2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)
October 31-November 2, 2013 / Ramada Plaza Jeju Hotel, Jeju, Korea

3D Pose and Target Position Estimation for a Quadruped Walking Robot

Kuk Cho[1], SeungHo Baeg and Sangdeok Park[2]
[1] Intelligent Robot Engineering, University of Science and Technology (UST), Ansan, 133-791, Korea
(Tel : +82-31-8040-6346; E-mail: googi33@ust.ac.kr)
[2] Field Robot Center (FRC), KITECH, Ansan, 113, Korea
(Tel : +82-31-8040-6272; E-mail: {shbaeg, sdpark} @kitech.re.kr)

Abstract – This paper describes a 3D pose estimation method for a quadruped robot and a target tracking method for its navigation. The estimated 3D pose is a key resource for walking robot operation. The pose is applied to two components: the robot's walking control and navigation. The estimated robot pose can be used to compensate sensor data such as camera and lidar. The estimated target is used as part of a leader-following system in a GPS-denied environment. In this paper, we show a 3D pose estimation method for the robot for and target tracking for leader-following navigation.

Keywords – quadruped robot, IMU, object detection

1. Introduction

Unmanned systems have been researched worldwide. Recently, the Boston dynamics DARPA project announced the next Bigdog model, LS3. Research projects on quadruped robots have also been launched in several countries [1]. One of the aims of quadruped robots is to support soldiers. In this application the robot is able to carry heavy loads and follow the soldiers anywhere, including across uneven and rough terrain. This is a key advantage of quadruped robots over mobile robots [2].

To achieve a mission in poor surroundings, the quadruped robot should have two abilities: autonomous driving and an environment recognition technique. All unmanned systems require a specific navigation system with sensor acquisition and data compensation depending on their system characteristics and environments. For an unmanned aerial vehicle, the navigation data are contaminated by system vibration and vehicle rotation and translation. For an unmanned ground vehicle, the data are contaminated by the terrain conditions and vehicle dynamic movement. This situation is exacerbated for the quadruped walking robot. Disturbances occur even during walking on flat ground. On uneven terrain, the errors are much greater. Therefore, the acquired data cannot be exploited because of contamination. The utility of the data can be enhanced by pose noise elimination though robot pose compensation. In addition, the inertial measurement unit (IMU) is thus very important to guarantee stable operation of the quadruped walking robot.

In this paper, a quadruped robot pose estimation and compensation method using an IMU for stable walking and target tracking with the estimated pose are presented.

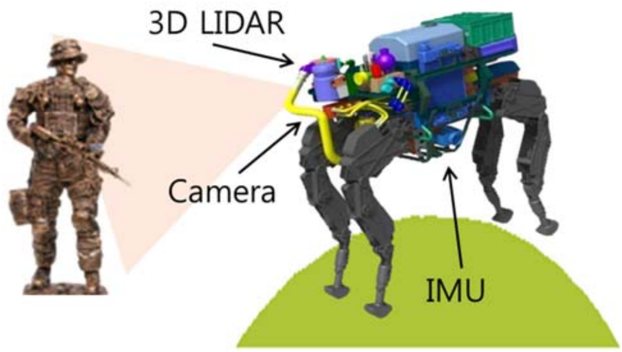

Fig. 1. Overview of the 3D pose and target tracking of the quadruped working robot

2. Robot Pose and Tracking

This chapter explains three steps: a pose estimation method of a quadruped walking robot using an IMU, a compensation method of 3D point cloud data using previously estimated pose data, and a target recognition method for leader-following navigation, as shown in figure 1.

2.1. Pose Estimation and Compensation

The quadruped robot pose is estimated using an IMU sensor that is mounted on the quadruped robot. It is based on the Extended Kalman Filter (EKF). The IMU sensor is composed of 3-axis gyro sensors and 3-axis acceleration sensors. The estimated output state is 6-DOF angular velocity and acceleration. In the EKF, the robot pose is estimated using the Euler angle, and updated using acceleration of gravity [3].

The Euler angle is estimated using the integral of the angular velocity. The state vector is composed of the Euler angle $\mathbf{x} = [\phi \quad \theta \quad \psi\ ]^{\mathrm{T}}$.

$$\hat{\mathbf{x}}_k^- = f(\hat{\boldsymbol{x}}_{k-1}) + \boldsymbol{w}_{k-1} \qquad [1]$$

where the system model $f(.)$ is integrated by the angular velocity from the gyro sensor on the 3D-Euler angle at the k-1-th step as given below:

$$f(\hat{\boldsymbol{x}}_{k-1}) = \begin{bmatrix} \dot{\emptyset} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 1 & \sin\emptyset\tan\theta & \cos\emptyset\tan\theta \\ 0 & \cos\emptyset & -\sin\emptyset \\ 0 & \sin\emptyset/\cos\theta & \cos\emptyset/\cos\theta \end{bmatrix} \begin{bmatrix} \mathrm{w}_x \\ \mathrm{w}_y \\ \mathrm{w}_z \end{bmatrix} \qquad [2]$$

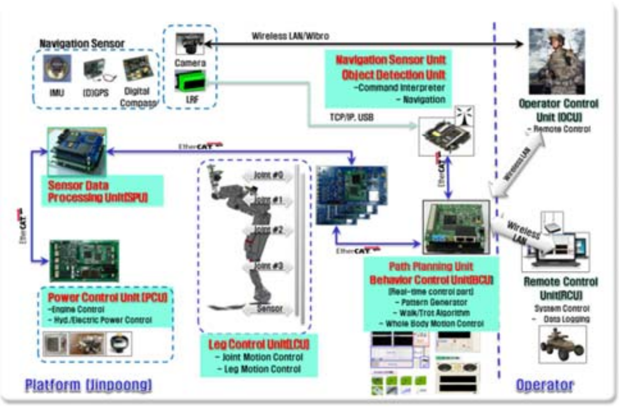

Fig 2. Data flow chart of data communication of a quadruped robot, Jinpoong.

Here, the angular velocities are $\mathbf{w} = [w_x \quad w_y \quad w_z]$. The estimated angle diverges by accumulated calculation error. The error of roll and pitch angles is compensated by the 3-axis acceleration sensors. The system estimates the robot pose based on the kinematic model, as delineated below:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} + \begin{bmatrix} 0 & v_z & -v_y \\ -v_z & 0 & v_x \\ v_y & -v_x & 0 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} + g \begin{bmatrix} -\sin\theta \\ \cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}$$

$$\phi = \sin^{-1}\left(\frac{-f_y}{g\cos\theta}\right), \ \theta = \sin^{-1}\left(\frac{f_x}{g}\right)$$

$$\boldsymbol{h}(\hat{\mathbf{x}}_k^-) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \phi_{acc} \\ \theta_{acc} \\ \varphi \end{bmatrix} + v \qquad [3]$$

Here, $\boldsymbol{f} = [f_x \quad f_y \quad f_z]$ is the acceleration sensor output. The acceleration of gravity is 9.81 in inertia coordinates. The method uses a kinematic equation. It is assumed that the system does not accelerate and hence the acceleration and angular acceleration are not updated. The two estimated Euler values are ϕ_{acc} and θ_{acc}. These values influence the robot pose. We define EKF variables and estimate the robot's pose. Figure 2 shows a data flow chart of data communication of a quadruped robot, Jinpoong.

2.2. Object Detection and Tracking

The camera data and LIDAR sensor data are compensated with previously estimated pose data, and then the leader in front of the robot is recognized for navigation. To recognize an object using LIDAR, the hierarchical clustering method is applied to an object that is a certain distance from the robot. Finally, the measured object information such as the number of points, width, height, and so on is extracted [4].

If any points exist in the initial region of interest area, the tracking object estimation method follows the target object. The object is updated whenever it is measured. It is updated by the sensor's frequency time. Our case is a 10Hz, HDL-32E LIDAR, Velodyne, Inc. It is enough speed for target tracking for a quadruped robot, which is not a fast response system.

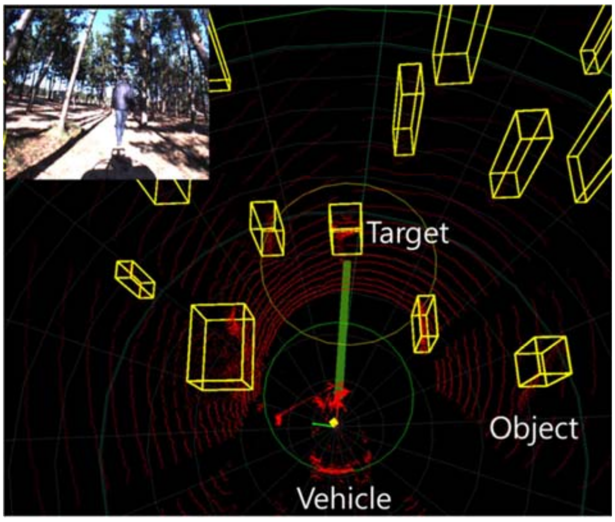

Fig 3. The experimental view of object detection and target tracking with multiple objects and a target that is one of the objects. The left-top inset is an image from the camera mounted on the vehicle.

It is an enhanced method for outliner. It includes noise filtering and 3D hierarchical clustering. It provides reduce computing time for the object detection process. The clustering method is a kind of nearest-neighbor method. It is expressed as follows:

$$\mathrm{d}(\mathbf{C}_i, \mathbf{C}_j) = m\dot{\imath}n_{x_i \in C_i, x_j \in C_j}\{\mathrm{d}(\mathbf{x}_i, \mathbf{x}_j)\} \qquad [4]$$

where $\mathbf{C}_i$ and $\mathbf{C}_j$ are the point cloud data which are i-cluster and j-cluster. $\mathbf{x}_i$ and $\mathbf{x}_j$ are the i-th and j-th point cloud data. The final result of clustering is the recognized object, as shown in figure 3.

3. Conclusion

This paper briefly explains a 3D pose estimatnion method and target object detetion and tracking. We show the real-time architecture for pose estimation and target tracking based on a pose sensor and an image sensor. For further research, we will focus on object classification of a segmented object expressed with point cloud data.

References

[1] Wooden, D., et. al, "Autonomous navigation for BigDog", IEEE ICRA 2010, 3-7 May 2010, pp 4736 – 4741

[2] R. Playter, et. al. "Bigdog," in Proc. SPIE DSS, April 2006.

[3] Alper Ozturk, "Developmnet, implementation and testing of a tightly coupled integrate INS/GPS System", Degree of Master of Thesis.

[4] D. Defays, "An efficient algorithm for a complete link method". The Computer Journal 20 (4): 364–366. 1977