P.E.T - Pi Enabled Tracking Bot

Sandipan Mandal
Indian Institute of Technology Kanpur
mandals@iitk.ac.in

Prakhar Kulshrestra
Indian Institute of Technology Kanpur
prakhark@iitk.ac.in

Amit Gupta
Indian Institute of Technology Kanpur
amigupt@iitk.ac.in

Preetansh Goyal
Indian Institute of Technology Kanpur
preetg@iitk.ac.in

June 24, 2014

Abstract

We have made a robot capable of Surveillance and also with an alternate application in detecting and following a pre specified object such as ball. The detection and recognition has been done using *OpenCV* library. The code was written in C++. And this all processing has been done on *Raspbian OS* on *Raspberry Pi*. For capturing the images we are using *Pi Camera Board*.For Surveillance, we have used video streaming over WiFi through a router from Raspberry Pi to laptop.

The whole code for object detection and following is written in C++ using OpenCV libraries.If the object is recognized as a familiar one based on the "hsv" color values then it is identified.Then, the coordinate of the identified object is determined in the grid of the frame seen by the Pi cam.Now, depending on the $x-cordinate$ of the object the bot moves to appropriately bring the $x-coordinate$ in an optimum range in the middle of the grid.

Introduction

The main aim of our project is to make a mobile, wireless robot capable of following a pre specified object and also can be used as a Surveillance robot. For achieving this task we have chosen Raspberry Pi as our Processing hardware with Raspbian OS. The visual input is taken by Pi Camera Module attached to Raspberry Pi via cable into the **CSI port** of Pi.

Now, for the surveillance part of the bot, we achieved it through streaming the video captured by the Pi cam to user laptop over Wi-Fi.We use a software *Mjpg streamer* to stream the video to a web page with the Pi IP Address. Now, this web page can be accessed by the user remotely through the Wi-Fi hotspot connection. The User's laptop is enabled with the VLC media player to convert and save the streamed video from the web page in .avi video format.

Motivation Behind the Project

The main motivation in pursuing the project was to gain experience over Raspberry Pi system.The surveillance system is a system that is capable of handling security of any facility using robots.The object detection gave us valuable knowledge of using OpenCV libraries.With such features in mind, we started working on this project as this can be advantageous for surveillance and tracking in clean environments, and in automation systems.

Moreover building such a project using Raspberry Pi and Pi Cam is economical too. As though there are such security cams available in market but they are too expensive and our robot is mobile and wireless controlled.

Background Theory

We provide you some technical details about the Hardwares and Softwares used:

Hardware

Raspberry Pi: For processing the whole code, we have used Raspberry Pi as it has 700 MHz processor and 512 MB RAM. It even has two USB ports, HDMI port and one 100mb Ethernet port. It works on *Raspbian OS* based on *Debian* which is Linux OS. It is a small sized single-board computer.
Pi Camera: For efficient image capture we required good camera compatible with our Pi. For this, we used Pi Camera Board which is capable of taking HD videos and stills. It has a 5 MP fixed-focus camera that supports 1080p30, 720p60 and VGA90 videos, and still captures. It can be accessed through MMAL and V4L APIs. It connects to Pi through a ribbon cable at the CSI port.
Atmega32: The high-performance, low-power Atmel 8-bit AVR RISC-based microcontroller combines 32KB of programmable flash memory, 2KB SRAM, 1KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.
Servo motor: A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration.It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Software

OpenCV: For image capturing and image processing we have used OpenCV on Raspberry Pi.OpenCV i.e. Open Source Computer Vision is a library of programming functions that helps in real-time image processing. OpenCV is written in C++ and its primary interface is in C++. The library is cross-platform. It focuses mainly on real-time image processing.
PuTTY: For accessing Pi through command line on windows PC we needed PuTTY. PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection.
Xming: For using remote desktop on Pi Xming was used. It is an `X11 Display Server`. Xming may be used with implementations of Secure Shell (SSH) to securely forward X11 sessions

from other computers.

Implementation Details

Locomotion

Our system's locomotion is controlled through wheels driven by servo motors which are controlled using **atmega32** micro-controller.To drive a servo motor the Atmega provides appropriate **PWM(Pulse Width Modulation)** so,that twin servo combination can move the robot in all four directions.

The Atmega32 micro-controller is coded using software **CodeVisionAVR**. The software provides an interface to code instructions to be given to the atmega in c language and compiles is to a **.hex** file to be understood by the atmega microcontroller. The program then is embedded to the microcontroller from the laptop using a programmer. It simply consists of an ATMega88 or an ATMega8 and a couple of passive components. The programmer uses a firmware-only USB driver, no special USB controller is needed. The atmega is connected to the **Raspberry Pi** computer and the **servo** motors.The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse.It uses a 16 GB SD card as memory and **Raspian** as the operating system.The Raspberry Pi is the main processing unit of our system and coordinates the servos and the atmega for the purpose of locomotion.Our system runs through instructions given by the user to the Raspberry Pi over WiFi using the Router, a process known as **SSH**.Secure Shell (SSH) is a cryptographic network protocol for secure data communication, remote command-line login, remote command execution, and other secure network services between two networked computers. It connects, via a secure channel over an insecure network, a server and a client running SSH server and SSH client programs, respectively. For Communication over WiFi, we first create an access point on the Raspberry Pi which is connected to the router to create a WiFi hotspot. Now, the user remotely connects a laptop with the hotpot to establish SSH connection with the Pi computer using SSH client softwares **Putty** and **Xming** installed on user laptop.An SSH client program is typically used for establishing connections to an SSH daemon accepting remote connections.PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection.On establishing the connection we access the raspberry pi terminal to start **Minicom**.

Image Capture and Processing

As our system moves, we use a **Pi Cam** to capture images continuously.The Raspberry Pi Camera Module is a custom designed add-on for Raspberry Pi. It attaches to Raspberry Pi by way of one of the two small sockets on the board upper surface.The sensor itself has a native resolution of 5 mega pixel, and has a fixed focus lens on-board.We use a software Mjpg streamer to stream the video to a web page with the Pi IP Address. Now, this web page can be accessed by the user remotely through the Wi-Fi hotspot connection. The User's laptop is enabled with the VLC media player to convert and save the streamed video from the web page in .avi video format.This helps in the Surveillance feature of the system.

Now, for the object detection and following part,the image processing part is taken care by a collection of libraries as **OPENCV**.Now, taking inspiration from open source codes using the opencv libraries we do the detection based on ”hsv” color values of the object.

Analysing the data obtained from Pi cam, the object’s coordinates in the frame are determined. The camera attached with the bot captures images at distinct times. Based on the $x - cordinate$ of the object the bot moves accordingly.So, the robot moves until we have the $x - coordinate$ in an appropriate range in the middle of the frame.The path is recovered incrementally pose after pose.

Limitations

The major hurdle in our implementation was due to the processing limitation of the raspberry pi computer, due to which we were unable to do robust on-board image processing.Also, we faced problems as we obtained low fps for streaming videos over Wi-Fi through the raspberry pi to the user computer.

The accuracy that can be attained for processing using OpenCV on Raspberry Pi depends on experimental conditions. So for that the program needs to be trained with many test images and that too in different test conditions.

Future Scope

- Our system can be improved further to be able to detect multi-coloured objects.
- After detection of object, interaction possible through additional appropriate hardware like limbs,etc.
- Processing of streamed video for multiple-object tracking.
- Implementing Visual Odometry, which is technique meant for determining the system’s location based solely on camera perception.

References

- http://www.raspberrypi.org/help/noobs-setup/
- http://www.raspberrypi.org/help/camera-module-setup/
- http://thepihut.com/pages/how-to-install-the-raspberry-pi-camera
- http://robertcastle.com/2014/02/installing-opencv-on-a-raspberry-pi/
- https://www.youtube.com/watch?v=8Dw2ow9uAiQ
- http://stackoverflow.com/questions/22383080/raspberry-pi-unable-to-connect-to-internet-with-wifi-shared-over-ethernet
- $http : //elinux.org/RPi_{R}emote_{A}ccess$
- $http : //sourceforge.net/projects/xming/$
- http://www.electroons.com/blog/2013/02/how-to-recover-the-unallocated-space-on-your-sd-card-due-to-raspberry-pi-os/
- http://sourceforge.net/projects/win32diskimager/
- 0 $http : //docs.opencv.org/trunk/modules/contrib/doc/fa$
- $http : //www.raspberrypi.org/facial - recognition - oper$
- http://thinkrpi.wordpress.com/2013/05/22/opencv-and-camera-board-csi/

- http://www.raspberry-projects.com/pi/pi-operating-systems/raspbian/auto-running-programs

- http://www.stuffaboutcode.com/2013/05/time-lapse-video-with-raspberry-pi.html

- http://www.raspberrypi.org/forums

- http://raspberrypi.stackexchange.com/

- https://www.youtube.com/watch?v=bSeFrPrqZ2A

- http://petrkout.com/electronics/low-latency-0-4-s-video-streaming-from-raspberry-pi-mjpeg-streamer-opencv/

- http://stackoverflow.com/questions/18077652/mjpeg-stream-fails-to-open-in-opencv-2-4

- http://www.raspyfi.com/wi-fi-on-raspberry-pi-a-simple-guide/