Image Tagging

Sloth

Setup

In order to tag images, we use a piece of software called Sloth. To install it from the robosub debian repository, run the following command:

```
sudo aptitude install sloth
```

Using Sloth

To use sloth and start tagging images, you can run the following command.

```
sloth -c /path/to/vision_dev/sloth/robosub_config.py
/path/to/annotation/file
```

-c Is a flag to give the path to a configuration file. This file is provided in the vision_dev/sloth directory. The path you provide should point to this file.

The last argument is a path to an annotation file. This is most likely named something like `labels.json`. You will need to provide this path in order to tag images.

Should you find this tedious, there is a script in the sloth directory of the vision_dev repository which performs some of this for you. It can be used as follows:

```
./robosub_sloth.sh /path/to/annotation/file
```

This removes the need to add the -c flag repeatedly. This script can also be symlinked to without issues.

Sloth Keybindings

The full list of keybindings used in sloth can be found in the robosub_config.py file near the bottom. A shorthand list is provided below

Default keybindings

- **Space**   Mark image as labeled/confirmed and go to next

- **Backspace**   Next image/frame
- **PgDown**   Previous image/frame
- **PgUp**   Previous image/frame
- **Tab**   Select next annotation
- **Shift+Tab**   Select previous annotation
- **Ctrl+f**   Fit current image/frame into window
- **Del**   Delete selected annotations
- **Esc**   Exit insert mode
- **Shift+l**   Mark current image as labeled
- **Shift+c**   Mark all annotations in image as confirmed

Robosub Specific

- **F5**   Toggle the visibility of label names on annotation boxes
- **Ctrl+Shift+Del**   Delete all annotations from the current image and mark it as unlabeled
- **c**   Copy all annotations from the previous image to this one
- **n**   Mark image as labeled/confirmed and copy annotations to next image. (Equivalent to **Space** then **c**)

Annotations

- **s**   Start Gate Post
- **p**   Path Marker
- **t**   Torpedo Target
- **w**   Roulette Wheel
- **1**   Die 1
- **2**   Die 2
- **3**   Die 3
- **4**   Die 4
- **5**   Die 5
- **6**   Die 6

Sloth Mouse Controls

While creating annotations, the following are useful mouse controls.

- **Right Click and Drag**

  Resize an annotation. Resizing is based upon the quadrant of the annotation clicked on.

- **Ctrl+Left Click**

  Select multiple annotations at once.

Data Management

First check if your default python is of version 2.7, and that your pip is for 2.7. To do so run

```
python -V
pip -V
```

The output should be `python 2.7.x`, the version must not be 3.5. For pip it will be `pip x.y.z from /usr/lib/python2.7/dist-package (python2.7)`. If it says that you have pip for python3.5 then you need to purge all versions of pip, and reinstall each.

If you do not have pip installed, run

```
sudo apt-get install python-pip
```

After pip is installed properly, you need to install `setup_tools` and `pysftp` library for python, to do so run

```
sudo pip install --upgrade pip
sudo pip install --upgrade pip setuptools
sudo pip install pysftp
```

Getting, validating, and returning labeling data is handled through the `rslabel` utility program. To install it, run

```
sudo pip install rslabel --extra-index-url
http://robosub.eecs.wsu.edu/python_repo/ --trusted-host robosub.eecs.wsu.edu
```

There are a number of commands to be used with `rslabel`, including `show`, `get`, `return`, `upload`, and `collect`.

Command	Description
`rslabel show`	Provides information about the number of datasets labeled, number of images properly validated, and counts any labeling sessions in progress.
`rslabel get`	Grabs an image dataset for labeling and places it in your current directory. The `--validation` flag can be supplied to get labeled data to validate.
`rslabel return [JSON]`	Returns a dataset to the server. If the data has not been completely labeled or validated, it will be returned for someone else to complete in the future.
`rslabel upload [ROSBAG]`	Takes a ROS bag file and break the images out into datasets for labeling. It will then upload the files to the server for labeling.
`rslabel collect`	Collects all of the labeled and validated images into a single dataset for use with object detection training.