Getting the Position and the Pose Using Stereo Vision

Youngjun Kim
youngjun@stanford.edu
Aeronautics and Astronautics, Stanford University, Stanford, CA 94305

***Abstract*—Controlling a quadruped robot is a challenging problem in Robotics. In this report I present an application to get the position and the pose of the Little Dog robot using stereo vision. I built a vision system on top of the Little Dog robot and reconstructed a 3-D terrain model using a stereo camera. Then I aligned sequential 3-D models to get the position and the pose using ICP algorithm. In the future, this information will be integrated to the controller of the robot as feedback so that it can get over a tough terrain without support of motion capture system.**

Figure 1. The Little Dog Robot, designed and built by Boston Dynamics, Inc.

Figure 2. Motion Capture System, built by Vicon MX system

I. Introduction

The Goal of my project is getting the position and the pose of the Little Dog robot shown in Figure 1, using stereo vision. The Little Dog robot has four legs and the same degree of freedom as a real dog. Its shape resembles a real dog but it doesn't have a head and a set of eyes. Thus, it cannot recognize new terrains, and it only knows the initial coordinates of the terrain which we internally programmed. In addition, the motion capture system shown in Figure 2 gives the position and the pose of the Little Dog robot. If we have a new terrain or the motion capture system is gone, it cannot see the new terrain and we cannot get the accurate position and the pose. My first work was to build the vision system using a stereo camera and integrate it within this system. The stereo camera gathered left and right images and I reconstructed the 3-D model from these images. But as the Little Dog robot was moving, it built sequential 3-D models which were not aligned, and aligning these 3-D models was a challenging problem. I solved this problem using ICP algorithm to align these 3-D models and with this, I could get the position and the pose of the Little Dog robot.

Figure 4. Left, Right Images and Depth Map from a stereo vision, made by Tzyx Inc.

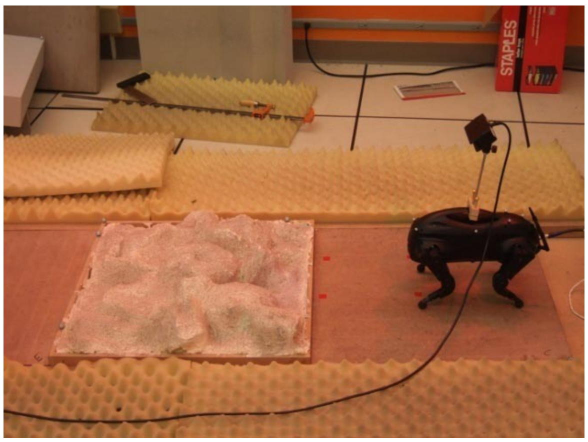

Figure 3. Terrain and the Little Dog Robot equipped with a Stereo Vision

II. Getting the Position and the Pose

As a first step, I set up a vision system and equipped the little dog robot with this system, as shown in Figure 3. Then I built a 3-D model from a series of stereo images. The original data were a series of depth maps Tyzx stereo camera had calculated from left and right images, as shown in Figure 4. I converted these depth maps first to 3-D point clouds then to the trigonal meshes. Then I used the efficient ICP algorithm to align images sequentially. Before I utilized aligning information for getting the position and the pose, I needed to know relative position between the robot body and the stereo camera. Using a camera calibration technique, I could calculate the relative distance. As a result, I could get the position and the pose information of the little dog robot.

A. *3-D Trigonal Meshes from Depth Maps*

Original data was a series of depth maps. I sampled points uniformly from depth map and reconstructed 3-D coordinates assuming perspective camera model [1]. Since the intrinsic parameters of the camera were known, I could have relative world coordinates. Then I used Delaunay algorithm [2] to build trigonal meshes.

There were some noises in a depth map. Among the many kinds of noises, the most prevalent one was salt and pepper noise. I applied the median filter to remove it and I also removed outliers which had long edges in trigonal meshes.

B. *Pairwise Alignment of Consecutive Images*

The dominant algorithm for geometric alignment of 3-D models is the ICP (Iterated Closest Point) algorithm. ICP algorithm takes in two triangular meshes and finds the translation and rotation between those two, as shown in Figure 5. Among a number of variants in ICP algorithm, I used the efficient ICP algorithm developed by Rusinkiwicz and Levoy [3]. The Little Dog will need to be able to understand the 3-D structure of the terrain as it moves on it, meaning that the reconstruction of the model should be real-time. For that purpose, I chose to use efficient variant of the ICP algorithm [3].

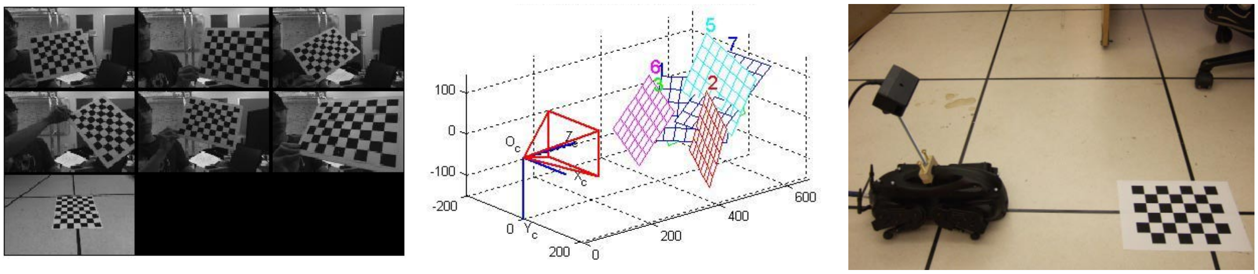

Figure 6. Calibration Images, Extrinsic Parameters (camera-centered), Setup for getting the relative distance

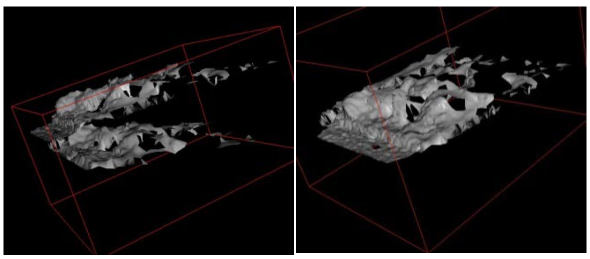

Figure 5. Unaligned and Aligned Meshes

This variant ICP improved performance in speed by choosing a projection-based algorithm to generate point correspondence, the point-to-plane error metric [7] and the standard "select-match-minimize" ICP iteration [7]. For the other stages that are not critical to high-speed, it uses simplest ones, random sampling of points, constant weighting of pairs, and the distance threshold for point rejections [3]. In addition, I found parameters of ICP algorithm to achieve high speed and accuracy by experiments.

The efficient ICP algorithm gives the transformation between the two range maps. The ICP algorithm first pairs points in one mesh with nearby points in the other and then finds a rigid 3-D motion that aligns the paired mesh points iteratively. The ICP needs initial rough registration to avoid failure to find the global minimum and achieve high speed. I assumed that the transformations between the consecutive images are small enough that initial guess of no transformation is acceptable.

C. *Relative Distance between the Robot Body and the Camera*

Using aligning information between two images, I can get a translation and angle between initial and next positions of the camera. But the goal is to get a translation and angle of the robot body. To get this information, knowing the relative distance and angle between the robot body and the camera is important. Because measuring these data directly is inaccurate, I used a calibration technique to measure the configuration. As calibrating the stereo camera for taking several pictures of a known square grid panel, I could get extrinsic parameters, and after setting up the robot body and the panel in a known position, I could get the relative distance and angle between the robot body and the stereo camera using the extrinsic parameters, as shown in Figure 6.

III. Experimental Results

At first, I compared aligned terrain meshes with a terrain model provided by IPTO (Information Processing Techniques Office) based on the given information about a terrain board. I used the volumetric method [8] to merge a set of aligned terrain meshes. RMS error between a given terrain model and a corresponding merged terrain mesh is within 3mm.

Secondly, I compared a position and a pose from the vision system with these from the motion capture system. As the little dog robot walked across a terrain, the vision system calculated a position and a pose ever y one second. Before the little dog robot finished to cross a terrain, the vision system gave 20~30 position and pose data on average. An average error of poses is within 3 degrees on each axis. This result is quite accurate. And an average error of positions is varied between 5~30mm. I think the errors of positions are mainly due to a calibration error and a motion capture system error.

IV. Future Work

In the future, the position and the pose information from the vision system will be integrated to the controller of the little dog robot as feedback so that it can get over a tough terrain without the support of motion capture system.

The errors of positions from the vision system still needs to be investigated where the errors are from and it might need to be decreased for giving this information to the controller of the little dog robot as feedback.

References

[1] E. Trucco and A. Verri, Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.

[2] B. Delaunay, Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793-800, 1934

[3] S. Rusinkiwicz and M. Levoy, Efficient Variants of the ICP Algorithm, Proceedings of the Third Intl. Conf. on 3-D Digital Imaging and Modeling, 2001.

[4] M. Deans, C. Kunz, R. Sargent and L. Pedersen, Terrain Model registration for Single Cycle Instrument Placement, Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, 2003.

[5] Yi Ma, Stefano Soatto, Jana Kosecka and S. Shankar Sastry, An Invitation to 3-D Vision, Springer, 2006.

[6] B. J. Besl and N. D. McKay, A method for registration of 3-d shapes, IEEE Trans.Patt. Anal. Machine Intell., 18(5):540-547, 1996

[7] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, Image and Vision Computing, 10(3):145-155, 1992

[8] B. Curless and M. Levoy, A Volumetric Method for Building Complex Models from Range Images, Proc. SIGGRAPH ’96, ACM, 1996