Tello

Welcome to the Tello Drone module

This module has two major components.

1. `warehouse.pyc` which connects to the drone, extracts the video feed and issues commands to the drone.
2. `controller.pyc` examines the incoming feed and determines the appropriate action. It issues the action command to `warehouse.pyc`.

Before setting up your system, you need to do a physical environment setup.

1. The workspace has a `bins.pdf` file which has the barcodes for all the bins. Please print this document and stick the barcodes on a wall. The barcodes are numbered. The numbering convention followed is **"Shelf #.Bin #**. For example, `01.03` denotes shelf # 1 and bin # 3. Please stick the barcodes in an ascending order.
2. The bin barcodes needs to be 1 - 1.2 meters above ground level. Also, the vertical distance between two shelves needs to be around 1 meter.

> **Note** : After sticking the bin barcodes, run one mission (after setting up the system) and evaluate the position of the barcodes. You might have to adjust the position of the bin barcodes depending on the elevation of the drone.

3. The `items.pdf` file contains the barcodes for the items. Please stick these barcodes above the bin barcodes. If you want to show multiple items in a bin then please stack the barcodes on top of each other above the bin barcode. You can have any barcodes for items. Just make sure that the barcode is 4.3" long as well as wide.

System Setup:

The entire module is a single virtual `Python` environment. In order to setup the environment,

1. Install OS specific version of Conda.

2. Then create a virtual conda environment by navigating to the module's root directory from the terminal window/command prompt, `cd <module's root directory>`

> **Note**: Please do not type `<module's root directory>` in the window. The root directory varies from system to system. On my system the root directory is in `/home/ericsson/Documents/tello`.

3. Once in root directory, run `conda env create -f environment.yml` in the terminal window/command prompt.
4. This will create the virtual environment. In order to start the virtual environment, run `source activate tello` in the terminal window. On a Windows system, run `activate tello` in the command prompt.
5. To stop a virtual environment, run `source deactivate tello` in the terminal window. On a Windows system, run `deactivate` in the command prompt.
6. Install the QR code scanning library by following instructions from this link.

Troubleshooting

Because of the complexity of `av` library, you might run into issues while creating a virtual environment. If you fail to create an environment then manually create an environment by running the following commands in terminal window.

1. `conda create -n tello python=2`
2. `source activate tello`
    - On a Windows system, run `activate tello` in the command prompt.
3. `conda install -c conda-forge opencv`
4. `conda install -c conda-forge av`

Installing pyzbar

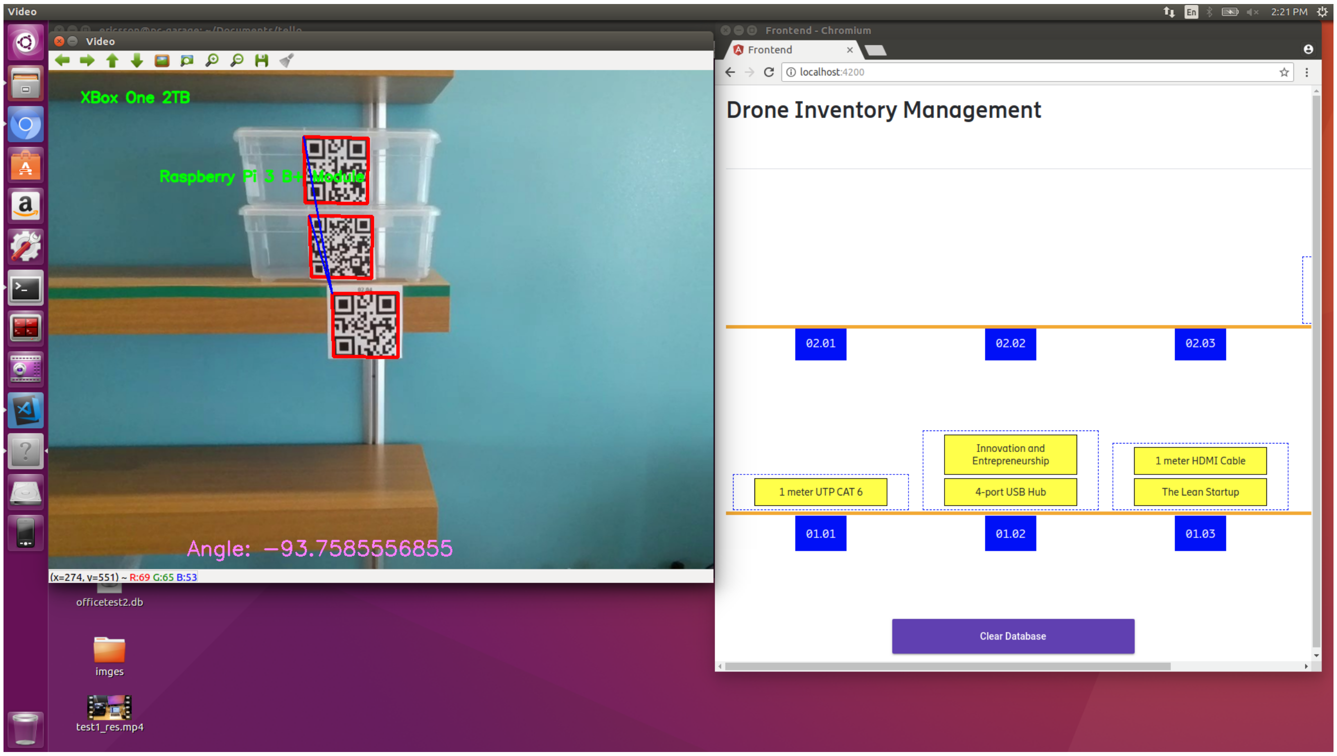
With the virtual environment active, follow the steps on this page to install pyzbar.

Running a mission

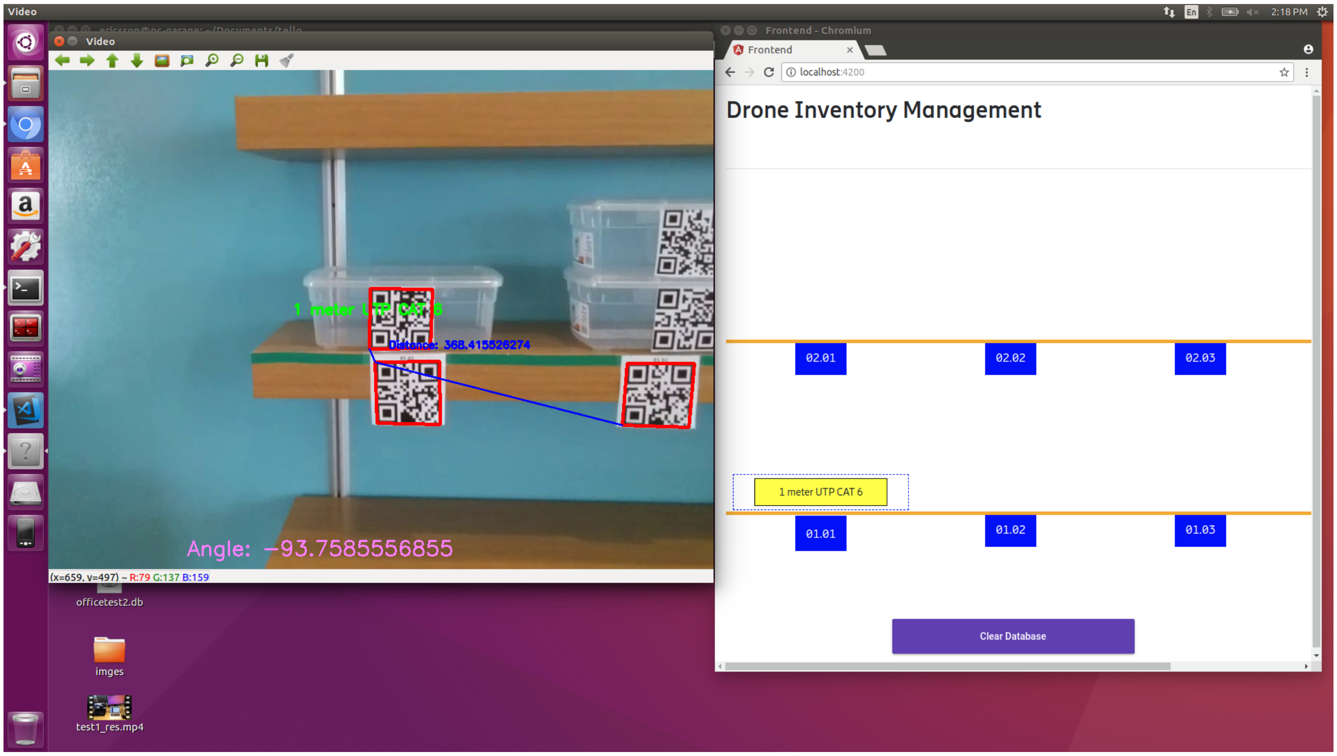
1. Once a virtual environment is created and activated. Open another terminal window/command prompt, navigate to the module's root directory, run `source activate tello` or `activate tello` depending on the OS.
2. Then in one of the window's, run `python controller.pyc`.
3. Then turn on the drone, place it on the ground, approximately 1 meter away from the wall/barcodes.

4. Connect to the drone's hotspoot. The drone hotspot is usually named `TELLO-XXXX`.
5. Once connected to the hotspot, run `python warehouse.pyc`. Then wait for 30 seconds for the drone to boot up and takeoff. The default speed of the drone is 5. However, you can specify a custom speed by running `python warehouse.pyc -s 10`. Make sure that the speed is less than 100. Ideally, keep the speed between 5 and 15.
6. Optional You can have the backend and frontend running to visualize the bins and the items on a webpage.

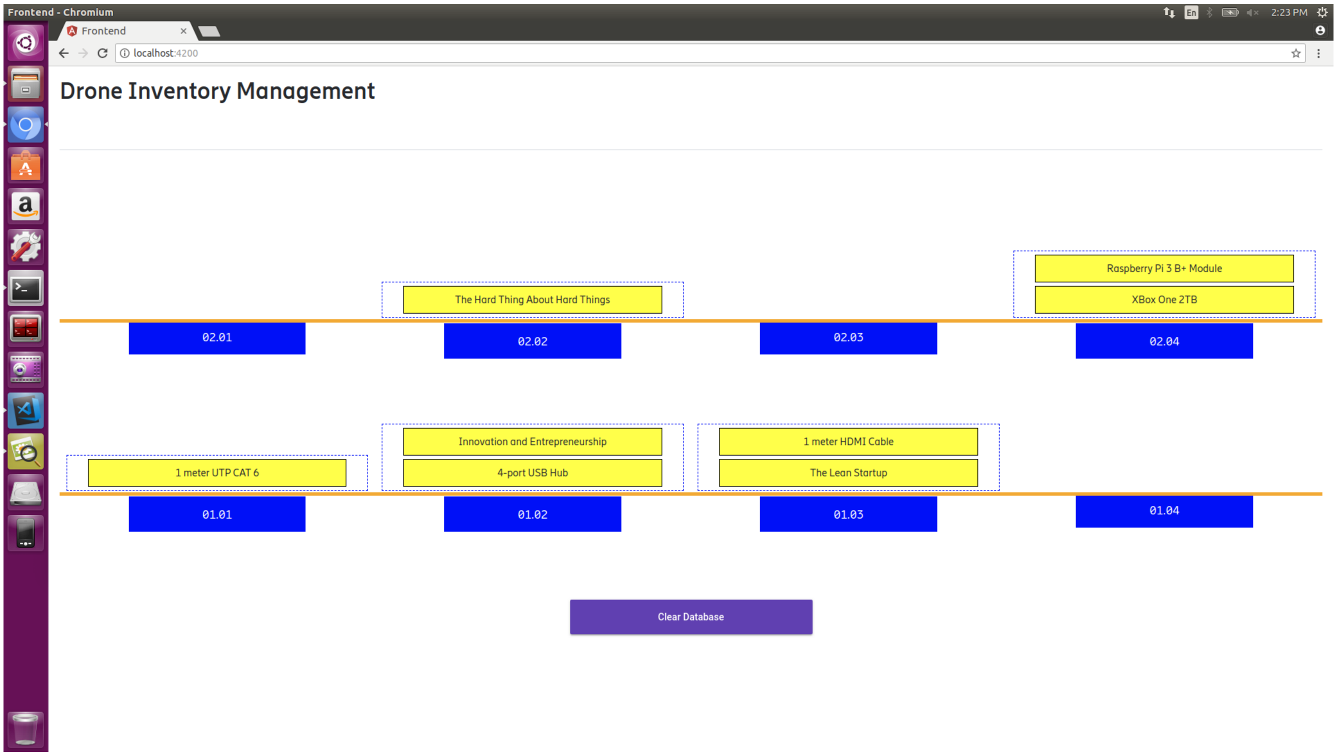
Once in mission, you should see the following in the video window.



If the drone is scanning the barcodes then you should see this. The drone overlays the item barcode information on the video feed. The program draws a blue line between the bin and the items. This blue line signifies the items associated to a specific bin. The association takes place by measuring the distance between the bin and the items and also, the angle between the bin and the items. The angle measured is overlayed at the bottom of the video feed in **pink** text.



The drone also claculates the distance between two bins. This calculated distance is indicated by drawing a blue line between two consecutive bins and the distance between them is overlayed in **blue** text. This distance is used for planning the next action.



At the end of the mission you should see an updated inventory like this,

Running in a distributed environment

In order to run this module in a distributed fashion please follow these steps

1. Clone the repository in the new system.

2. Do the exact same system setup as described above.

3. You will be running `warehouse.pyc` and `controller.pyc` on separaate systems. To do that first update the IP addresses of the two systems in `config.json`.
4. Then run `python controller.pyc` and `python warehouse.pyc` on their respective systems.

> **Note**: Make sure that backend and frontend are installed and running on the system which is running `controller.pyc`.

Failsafe

The drone has a tendency of going haywire at times. In such situations, perform the following maneuvers.

1. If you want to terminate the mission then focus on the video window by clicking on the video window. Then press `ESC` to terminate the mission. You can press `ESC` anytime to terminate the mission.
2. To switch to a manual control,
   - Make sure that you are focusing on the video window. Press `SPACE` to switch to a manual mode. Pressing `SPACE`again will put the drone back in autonomous mode.
   - Once in manual mode, press `w` to move forward, `a` to move left, `s` to move back, `d` to move right, `q` to rotate counter clockwise, `e` to rotate clockwise, `c` to move up and `v` to move down.