and image data to human operators without requiring physical intervention with the robot. This requires wireless communication between the robot controller and the user. The system is designed so that the command data are transmitted by a Bluetooth module while image data are transmitted by a RF module. The overall RMS structure is shown in Fig. 14. The remote controller receives input data from the remote user. The RMS program is executed on a PC and delivers control commands to the robot. By adopting this structure, the RMS supervises all functions and three-dimensional displays, and all information such as the current communication status, robot location, and jumping potential of encountered obstacles. This configuration provides users with flexibility when controlling and monitoring the robot without relying on direct observations.

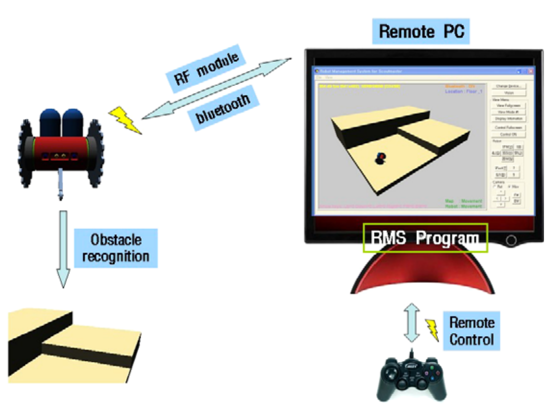

Fig. 14. System structure of the remote RMS.

5.2. Formation of a three-dimensional viewing image

Several coordinate transforms are required to create the robot display, especially the three-dimensional view. To describe the coordinate transforms, world and moving coordinates are defined for the initial phase; view and projection transforms are used for the subsequent phases. The world coordinate system describes all points with respect to origin-referenced coordinates, while the moving coordinate system is defined on the object with respect to local coordinates.

The position and orientation of a local coordinate-based object can be expressed in world coordinates using a translation and orientation transformation matrix. A view transformation is used to transform all objects into camera coordinates and places a viewer in the world coordinate system. In camera space, the camera or viewer is placed at the origin, viewing in the positive z-direction. The corresponding view matrix for the view transformation relocates objects near the cameras from the world coordinate system. The view matrix is derived by combining the translation and rotation matrices for each axis.

We apply a projection transformation after the view transformation. This is similar to adjusting the inside of a camera or selecting a camera lens, and consists of a scale and perspective adjustment. The transformation converts the viewing frustum into a cube. Since the near end is smaller than the far end at the viewing frustum, the transformation effectively enlarges the object.

The far and near transformations move the viewing frustum to new coordinates. When the frustum is modified to a cube, the origin moves from the top left corner to the center. However, the projection matrix adjusts the translation and orientation of the object based on the distance from the camera to the front clipping plane. This gives rise to an indeterminate depth comparison between objects if we do not consider the field of view, which means that every object over a range of distances will have almost the same z value. To overcome this problem, an aspect ratio of the viewport is taken into consideration for each point of the object. Fig. 15 illustrates the projection and perspective projective transformations for the Scoutrobot, and implies that the latter is more realistic because a robot located in the distance is drawn smaller than a robot located close by.

An advanced three-dimensional display technique was developed for the Scoutrobot to provide a more realistic robot display. The three-dimensional projection was a mathematical transformation that projected a three-dimensional point onto a two-dimensional plane. This is the first step of a computer graphics rendering process, where the location and orientation are required.

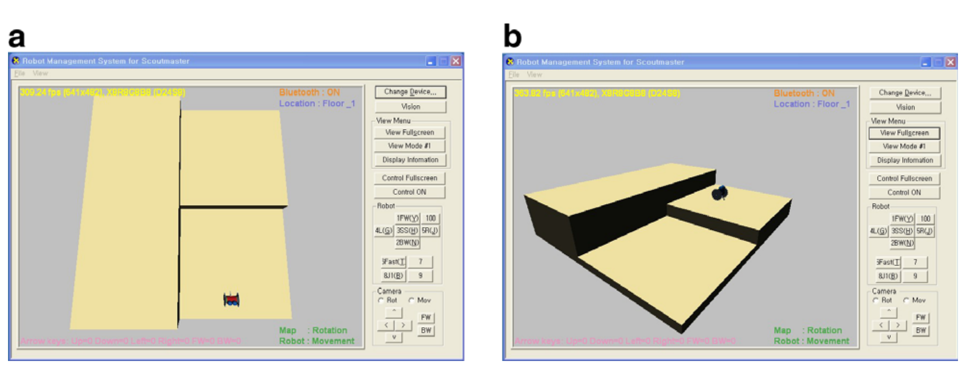

Fig. 15. Robot and map display using a (a) projection transformation and (b) perspective projection transformation.

Similar to the previous schemes, a three-step procedure was adopted for the three-dimensional projection, consisting of world, camera, and perspective transformations. In the world transformation, four matrices were used for the rotation about the x, y, and z axes and the translation. The world transformation was the product of the translation and rotation matrices, with the translation matrix applied first. A scaling matrix was also included to adjust the magnitude of the matrix elements along each axis. The final form is as follows: world transformation = translation × rotation × scaling. After the world transformation was completed, a camera transformation was performed by introducing viewer coordinates and transforming the coordinates of each point from the reference frame to the viewer frame. The camera transformation was the reverse of the world translation and orientation transformations. The result was multiplied by a perspective transformation. In practical computer renderings, the viewer can select the

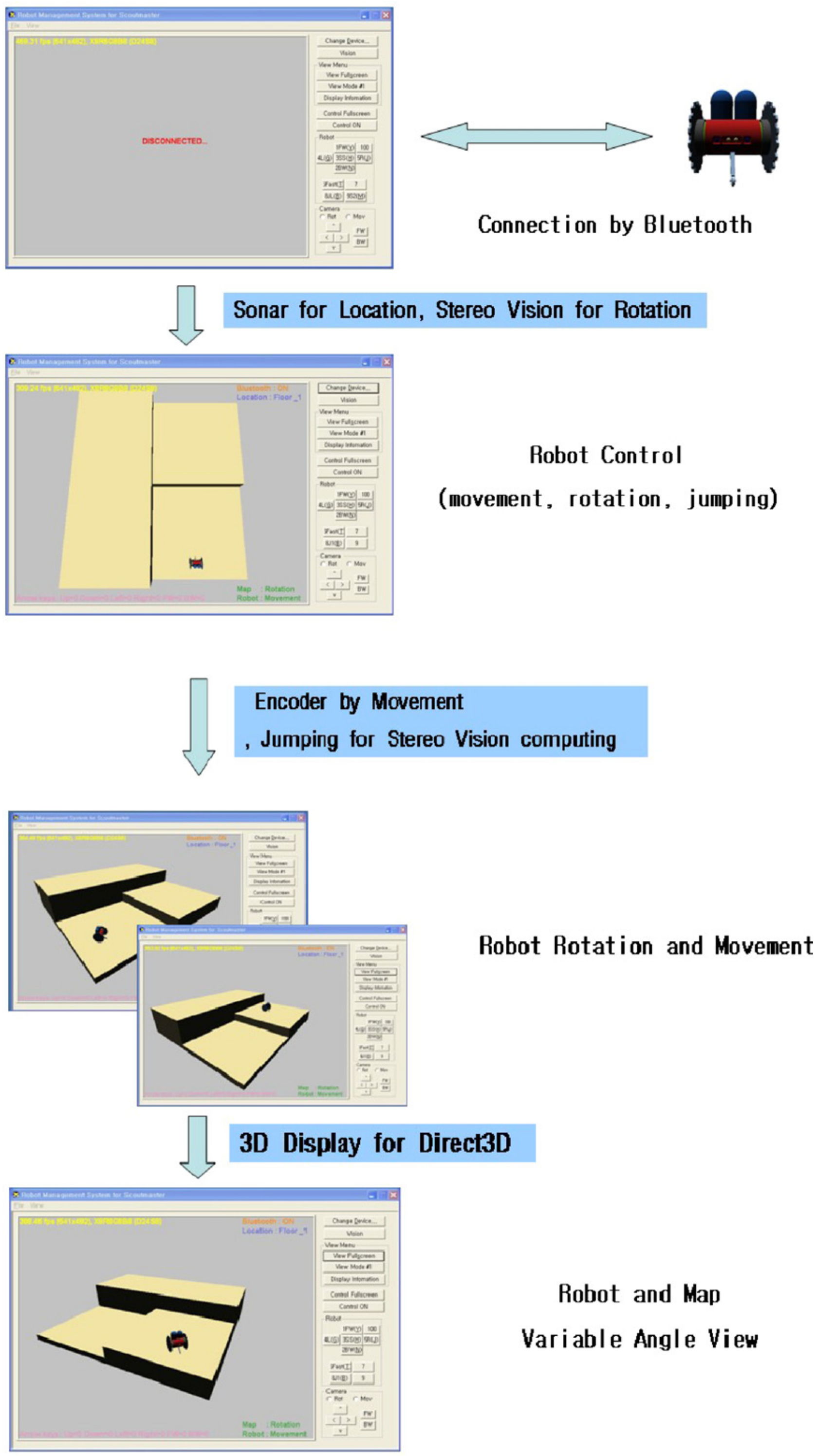

Fig. 16. RMS procedures of the Scoutrobot with three-dimensional projection display.

distance of objects in a simulated view by considering the perspective distortion. Therefore, to obtain a three-dimensional projection on a two-dimensional screen requires the following matrix product: perspective transformation $\times$ camera transformation $\times$ world transformation.

5.3. Application to the Scoutrobot

The procedures used to control and observe the three-dimensional visual display of the Scoutrobot are illustrated in Fig. 16. As the user operates the RMS, the Bluetooth module communicates with the robot using serial connections, and then the location and orientation of the robot is identified on a map based on the information supplied by the ultrasonic sensors and stereo vision. This information is transferred to the RMS and so that the location of the robot can be visualized on a screen for the user. The user can command the robot to jump, move, or rotate to the desired location, or can let the robot assume a specific pose to cope with its environment. The distance moved is estimated by encoders attached to the wheels and is displayed on the screen. The orientation of an object is also estimated by computing its geometry based on image frames. Eventually, the translation and orientation of the robot is displayed by the Direct3D system [8–10]. The Visual C++ based OpenGL tool is utilized in this display representation [11]. Views of the jumping trajectory, location, and orientation of the robot can be shown on the three-dimensional display with the help of the world, view, and projection transformations, and the three-dimensional projection, which represents the live three-dimensional motion of the robot.

6. Conclusions

The newly developed Scoutrobot is distinguished from other robots because its control scheme allows it to jump over obstacles or harsh environments without requiring complex mechanisms employed in other obstacle-overcoming robots. The jumping mechanism and control scheme produce a flexible robot that can adapt to various obstacle heights. The jumping mechanism consists of a pneumatic cylinder, which provides sufficient force and is relatively simple. The jumping height is controlled by a combination of the jumping and traveling speeds. The height of an obstacle is estimated from stereo vision cameras and ultrasonic sensors on the robot. The jumping speed is controlled by assigning an appropriate duty ratio value to a PWM-controlled valve; the ratio varies depending on the obstacle height and traveling speed. Two ultrasonic sensors provide continuous alignment during travel and ensure that the robot has well aligned poses before jumping, enhancing the reliability of the jumping procedure. A RMS was designed to observe and control the robot on a remote screen with a three-dimensional display, which required world, camera, projection, and three-dimensional projection transformations. The environment around the robot is displayed on the screen at the same time so that users can perceive as if they were navigating with the robot. Further study is required to upgrade the Scoutrobot's performance in terms of jumping on or over different types of obstacles. Our group is currently focused on this task. New types of mechanism and control algorithms are being investigated, being introduced in short term.

References

[1] Owens N, Peterson T. Using a reinforcement learning controller to overcome simulator/environment discrepancies. In: IEEE international conference on systems, man, and cybernetics; 2001. p. 1424–29.
[2] Stoeter SA, Burt IT, Papanikolopoulos N. Scout robot motion model. In: Proceedings of the 2003 IEEE international conference on robotics and automation, Taipei, Taiwan, September 14–19; 2003. p. 90–5.
[3] Drenner A, Burt I, Kratochvil B, Nelson BJ, Papanikolopoulos N, Yesom KB. Communication and mobility enhancements to the Scout robot. In: International conference on intelligent robots and systems; 2002. p. 865–70.
[4] Kratochvil B. Implementation of scout behaviors using analog sensing. University of Minnesota – Computer Science and Engineering Technical Report Abstract; 2002.
[5] Stoeter SA, Papanikolopoulos N. Autonomous stair-climbing with miniature jumping robots. IEEE Trans Syst, Man, Cybern–Part B 2005;35(2):313–25.
[6] Kikuchi F, Ota Y, Hirose S. Basic performance experiments for jumping quadruped. In: Proceedings of the 2003 IEEE/RSJ international conference on intelligent robots and systems, Las Vegas, Nevada; 2003. p. 3378–83.
[7] Tsukagoshi H, Sasaki M, Kitagawa A, Tanaka T. Jumping robot for rescue operation with excellent traverse ability, In: ICRA05 Proceedings, 12th international conference on advanced robotics; 2005. p. 841–8.
[8] Luna FD, Lopez R. Introduction to 3d game programming with DirectX 9.0. Wordware Publishing Inc.; 2003.
[9] Pazera E. Isometric game programming with DirectX. MinPress; 2001.
[10] Kelly D. Real time rendering tricks and techniques in DirectX. Course Technology PTR 2002:182–201.
[11] Wright RS, Lipchak B. OpenGL superbible. Macmillan Computer Pub; 2004. p. 189–225.

Please cite this article in press as: Kim DH et al., Mechanism, control, and visual management of a jumping robot, Mechatronics (2008), doi:10.1016/j.mechatronics.2008.05.004