Term Project
Virtual Environment CIS 6930, SangHoon Han
Due on Thu. Dec.6, 2007

Title: Virtual grove and virtual harvesting robot
The purpose of this term project is to develop virtual harvesting robot which is controlled by using markers. The function of tracker is to detect markers and estimate its positions and orientations (transformation matrix). The estimated transformation can be used for control input in another virtual environment. In this project, these inputs are used to explore and signify target position and orientation in the constructed virtual environment.

Technical component:
Two programs that are working independently were developed by using two different IDEs. Marker tracking program was coded by modifying an example of ARToolkit (2.72 for win32) in the VS C++2005 and virtual robot simulation program was implemented by referring a couple of examples of GLScene Library (1.0.0.7.14 for Delphi 2006) in the Turbo Delphi 2006. These programs communicate through window messages.

- Cubic Marker Tracking: novel interaction method, at least two 6 DOF, calibration
  - Local Marker: Local marker is used to position end-effector based on camera, and AR provides the transformation of markers. Cubic marker has a couple of markers on each face. Since marker in AR is plane, the rotational movement of marker is constrained. Although one of marker disappears in the cubic marker, other marker may keep being tracked. I used two markers here for convenience. You can see that markers having different position and orientation are indicating the same position as shown Figure 1 (a) and (b). Side marker of Figure (b) is used to send command like a clicking of mouse. When this marker rotates over 90deg, the current position is saved.

- Global Marker: Global marker is origin of world coordinates, and AR provides the transformation of camera not marker. Camera transformation is solved from inverse of the global marker. Theoretically, when both global and local markers are stationary and only camera moves, the local marker should be still stationary with respect to the origin of world

Figure 1 (a) Cubic Marker; front marker disappeared. (b) Side marker for signification

- Global Marker: Global marker is origin of world coordinates, and AR provides the transformation of camera not marker. Camera transformation is solved from inverse of the global marker. Theoretically, when both global and local markers are stationary and only camera moves, the local marker should be still stationary with respect to the origin of world
coordinates in the 3D view. However due to various error, the tracking is pretty unstable even if mean function is used to be robust. Therefore, for a time, I’m going to fix camera and move only cubic marker.

Figure 2. Input coordinates from each marker

Robot Control: inverse kinematics

Haptic is one of input devices which input 3D position and orientation information into computer directly. The angle information sent from haptic is used to place 3D pointer (6DOF) by using forward kinematics. Inverse kinematics is a process to solve the angles reversely based on given position and orientation. The most difficult thing in this project was to solve this inverse kinematics of a robot arm. Two solutions to the inverse kinematics are known, inverse transformation method and closed loop method. Inverse transformation method is to find parameters step by step by multiplying inverse transformation both sides of equation. Closed loop method supposes 7th link between end-effector and base joint. Then the movements of links are constrained within spherical. Both methods give us 8 possible solutions about 6 DOF robot arm. Furthermore the robot model used in this project has 7 DOF. It was very difficult to not only figure them out, but also apply them to this robot model. Thus I devised a simpler own method by myself (I named Triangular Area Method). My approach may not be applied to general robot arm. This approach gives us a unique solution about this robot with the previous position of elbow joint. Even though 6 DOF of end-effector are given, wrist position is determined. But elbow position is not determined uniquely. i.e. the possible positions of elbow can be placed along the edge of disk, which is perpendicular to a line connecting between wrist and shoulder as shown in Figure 3.
To solve this uncertainty, I set new closest position of elbow intersecting between the edge of disk and a projected point which the previous elbow position is projected on the disk plane. Therefore the elbow position may be different every positioning with respect to the same position. Once elbow position is determined, 3 triangles are formed. Since length of each edge is known, the inner angles are also determined. Revolute joint is one of them. Axial joint is a angle between normal vectors of triangle plane. Angle between two vectors is

$$\theta = \cos^{-1}\left( \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||} \right)$$

One of difficult things in this inverse kinematics was that the angle between vectors is always positive sign. This equation does not indicate sign of angle based on one of vectors. I had to know the sign of angle so that the axial joints operate within finite range. To determine sign of angle, plücker line coordinates system was applied. Since plücker line defines moment of line as well based on origin, the angle of given two line vectors represented with plücker system can be either positive or negative sign. A plucker line is defined by

$$\begin{pmatrix} S \\ S_{0L} \end{pmatrix} = \begin{pmatrix} P_2 - P_1 \\ \frac{||P_2 - P_1||}{S \times P_i} \end{pmatrix}$$

Where $S$ is direction of line, $S_{0L}$ is mement of line with respect to origin. Then the sign of angle is

$$\text{sign}(\theta) = \text{sign}(S \times S_{0L} + S_2 \times S_{0L})$$

The target position should be within length of upperarm and forearm. This inverse kinematics was developed with Matlab first and then ported into Delphi. Figure 4 shows visualized inverse kinematics in the Delphi.

Figure 3. Circle indicates possible elbow position.
Compelling experience: cohesive background story

 Scenario: cohesive background story

Some groups who have a plan to employ a robot for unmanned harvesting system can simulate their algorithm without using a real robot in the developing step. This virtual system may help them to reduce operation cost and enhance the efficiency of development. It would be good for propaganda effect as well.

Unmanned scouting system such as mars explorers should be able to recognize their environment by itself. The acquired information makes it possible to visualize the real environment with virtual environment and human can handle it remotely. Once visualization is established, we can be applied it to remote control application.

 Physical Realism: Timer, Collision, Universal gravitation

Once a fruit is placed within the range of end-effector, collision event occurs and the fruit color is changed. You can save multiple positions. During harvesting, the belonging of fruit is changed into the end-effector based on the collision. Figure 5 shows that the end-effector is detecting a fruit.

After releasing fruit, the delivered fruit belongs to basket. At this time, the fruit moves with acceleration of gravity like Figure 6. Every animation was implemented by TTimer component.
Figure 5. Detect of Collision

Figure 6. Harvesting and dropping with acceleration of gravity
Robot Design: base robot model

The modeled robot arm used in this project is based on R1207a model of Robotics-Research™ as shown in Figure 7 (a) and (b). This model has 7 DOF (3 revolute joints and 4 axial joints) similar to human’s arm. The range of axial joint is -179~179° and revolute joint is 0~179°. Vehicle mode is originated from also existing vehicle. Every size of equipment was approximately measured by hand.

Figure 7. (a) Stationary setup of R1207 (b) Mobile setup of R1207

At the beginning, I tried to model it with VRML then load it. But this model was simple enough to model with VCL in Delphi. Figure 8 shows modeling environment of VRML and VCL.

Figure 8. (a) Robot Design with VRML. (b) Robot Design with VCL in Delphi
Visualization and User Interface

I put a large number of controls so that users can get operating information and choose various options. You can change the viewpoint or zoom-inout by using mouse in the 3D view panel. Since too many of visualization features were added, it also provides a simple rendering mode for a little slow computer system.

- Background image: Since the motivation of this project is citrus grove, I cut into 4 pieces of wide panoramic field and then paste them on the background box. If you take a look carefully, you may feel edge of the skybox. To prevent seam line between faces, the skyline should be aligned at the center of face. I added a lens flare effect of the sun and clouds because of outdoor scene.

- Imposter: To represent grove scene, plenty of trees are needed. Since a tree model consists of many meshes or polygons in general, it is expensive computation to represent whole grove. Imposter technique renders 3D models into rasterized texture images in order to make it faster. I could represent grove scene with only one canopy.

- Metallic texture: I’ve tried to represent metallic surface of robot. There are two way of mirror effect. The mirror effect applied to robot arm is static mapping, and for teapot it is dynamic mirror mapping.

Figure 9 shows the special effects used in the virtual environment.

- I placed a basket and a vehicle on which robot is mounted like this picture. In addition, I added another camera view in end-effector like a real system as shown Figure 10. This additional camera view help to aim target fruit.
Future work: Obstacle Avoidance, Connection to real robot

This virtual grove and virtual harvesting robot help us to explore the real grove and to simulate the performance of robot without going to the spot. Thus we can apply this virtual environment to tele-operation. This robot system provides us SDK and APIs. To control the real robot remotely, the remaining work would be connection and communication between this virtual environment and real robot system through protocols such as TCP/IP. The rest of remaining works can be obstacle avoidance, path planning and other virtual sensor.

Reference
- Kinematic Analysis of Robot Manipulators, Crane and Duffy
- GLScene Visual Component Library