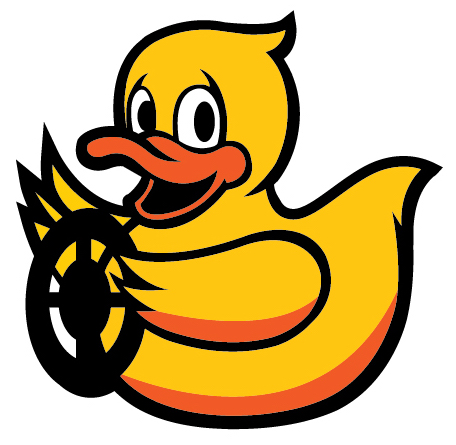
Setup Step 2.2.2   
**Camera Calibration**

[1. Setup](#_hw9z5kr7fwyc)

[2. Intrinsic Calibration](#_rotqnrgtayeg)

[3. Extrinsic Calibration](#_17kexsxeypis)

# 1. Setup

This document assumes that the car setup is done up to [Step 2.1](https://docs.google.com/document/d/1FB25mF8703TtEBUfNR6s8NXYc8_22lCU_7gAfQ_Rw_Y/edit?usp=sharing).

We will use a checkerboard to do both intrinsic and extrinsic calibrations.

Print [calibration.pdf](https://drive.google.com/open?id=0B1iMTx9IcQVwN2pEcXE4RUF1VVk) with US letter paper. When you print it, don’t shrink or extend when print the pdf file. If it is properly printed, the side length of a black square should be 3.1 cm (about 1.22 in).

Put the two pages side by side on a planar surface as show below.



A table will be a good place. Fix them on the table top so that they will not move during calibration. Make sure that coordinate frames are conforming.

Install the camera\_calibration node (<http://wiki.ros.org/camera_calibration>) on your external machine (e.g. laptop) if not installed

laptop $ sudo apt-get install ros-indigo-camera-calibration

# 2. Intrinsic Calibration

Update your duckiebot:

duckiebot $ cd duckietown

duckiebot $ git checkout master

duckiebot $ git pull

duckiebot $ make build-parallel

On your laptop:

laptop $ cd ~/duckietown

laptop $ make

laptop $ roslaunch duckietown intrinsic\_calibration.launch veh:=${VEHICLE\_NAME} raw:=true

*Note: do not run the above using byobu* *, because the calibrator needs to access the screen.*

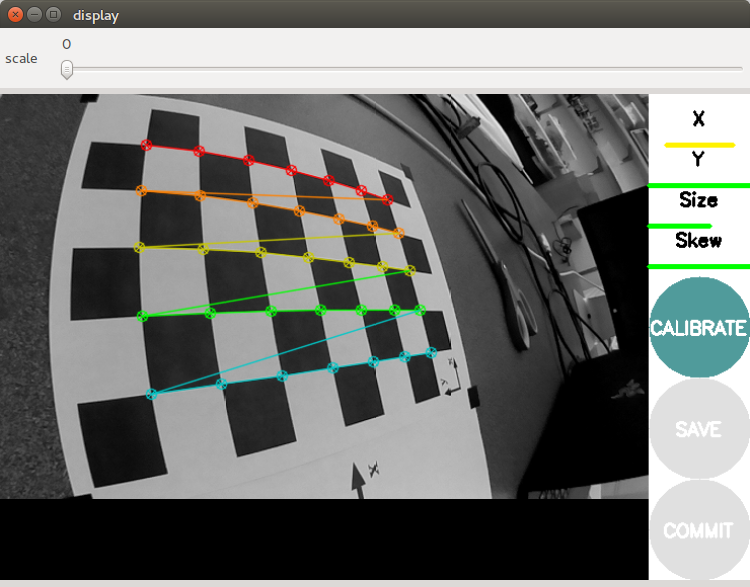
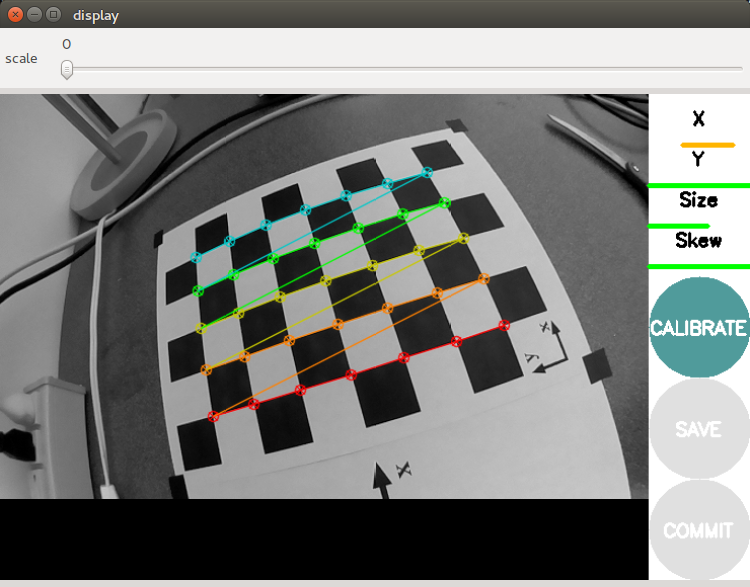
A window will pop up. Grab your Duckiebot and point the camera towards the checkerboard (or move the checkerboard in front of the duckiebot).

Move it around the checkerboard. If the camera sees all corners of the checkerboard, it automatically collects data. For good calibration, we need to capture diverse views.

Zzxxx

As you move the car, you will see four bars on the upper right side increase. Each bar shows the observed range of the checkerboard in the camera’s field of view.

* X bar: the observed horizontal range (left - right)
* Y bar: the observed vertical range (top - bottom)
* Size bar: the observed range in the checkerboard size (forward - backward from the camera direction)
* Skew bar: the relative tilt between the checkerboard and the camera direction



Once you collected enough images, all bars on the upper right side will be green and the ‘CALIBRATE’ button will be enabled. Press the ‘CALIBRATE’ button and it will do calibration. Depending on the number of images collected, it may take more or less a minute. During the calibration, the window might be greyed out; wait. After the calibration, it will show rectified video (i.e. undistorted video).

If you are satisfied with the current calibration, press the **‘COMMIT’** button. It will automatically save to calibration file on your car (**not on your external machine**). The location of the file is: ~/duckietown/catkin\_ws/src/duckietown/config/baseline/calibration/camera\_intrinsic/${VEHICLE\_NAME}.yaml

Now let’s push the${VEHICLE\_NAME}.yaml file to git repository. Connect to your vehicle and push ${VEHICLE\_NAME}.yaml to git, i.e. do this

laptop $ ssh ubuntu@${VEHICLE\_NAME}.local

duckiebot $ cd ~/duckietown

duckiebot $ git pull

vehicle $ git add ~/duckietown/catkin\_ws/src/duckietown/config/baseline/calibration/camera\_intrinsic/${VEHICLE\_NAME}.yaml

duckiebot $ git commit -m "add intrinsic calibration file of ${VEHICLE\_NAME}"

duckiebot $ git push

# 3. E

**Temporary step for Beta version:**

The ground\_projection node is still under development. It is developed in C++, and thus if there is change in the source code, you should re-compile the node. To compile catkin package, run catkin\_make in the catkin\_ws directory as:

$ cd ~/duckietown/catkin\_ws

$ catkin\_make

To run it, press Ctrl+C in the terminal launched intrinsic\_calibration.launch to exit the intrinsic calibration. Then launch camera on your external machine (laptop):

**laptop** $ roslaunch duckietown camera.launch raw:=1 veh:=${VEHICLE\_NAME}

And in another terminal, run ground\_projection node with

**laptop** $ roslaunch ground\_projection ground\_projection.launch veh:=${VEHICLE\_NAME} **local:=1**

You will see new ros topics:

v

$ rostopic list

/<VEHICLE\_NAME>/camera\_node/camera\_info

/<VEHICLE\_NAME>/camera\_node/image/compressed

/<VEHICLE\_NAME>/camera\_node/image/raw

/<VEHICLE\_NAME>/ground\_projection/lineseglist\_out

/<VEHICLE\_NAME>/line\_detector\_node/segment\_list

/rosout

/rosout\_agg

The ground\_projection node has two services. They are not used during operation. They just provide a command line interface to trigger the extrinsic calibration (and for debugging).

laptop $ rosservice list

...

/**veh\_name**/ground\_projection/estimate\_homography

/**veh\_name**/ground\_projection/get\_ground\_coordinate

...

To do extrinsic calibration, put your car as shown in the figure below

Note that the axis of the wheels is aligned with the y-axis.

IMPORTANT: put a uniform white wall behind the checkered board to block off any clutter.

The next step is to estimate a homography. Execute the following command:

laptop $ rosservice call /**${VEHICLE\_NAME}**/ground\_projection/estimate\_homography

Once it estimates a homography, it automatically saves the estimated homography to **${VEHICLE\_NAME}**.yaml file. The location of the yaml file should be **on your laptop** at

~/duckietown/catkin\_ws/src/duckietown/config/baseline/calibration/camera\_extrinsic/${**VEHICLE\_NAME**}.yaml

Note that since we launch the ground\_projection\_node with local:=1, the node is running on your laptop. And the resulting **${VEHICLE\_NAME}**.yaml file will be on your laptop too. Don’t forget to get **${VEHICLE\_NAME}**.yaml from your laptop to your vehicle since during operation we will run the ground projection node on the vehicle. You can by commit and push on laptop and pull on the vehicle.

To make sure if the estimated homography is correct, run the [Testing in Part3 (method 3)](https://docs.google.com/document/d/1UgCuAcE0WJKja-y0WTXR8pFo9f3srQ4TNtAiX8Nhp08/edit#heading=h.6rbv64wwforq) in the [Testing Plan](https://docs.google.com/document/d/1UgCuAcE0WJKja-y0WTXR8pFo9f3srQ4TNtAiX8Nhp08/edit#heading=h.nslreu8cfwor) document. <https://docs.google.com/document/d/1UgCuAcE0WJKja-y0WTXR8pFo9f3srQ4TNtAiX8Nhp08/edit#heading=h.6rbv64wwforq>