Kinect Controlled Robotic Arm

CS308 Project - 2012

Group 10

Aakash Rao N S	(09005069)	Aayush Singhal	(09005041)
Sivakanth Gopi	(09005043)	Sriram Bhragav K	(09005067)

under the guidance of
Prof. Kavi Arya

Motivation

- Controlling a robotic arm with a traditional controller can be very difficult
- Non-intuitive and increases the complexity of operation

What we need is a Natural User Interface (NUI)

- With NUI the human arm can directly control the robotic arm which tries to imitate its movements
- Thus controlling the arm becomes very intuitive
- Opens a plethora of applications

Problem Statement

Provide a NUI for remotely controlling a robotic arm

- robotic arm should imitate the movements of the user arm
- use kinect to capture human arm movements
- develop an algorithm to convert human arm movements into robotic arm movements in a way that makes the control intuitive

Description

- user moves his arms to control the arm
- movements and gestures of human arm are captured using kinect which are sent to the controller (computer)
- the controller uses an algorithm to convert this movements into movements of robotic arm
- robotic arm imitates the the movements of the human arm
- user can monitor his movements in a GUI

Functional Requirements

1. Robotic Arm
   - The robotic arm should have sufficient degrees of freedom to accurately imitate the human arm
   - The response time of the robotic arm should be less to provide fast real time movement
2. 3D capturing device
   - The device should capture the 3D positions of the joints of the human arm in real time
   - It should have a decent of frame rate of atleast 15 fps

Work Execution (1/2)

- Develop an API for controlling the robotic arm (2 days)
- Develop Kinect interface and some related classes for getting positions of human arm joints and required angles(1 day)
- Mapping each axes of the robotic arm to those of human arms, testing and fine tuning(1 day)
- Developing and implementing an improved algorithm for providing an much more intuitive user control (3 days)

Work Execution (2/2)

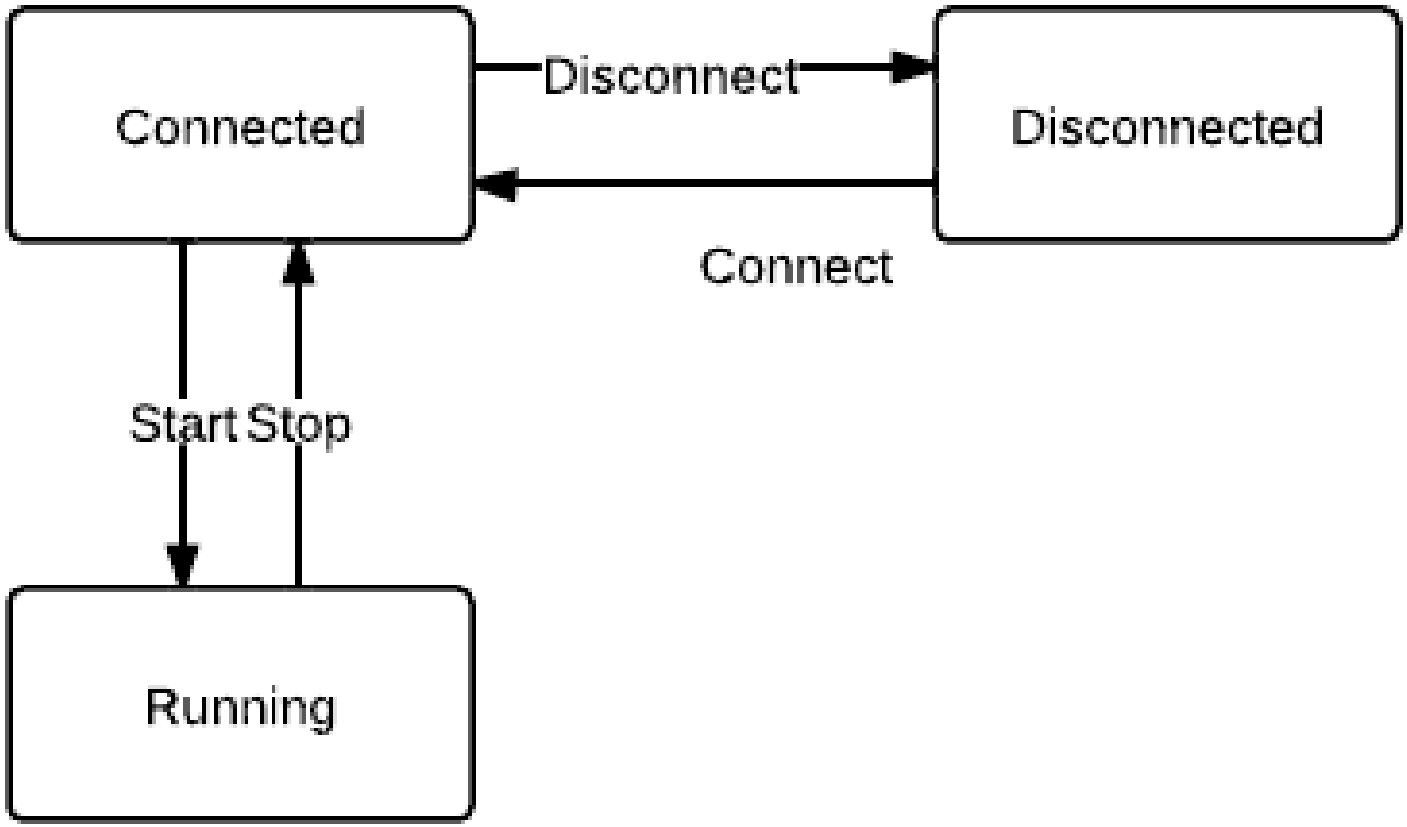
- Testing the new algorithm and fine tuning of various parameters (1 day)
- Documenting and cleaning code, making final presentation and video script (2 days)

Work Division

Work/Critical Tasks	**People Responsible**
Module for Real Time Serial Port Communication with the arm	Aakash, Gopi, Bhargav
Module for Handling Skeletal Data and Performing Calculations	Aayush, Gopi, Bhargav
Design and Coding of the Two-Circle Algorithm	Gopi, Aayush, Aakash
Graphical User Interface (GUI) for the Application	Bhargav, Aayush, Gopi
Documentation	All

Project Completion Date : 19th April 2012

GUI FSM


Connected
Disconnect
Disconnected
Connect
Start Stop
Running

Innovation and Challenges

- Developing an API for robotic arm control
- Designing an intuitive control mechanism for the user
- Developing an efficient algorithm for finding the angles of the robotic arm
- Implementing some averaging features to remove deviations in the kinect data

Problems and Solutions(1/3)

- API/Hardware manual for robotic arm not available
  - decompiled the robotic arm control GUI, read and understood the code and developed an API/library which future projects can use
- Kinect cannot capture finger and wrist movements
  - use the left arm to intuitively control the wrist and pinch motors

Problems and Solutions(2/3)

- Kinect data contains many disturbances and leads to the arm deviating unnecessarily
  - used the median of the last few values to remove extremes. Number of samples used shouldn't be too less or too high, 10 seemed a good value after testing

Problems and Solutions(3/3)

- Human arm and the robotic arm are structurally very different
  - initially performed a direct angle-to-angle mapping from the human arm to the robotic arm, but it was not very intuitive

Axis Movement
Axis 1: Base rotation
Axis 2: Shoulder rotation*
Axis 3: Elbow rotation
Axis 4: Wrist pitch
Axis 5: Wrist roll

Figure: Direct Mapping of Angles

  - developed an algorithm for mapping human arm positions to the positions of the robotic arm (most difficult part)

Algorithm - Problem

We will basically aim to keep the hand of the robotic arm in the same 3D position as the human hand relative to the body. The angles at each axis of the robotic arm are automatically generated by the algorithm

- the angles found by the algorithm should continuously vary with the position of the human hand, to ensure that the movement of the robotic arm is smooth
- the algorithm should find a solution whenever it exists

Algorithm - Solution

- Rotate the base angle so that the robotic arm lies in the same vertical plane as the point we intend to reach
- In this plane the arm looks as shown in the figure:

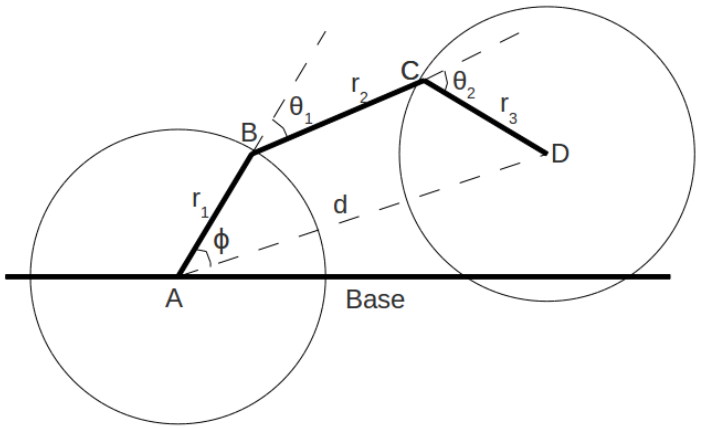


Figure: Two Circle Algorithm

Algorithm - Solution

- We should find ϕ, θ_1, θ_2 under the constraints: $Ad = d, AB = r_1, BC = r_2, CD = r_3, -\frac{\pi}{2} \leq \phi, \theta_1, \theta_2 \leq \frac{\pi}{2}$
- Observe that minimum value of BC occurs when $\theta_1 = \theta_2 = 0$ and maximum value of BC occurs when $\theta_1 = \theta_2 = \frac{\pi}{2}$(when BC is the direct common tangent for the two circles)
- So there will always exist a solution where $\theta_1 = \theta_2 = \theta$ and $0 \leq \theta \leq \frac{\pi}{2}$
- Moreover BC will be a monotonic and continuous function of θ, so we can do a binary search for the solution where $BC = r_2$
- If $r_2 > max\ BC$ then $\theta = \frac{\pi}{2}$ and if $r_2 < min\ BC$ then $\theta = 0$

Modular Testing

- Testing the API/library for real time serial port communication with the arm.
- Microsoft Visual Studio Professional 2010
- Kinect for Windows SDK v1
- .NET Framework 4.0 or higher
- FT232 USB to Serial Converter (provided with the source)
- Testing the skeletal data obtained using Kinect.
- Testing the calculated angles and lengths required for the Two Circle Algorithm.
- Testing the output of the Two Circle Algorithm.

Combined Testing

- Testing the intuitiveness of the control by asking subjects to try to pick and place objects using the arm.
- Testing the GUI for application, which is controls and connection various parameters.
- Testing the effect of modifying various parameters of the algorithm and selecting the ones that give the best results.

Performance Metrics

- The servo motors have response time of somewhere between 0.5 - 1.0 s. This is leads to a delay in the response produced and takes some time to get used to.
- We asked 10 random subjects to try to pick and place a block. The average time taken by them to complete the task was recorder. This is a measure of the intuitiveness of the algorithm. It was found to be around 40 - 60 seconds.

Code Reusability

Modular Design:

- modules independent of each other, hence can be changed without changing others
- developed an API/library for robotic arm control which future projects can use
- the algorithm for finding the arm angles also has wrappers which can be reused

Future Directions

- A more sophisticated algorithm using quaternions and some mechanism for stabilising against disturbances could be the next direction of the project
- The arm can be remotely deployed by communicating through ZigBee and mounting a camera near the arm
- An arm with a faster response time can be reduced to make the control much more intuitive