

---

## pykitti



This package provides a minimal set of tools for working with the KITTI dataset [1] in Python. So far only the raw datasets and odometry benchmark datasets are supported, but we're working on adding support for the others. We welcome contributions from the community.

### Installation

#### Using pip

You can install pykitti via pip using

```
1 pip install pykitti
```

#### From source

To install the package from source, simply clone or download the repository to your machine

```
1 git clone https://github.com/utiasSTARS/pykitti.git
```

and run the provided setup tool

```
1 cd pykitti
2 python setup.py install
```

### Assumptions

This package assumes that you have also downloaded the calibration data associated with the sequences you want to work on (these are separate files from the sequences themselves), and that the directory structure is unchanged from the original structure laid out in the KITTI zip files.

---

## Notation

Homogeneous coordinate transformations are provided as 4x4 `numpy.array` objects and are denoted as `T_destinationFrame_originFrame`.

Pinhole camera intrinsics for camera `N` are provided as 3x3 `numpy.array` objects and are denoted as `K_camN`. Stereo pair baselines are given in meters as `b_gray` for the monochrome stereo pair (`cam0` and `cam1`), and `b_rgb` for the color stereo pair (`cam2` and `cam3`).

## Example

More detailed examples can be found in the `demos` directory, but the general idea is to specify what dataset you want to load, then access the parts you need and do something with them.

Camera and velodyne data are available via generators for easy sequential access (e.g., for visual odometry), and by indexed getter methods for random access (e.g., for deep learning). Images are loaded as `PIL.Image` objects using Pillow.

```
1 import pykitti
2
3 basedir = '/your/dataset/dir'
4 date = '2011_09_26'
5 drive = '0019'
6
7 # The 'frames' argument is optional - default: None, which loads the
  # whole dataset.
8 # Calibration, timestamps, and IMU data are read automatically.
9 # Camera and velodyne data are available via properties that create
  # generators
10 # when accessed, or through getter methods that provide random access.
11 data = pykitti.raw(basedir, date, drive, frames=range(0, 50, 5))
12
13 # dataset.calib:           Calibration data are accessible as a named
  # tuple
14 # dataset.timestamps:      Timestamps are parsed into a list of datetime
  # objects
15 # dataset.oxts:            List of OXTS packets and 6-dof poses as named
  # tuples
16 # dataset.camN:           Returns a generator that loads individual
  # images from camera N
17 # dataset.get_camN(idx):   Returns the image from camera N at idx
18 # dataset.gray:           Returns a generator that loads monochrome
  # stereo pairs (cam0, cam1)
19 # dataset.get_gray(idx):   Returns the monochrome stereo pair at idx
20 # dataset.rgb:            Returns a generator that loads RGB stereo
  # pairs (cam2, cam3)
21 # dataset.get_rgb(idx):    Returns the RGB stereo pair at idx
```

---

```
22 # dataset.velo:          Returns a generator that loads velodyne scans
    as [x,y,z,reflectance]
23 # dataset.get_velo(idx): Returns the velodyne scan at idx
24
25 point_velo = np.array([0,0,0,1])
26 point_cam0 = data.calib.T_cam0_velo.dot(point_velo)
27
28 point_imu = np.array([0,0,0,1])
29 point_w = [o.T_w_imu.dot(point_imu) for o in data.oxts]
30
31 for cam0_image in data.cam0:
32     # do something
33     pass
34
35 cam2_image, cam3_image = data.get_rgb(3)
```

## OpenCV

PIL Image data can be converted to an OpenCV-friendly format using numpy and `cv2.cvtColor`:

```
1 img_np = np.array(img)
2 img_cv2 = cv2.cvtColor(img_np, cv2.COLOR_RGB2BGR)
```

Note: This package does not actually require that OpenCV be installed on your system, except to run `demo_raw_cv2.py`.

## References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. J. Robot. Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013. <http://www.cvlibs.net/datasets/kitti/> ‘