EE128/ME134 Visual tracking notes v 1.02

November 2020

1 Making aruco tags

Directions on making tags are here:

`www.learnopencv.com/augmented-reality-using-aruco-markers-in-opencv-c-python/`

> Generating ArUco markers in OpenCV
>
> *We can generate these markers very easily using OpenCV. The aruco module in OpenCV has a total of 25 predefined dictionaries of markers. All the markers in a dictionary contain the same number of blocks or bits(4×4, 5×5, 6×6 or 7×7), and each dictionary contains a fixed number of markers(50, 100, 250 or 1000). Below we show how to generate and detect various kinds of aruco markers in both C++ and Python.*

Note that the aruco tag detection must be using the same dictionary type, here for a 6x6 bit tag.

```
dictionary = cv2.aruco.Dictionary_get(cv2.aruco.DICT_6X6_250)
```

2 Reading aruco tags

This repository (available with MIT license) has nice tracking from laptop camera showing center of tag as well as orientation.

```
github.com/tongplw/OpenCV-ArucoDetection/
  blob/master/src/7%20Get%20ID%2C%20tvec%20and%20rvec.py
```

For each visible tag, this code returns

- id tag
- translation vector
- rotation vector

2.1 Camera Calibration

We are not expecting high precision from Tello, thus camera lens calibration should not be necessary. We suggest you measure the distance of the tag from the camera, and use that as a scale factor. Then you can get distance in absolute units.

If after everything else is working, you can use a checkerboard to calibrate for camera/lens errors.

`docs.opencv.org/master/da/d13/tutorial_aruco_calibration.html`

`upload.wikimedia.org/wikipedia/commons/thumb/7/70/Checkerboard_pattern.svg/1200px-Checkerboard_pa`

3 Video streaming from Tello

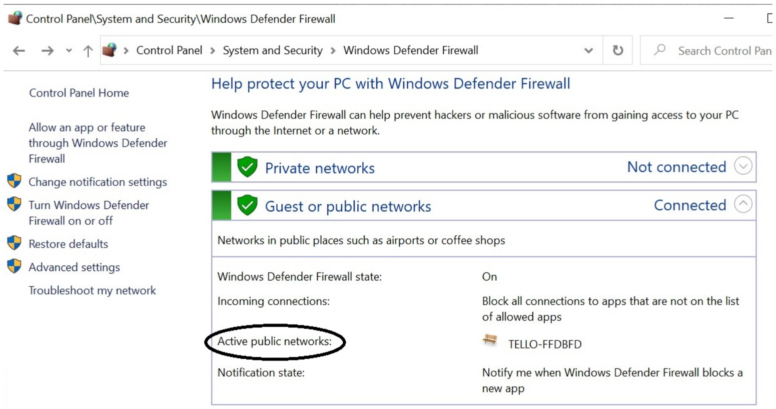
To be written. For initial work, instead use video streaming from another camera, such as laptop camera. Tello streaming adds extra complexity and it is important to get the simpler version working first.

The first thing to try is `tello_opencv_Cam.py` from gist.github.com/f41ardu/b75da46dae383e5c835295600deef604

which has been tested on Windows, Mac, and Ubuntu.

On Windows, you need to change settings on Windows Defender Firewall to allow video streaming.

1. Inbound rules
   Allow UDP access to port 11111 for details see
   www.mathworks.com/help/supportpkg/ryzeio/ug/enabling-video-receive-udp-port.html
2. Public network access
   Make sure python has access to the public network which has the streaming video.
3. Enable Tello streaming
   Make sure to use another window to send "command" and then "streamon" to enable video streaming on this port.
   github.com/ronf-ucb/TelloControl/blob/master/TelloKeyboardCommands.py



Control Panel\System and Security\Windows Defender Firewall
Control Panel > System and Security > Windows Defender Firewall
Search Control Pan
Control Panel Home
Allow an app or feature through Windows Defender Firewall
Change notification settings
Turn Windows Defender Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network

Help protect your PC with Windows Defender Firewall

Windows Defender Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.

Private networks — Not connected

Guest or public networks — Connected

Networks in public places such as airports or coffee shops

Windows Defender Firewall state: On

Incoming connections: Block all connections to apps that are not on the list of allowed apps

Active public networks: TELLO-FFDBFD

Notification state: Notify me when Windows Defender Firewall blocks a new app

Figure 1: Tello uses public network, so make sure python has permission for public network.