

Documentation for the Image Processing Tools.

1 Background

This code is designed to allow teams to use image processing on devices other than the National Instruments CompactRIO or the Drivers Station (e.g. a Raspberry Pi, Laptop). This project was started after the FIRST FRC 2012 Rebound Rumble game. Ideally, teams will be able to use this code, with few modifications on their own, in future game years.

1.1 Present Status (v2.1)

The code is designed to read data in from the Microsoft Kinect using the OpenKinect libfreenect. It can read from both the RGB Camera and the Depth Sensor. Processing functions are provided which use the OpenCV library. The code provides support for transferring images to the Driver Station and for transferring processed data to the CRIO.

1.2 Support

The code should run on any machine running Linux. Ubuntu 12.04 is the primary development platform. Sample CRIO & Driver Station code is provided using Labview Code.

2 Installation

Required Libraries: OpenCV, libfreenect, libturbojpeg, CvBlobsLib

Required: Internet Connection

2.1 Installing libfreenect

Open up the terminal and run the following command:

```
sudo apt-get install libfreenect-dev
```

2.2 Installing libturbojpeg

Open up the terminal and run the following command:

```
sudo apt-get install
```

2.3 Installing OpenCV

- 1) Goto: <http://opencv.willowgarage.com/wiki/InstallGuide>
- 2) Download the latest release for Linux (listed under Overview). Unzip this to a directory.
- 3) Open up the terminal and run the following commands.
 - a) `sudo apt-get install pkg-config libgtk2.0-dev glib gthread`
- 4) goto the directory you unzipped the OpenCV source code to. Make sure you are in the director with the README.

- 5) Run the following commands
 - a) `mkdir release`
 - b) `cd release`
 - c) `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D BUILD_PYTHON_SUPPORT=ON ..`
Note: Make sure you include the "<space>.<space>" in the command
 - d) `make install`
 - e) Go get something to eat. If you are running this on a Raspberry Pi, go get some sleep.

2.4 Installing cvBlobsLib

- 1) Goto: <http://opencv.willowgarage.com/wiki/cvBlobsLib> and download the latest release
- 2) Unzip this to a folder
- 3) Open terminal and go into the folder containing cvBlobsLib. Make sure you see the README file
- 4) Run the following commands
 - a. `sudo make`
 - b. `sudo cp libbolb.a /usr/local/lib`
 - c. `ldconfig`
 - d. `sudo cp *.h /usr/local/include/cvblobs`

2.5 Installing CodeBlocks (recommended)

- 1) Open up terminal and run the following command
 - a. `sudo apt-get install codeblocks`

3 Code Documentation

3.1 Kinect Functions

Provides access to the Kinect Sensor

3.1.1 struct FRC_Kinect

This is the basic datatype which stores all the Kinect information

3.1.2 initialize_kinect

This function initializes the FRC_kinect datatype

3.1.3 freenect_threadfunc_server

This function is initialized as a thread. It provides communication between the dashboard and your processor as well as running the image processing functions.

3.1.4 treenect_threadfunc_local

This function is initialized as a thread. It runs the image processing functions with the dashboard communications features.

3.2 TCP Server Functions

Provides TCP/IP Communication functions

3.2.1 struct FRC_TCP_Server

This is the basic datatype which stores all the TCP Connection information

3.2.2 int start_server(FRC_TCP_Server* server, int port)

Starts listening for TCP connections on the given port. The pointer to server must point to an FRC_TCP_Server handle.

3.2.3 int xmit_tcp_data(FRC_TCP_Server* server, uint8_t* buffer, unsigned long length)

Transmits length bytes of data stored in buffer over the tcp connection server.

3.3 Vision Functions

Functions which process the RGB camera images.

3.3.1 Struct cvVision

This is the basic datatype which stores all the image processing information

3.3.2 void Initialize_RGB_lookup(CvScalar Lower, CvScalar Upper, uint8_t* RGB_Lookup)

3.3.3 int RGB_lookup_initialize(uint8_t** RGB_Lookup)

3.3.4 void RGB_lookup_set(CvScalar Lower, CvScalar Upper, uint8_t* RGB_Lookup)

3.3.5 int RGB_lookup_update(int X_one, int Y_one, int X_two, int Y_two)

3.3.6 int HLS_set(int X_one, int Y_one, int X_two, int Y_two)

3.3.7 rgb_server

This image transmits the image from the Kinect RGB sensor to the dashboard.

3.3.8 rgb_blobs

This function uses the CvBlobs library to find blobs in RGB camera. Image color thresholds are set using min_color and max_color. These can be set using the sample dashboard data. Colors are filtered in HLS colorspace. The resulting RGB image is transmitted to the dashboard.

3.3.9 rgb_blobs_lookup

This function uses the CvBlobs library to find blobs in RGB camera. Image color thresholds use the RGB_Lookup table data. These can be set using the sample dashboard data. Colors are filtered in RGB colorspace. The resulting RGB image is transmitted to the dashboard.

3.3.10 rgb_convex_contour

This function is based primarily on FIRST FRC Team 341 Driving Miss Daisy's released code from the 2012 Rebound Rumble game.

This function detects rectangles in the image. Image color thresholds are set using `min_color` and `max_color`. These can be set using the sample dashboard data. Colors are filtered in HLS colorspace. The resulting RGB image is transmitted to the dashboard.

3.4 Depth Functions

Functions which process the depth sensor data.

3.4.1 `struct kinDepth`

This is the basic datatype which stores all the depth processing information

3.4.2 `depth_xy_distance`

This function is passed a pair of XY coordinates. It displays the depth sensor reading on the screen.

3.4.3 `depth_circle_target`

This function analyzes the depth sensor data and attempts to find spherical objects.

3.4.4 `depth_surface_target`

This function analyzes the depth sensor data to find surfaces.

3.5 Global Variables

Variables used for sharing data amongst all the functions.

3.6 Test & Experimental Functions

Functions used for testing out code.

3.6.1 `Void jpeg_test(void)`

This function reads in a jpeg image from a file and decodes it. It is then available for use for testing out image processing functions.

4 Customizing for your Application

4.1 Main

- 1) Comment out the initialization routines not used.
- 2) In the line beginning
`res = pthread_create`
set the third value as `freenect_threadfunc_local` or `freenect_threadfunc_server` as desired. If you decide to develop your own function and give it a different name, change this to that name.

4.2 `Freenect_threadfunc`

- 1) In `freenect_threadfunc_XXXXXX`
Find `freenect_set_depth_callback` and `freenect_set_video_callback`.
The 2nd value needs to be set to the video or depth processing function you will use.

- 2) If you are not using either the video (RGB) or depth, comment out the lines further down `freenect_start_depth`, `freenect_start_video`, `freenect_stop_depth`, and `freenect_stop_video` as required.

5 Labview Interface

The labview interface is fairly basic at this point but should be expandable.

5.1 Image Display

Displays the image transmitted by the image processing device.

5.2 HLS Values

Entering six the values into HLSHLS box and clickint xmit will update the color threshold values. The first three values hold the lower limit and the second three values hold the upper limit. Currently, lower hue values above the upper limit are not supported.

5.3 Image Interface

5.3.1 Single Left Click

Single Left Click on the image to transmit the image coordinates to the image processor. This is presently only used by 3.4.2 `depth_xy_distance`

5.3.2 Shift Left Click

Shift left click twice in a row on two different locations. This will transmit the two coordinates to the image processor. This is used by 3.3.5 and 3.3.6.

6 Troubleshooting

This chapter is setup for issues which have been found during the course of development.

- 1) Wrong JPEG library version: library is 62, caller expects 80
When: At Runtime
Cause: Unknown
Solution: reinstall libturbojpeg, see 2.2

7 Further Reading

| | |
|--------------------------|---|
| Microsoft Kinect | http://www.microsoft.com/en-us/kinectforwindows/ |
| OpenKinect – libfreenect | http://openkinect.org/wiki/Main_Page |
| OpenCV | http://opencv.org/ |