Using C++ ARToolkit in Determining Knee Joint Angles for Real-Time Gait Analysis (May 2009)

Linfang Yang
Department of Mechanical Engineering
University of Utah

This paper describes a method to determining knee flexion/rotation angle in real-time for gait analysis, using transformation matrix between tibia and thigh derived through ARToolkit marker system. This simple gait analysis system contains a stand-alone exe file, and a markers set that need to be put on the patient’s leg. By using ARToolkit to determine marker coordinate systems with respect to camera coordinate system, the joint angle can also be derived, and will be shown on the screen in a user-interface panel. Experiment result shows that, after applying a low pass filter, joint angles data collected by the system is within the acceptance range, and the measurement is repeatable.

I. INTRODUCTION

In traditional gait analysis, motion capture technology is very important. They mainly used to capture rang of motion, time/length of movement. Existing devices are mostly for hospital use, such as electrogoniometer, triaxial goniometry, acceleomometry.

This paper introduces a software based gait capture system. It just need a camera, a marker set, and a computer, then the software run on the computer can recognize the marker attached to the subject’s leg, and determine joint angles. This system can be easily used at home. The data collected can be saved in a separate file for the PT doctor to do further analysis afterwards.

The second section of the paper described the hard ware needed for the system, and the algorithm for developing the software. Then the third section showed brief experimental result, and discussed the precision of the system. Then the fourth section talked about the conclusion.

II. APPROACH

A. Hardware

A camera and a 4-marker set will be needed for motion capture.

1) Camera

A webcam, with 1.0-megapixel photo resolution or above. Other kind of camera that can link to the computer through a USB port will also be appropriate.

2) Markers

A 4 set markers, size 1.5 inches by 1.5 inches, are needed, one for the thigh, one for knee joint, one for tibia and one for the ankle joint. The Markers are printed images on flat plastic board.

Any occlusion of the marker shape will result in inaccurate result, so keep the markers being seen by the camera, and keep the markers’ shape flat is very important.

B. Software

ARToolkit was used as an additional library to program in Microsoft Virtual C++.

1) Marker Detection

As stated earlier, the bottom light sensor is responsible for edge detection, which is done by selecting edges with a distinguishable contrast from the grid centers. When the sensor reading changes from low to high (i.e. carpet to tape) the edge is detected. The same approach is used for pausing the sensor at the edges for ultrasonic scanning, using the front light sensor.

2) Marker Localization

A 5x6 grid map, figure 2, with a 12 inch resolution is designed to locate the robot while mapping the environment. The robot starts by heading in the positive y direction with the positive x direction is on its right. In the program, turning is always set to the right such that a full revolution is obtained by 4 right turns for calculation purposes. In reality, the robot can turn to the left by setting the number of turns to three. Knowing the total number of turns allows identifying the robot heading and thus accounts for the correct x or y increment, i.e. if the robot is heading in the positive y direction then makes a right turn the program will increment the positive x direction by 1 once the bottom light sensor detects the grid edge.
3) **Thigh/Tibia coordinate system**

The transformation matrix for each marker is the transformation from the camera to the marker, to get the transformation matrix from tibia marker to the thigh marker:

\[ \text{thT} = \text{thT}_c * \text{cT}_t = \text{cT}_\text{th}^{-1} * \text{cT}_t \]

'ti' for Tibia, 'th' for thigh, 'c' for camera

4) **Calculate Joint angle**

The knee joint angles are defined as:

Knee flexion: tibia rotation along the knee hinge axis which goes through the lateral and medial epicondyle, which in this paper the z axis.

Tibia internal/external rotation: tibia rotation along the axis goes through hip joint center and knee joint center, which in this paper the y axis.

Tibia rotation along the axis that go through the knee center and center of patella, which in the paper the x axis.

III. **RESULTS**

Several experiments were conducted to test the system:

![Fig. 3. Ultrasonic sensor readings when surrounded by three obstacles at different distances; the obstacle in front is the nearest then the one on the right and finally the one on the left.](image)

In figure 4, it can be seen that the grids with obstacles are shaded with dark blue. The surrounding grids where marked with a lighter shade of blue depending on the probability. This error is mainly due to the uncertainty in the ultrasonic sensor measurements. Also when adding the measurements to the map (the green points) one can see that there is some sort of a shift in the measurements due to the fact that the robot is not always centered between obstacles when performing the ultrasonic scans.

![Fig. 4. Global map: obstacles shaded dark blue and green points refer to ultrasonic readings.](image)

IV. **CONCLUSION**

The goal of mapping with known robot localization was achieved using occupancy probability grid technique. Edge detection was used to compute robot location along the path. Obstacles were detected using fairly consistent ultrasonic measurements. A global map containing obstacle locations was generated. Some issues that might be worth revisiting are motor driving, accurate detection of obstacle shape and exact location and improving the map accuracy to only mark grids with obstacles as occupied. Never the less, this approache performed fairly well for the purposes of this paper.

REFERENCES
