

CMUCAM3 Turret with Custom GUI, Joystick Interface, and Image Processing Library

Phillip Hoang

California State University, San Bernardino

1. Robot

The goal of this project was to create a platform for image processing and computer vision. Many robots that use computer vision consist of a camera with the ability to pan and tilt. Therefore, a CMUCAM3 Turret was purchased from Acroname with two servos for pan and tilt. The turret was assembled according to the assembly instructions. It is connected through the serial port to a computer, and a USB joystick was added as a user interface. Firmware software for the CMUCAM3 camera board can be developed and loaded onto the camera board. In this project, the CMUCAM2 firmware was loaded onto the camera board to simulate CMUCAM2 serial interface commands. A windows GUI (Graphical User Interface) was developed using OpenGL and GLUT (OpenGL Utility Toolkit). The windows GUI includes features for image processing, serial interface, and joystick input. The image processing includes: sobel edge detection, gaussian blur, sharpening, and sinc interpolation for upsampling. The whole platform was developed from scratch and modularized so that it would be easy for future projects to be developed in C++ (since there is not one readily available).

2. Program

The program was developed using C++ and tested using 1) the g++ compiler under Cygwin and 2) Microsoft Visual Studio C++ compiler. The GUI part of the code uses OpenGL and GLUT so that it is portable to non-Windows operating systems as well. The serial interface is specific to windows, but it is much simpler on a UNIX machine (i.e. serial ports are treated as files on UNIX). The joystick interface uses the GLUT library to enable joystick input. Image processing features were developed using information found online. Websites that I used as references are: http://en.wikipedia.org/wiki/Sobel_operator, <http://www.gamedev.net/reference/programming/features/imageproc/page2.asp>, and <http://slacy.com/upsample/>. The program is a main loop that processes user commands, sends commands to the serial port, grabs images, and process the images according to the user selection. To compile on Cygwin, type "make robot" or in the msvs6 directory open cmucam gui.dsw in Microsoft Visual Studio C++ 6.0. To run, double click robot.exe.

1. Problems

There were several challenges in developing this project: GUI, serial interface, interpreting CMU camera commands, and implementing image processing capabilities. The GUI was challenging because it took some time to design an interface that would be portable to other platforms and provided flexibility for development. The serial interface was the most challenging because it took awhile to find documentation on how to access the serial port on windows using C++. It also took some time to configure, read, and write to the serial port. To help with debugging the serial port interface, a serial port monitor was used to monitor traffic on the serial port. Serial data bytes could be compared directly with software that was supplied for the CMUCAM. After the serial interface problem was solved, it took some time to extract the image data from the serial port data efficiently. The old CMUCAM software downloaded camera images byte by byte and stored it to a file, which took a few seconds to receive and display the image. My serial interface calculates exactly the amount of bytes to be read and stores all of the data to memory. Lastly, the image processing part of the project took some time to understand the algorithms and how to implement them correctly into the project.