Project Status Report: Tiny Motion Capture System using Wii Remote

Wei Zhang, Yi Gong
Department of Computer Science
University of California, Santa Barbara
wei, ygong@cs.ucsb.edu

1. Introduction

Motion capture technique is a well studied area in the past decades, many commercial systems are available in the market. However, most of them are complicated and very expensive, built only for professionals.

The Nentindo Wii console has a built-in IR sensor in its remote controller, with on-board hardware processing capability to track up to four IR markers simultaneously. This give us the possibility to build the cheapest motion capture system ever, and if consider that Wii has already been sold to over 20M families, our system will only cost them only a few IR markers. Such kind of low-cost and tiny motion capture system can be used as a basis to build many new applications for home interactive entertainment.

Several interesting application of Wii remote has been discovered and result in great impact, the most famous ones are the head tracking and multi-touch whiteboard developed by Johnny at CMU[4] [5]. However, all of current applications only use a single Wii remote for its built-in 2D tracking capability.

In our project we are going to build a 3D tracking system using two Wii remote, this is a classic 3D reconstruction from multiple view problem which has been explored by many researchers. For the 2 view cases, a lot of studies have been made on the fundamental matrix and calibrated or uncalibrated image marching [7],[6] [9] [3]. [2] is also a good book that cover these issues. In our method, we will calibrate our camera by [8]'s algorithm first, then calculate the fundamental matrix and reconstruct 3D refer to [1].

2. Our Approach

2.1. Hardware Platform

Our system consist of two Wii remote as IR cameras, and four IR leds as markers. Every Wii remote contains a 1024x768 infrared camera with built-in hardware on-board processing to provide tracking of up to 4 points (e.g. IR led lights) at 100Hz. In addition, Wii remote can be connected to the PC via bluetooth. Our marker is just a regular IR led attached to a AA battery, the most sensitive wavelength for Wii remote is between 800 and 950 nm.

2.2. Softawre Infrastructure

We want to build our software as an open 3D tracking library so that other developers can use it to make their own applications. In this project, the software infrastructure can be described from bottom to top as following:

- The bottom level is WiiRemoteLib. This is the hardware driver that enable us to communicate and control the Wii remote through bluetooth wireless, the 'image' we get from Wii remote is actually coordinates of points in camera's image plane.
- Camera calibration provides the intrinsic and extrinsic parameters of two cameras during initialization.
- At the stereo vision level we use the camera parameters and detected marker positions to produce their 3D position.
- Finally the tracking results are provided to our interactive application, for example, to control the role's movement in a fps game base on the player's certain specific postures.

3. Stereo Vision

The stereo matching problem consists in finding pairs of detected markers in different images captured at the same time with different viewpoints such that each pair corresponds to the projections of the same scene point. Unlike common commercial motion capture systems using a cloud of markers, we use only four, so most of time the basic technique for spatial correspondence will work just fine. But when there are some special cases happen, for example, two epipolar lines constructed from image1 are very close in image2, or two points in image2 are too close to the same epipolar line, then temporal correspondence information can be used to improve the correctness of matching.

3.1. Camera Calibration

We will first attach four LEDs on a square plane, then use the OpenCV"s cvCalibrateCamera2 and cvFindExtrinsicCameraParams2 [8] functions to get our camera"s intrinsic and extrinsic parameters.

The camera calibration will be done automatically in the initial phase of running our application.

3.2. Spatial Correspondence

In the 3D reconstruction step, we will find the correspondence between points pairs on the two image plane by the algorithm based on Epipolar Geometry. From the definition of Epipolar Geometry, we can see that the projections of point p onto two image plane, p1 and p2, have the relationship that:

$$p^T F p'^{-1} = 0$$

where F is a 3 x 3, rand 2 matrix, which is also called fundamental matrix[1]. Fundamental matrix has proven to be related with essential matrix E:

$$E = t \times R$$

, where R and t is the rotation and translation from first camera to second camera.

Since we"ve get our cameras calibrated, the calculation of the fundamental matrix becomes straightforward. After getting the fundamental matrix, we can relate the tracked points in two images from the two cameras. After finding the correspondence between the points in two image planes, calculating the 3D position of marks is not hard to handle.

3.3. Temporal Correspondence

The temporal correspondence problem (tracking) involves matching two sets of 3D points representing detected markers at two consecutive frames, respectively. Given the correspondence between consecutive frames, a time series of 3D coordinates is built.

In order to match points in image to objects, we must first estimate where the objects are supposed to be. Since our system are used on low-speed human motions, the mathematical model of object movement can be simulated with a uniformly accelrated rectilinear motion. Based on this model, there exist many object tracking and position estimation algorithm, we plan to use the classic Kalman filter to calculate the estimation.

So our stereo matching algorithm can be described as follows: For each time *t*, we calculate the position prediction for every points based on its previous position and estimation of speed. Then we project the points in image1 to epipolar lines in image2, and match the points in image2 to its corresponding lines to build the spatial correspondency. Finally we project the estimated position to image2, match them with the points in images to build the correspondency between points and physical objects.

4. Planned Experiments

Our experiment includes two phases. Phase1 would be evaluating our stereo vision algorithm. For the camera calibration, we will use two cameras, with 4 markers on a square board to measure the error of openCV's algorithm. For spatial correspondency algorithm, we are going to use 2 cameras and 1 marker, for different angle and distance, measure the average error and jitter of calculation result and physical measurement. For testing the temporal correspondency, we plan to use 4 markers, stick them to two hands and elbows of a person, and see how well can our algorithm match markers to their corresponding places, important parameters including the distance between the person and cameras, and marker's moving speed.

Phase2 would be building demo applications to show the potential of this system. Base on previous experiment, this first demo would be place markers on the arms to show control of a simple boxing/acting game. If we have enough time, a complicated application could be attach 4 markers to legs and arms respectively, then try to match the person's posture to our pre-defined templates, each template will correspond to a motion(such like jump, run, squat) or a command of the role in the virtual world.

The expected result is that our system should work well when get perfect input – no occlusion and motion is normal speed. We will try to strengthen our application by estimate points' position, and tolerate error of detected points data by march the most likely points when searching correspondence. These kind of experiments of robustness is not predictable yet. But we want it can handle short and partial occlusion and have the ability of error tolerance.

References

[1] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 563–578, London, UK, 1992. Springer-Verlag. 1, 2

[2] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2003. 1

[3] R. I. Hartley. In defence of the 8-point algorithm. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 1064, Washington, DC, USA, 1995. IEEE Computer Society. 1

[4] J. C. Lee. Head tracking for desktop vr displays using the wii remote. Http://www.cs.cmu.edu/ johnny/projects/wii/. 1

[5] J. C. Lee. Low-cost multi-point interactive whiteboards using the wiimote. Http://www.cs.cmu.edu/ johnny/projects/wii/. 1

[6] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. pages 61–62, 1987. 1

[7] Q. Luong and O. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis, 1995. 1

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 1, 2

[9] Z. Zhang, R. Deriche, O. D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995. 1