pyransac

Release 1.1.3

Adam Morrissett

Jul 04, 2020

CONTENTS

`pyransac` is a general-purpose random sample consensus (RANSAC) framework written in Python. You can use it to remove outliers from your data sets given a data model to which you expect your data to fit. For convenience, some data models (such as a 2D straight line) are already provided. However, you are free to define your own data models.

CHAPTER

ONE

INSTALLATION

You can install `pyransac` using `pip` with the following command:

```
$ python3 -m pip install pyransac
```

CHAPTER

TWO

GETTING STARTED

After installing `pyransac`, all you need to do is create a data model definition (or use one of the built in models) and specify the RANSAC algorithm parameters. From there you can call the `find_inliers` function and pass in your data, model, and parameters. The function will return a list of your inliers.

CHAPTER

THREE

EXAMPLE PROGRAM

The following is a simple example of filtering data against a 2D line model.

```
import pyransac
from pyransac import line2d

# Create data
inliers = [line2d.Point2D(x, x) for x in range(0, 10)]
outliers = [line2d.Point2D(x ** 2, x + 10) for x in range(0, 5)]
data = inliers + outliers

# Specify our RANSAC parameters
params = pyransac.RansacParams(samples=2,
                               iterations=10,
                               confidence=0.999,
                               threshold=1)

# Create our model object
model = line2d.Line2D()

# Get the inliers
inliers = pyransac.find_inliers(points=data,
                                model=model,
                                params=params)

# Compare the data sets
print(data)
print(inliers)
```

CHAPTER

FOUR

CUSTOM DATA MODELS

All data models are derived from the `Model` base class. This class defines an interface consisting of two functions: `make_model` and `calc_error`. The `make_model` function generates a data model from a set of data points. The `calc_error` function calculates the error between a data point and the generated model. See the `Model` *reference* for more information.

You can define custom data models by extending the `Model` class. `pyransac` provides the following built-in data models:

- *Line2D* — a 2-dimensional line model

CHAPTER

FIVE

TABLE OF CONTENTS

5.1 Reference

5.1.1 RANSAC

class pyransac.ransac.**RansacParams**(*samples: int*, *iterations: int*, *confidence: float*, *threshold: float*)

Random sample consensus (RANSAC) function parameters.

This class contains the parameters for the RANSAC algorithm.

confidence: float
The RANSAC confidence value (0 <= confidence <= 1).

iterations: int
Maximum number iterations to complete.

samples: int
The number of random samples to take per iteration.

threshold: float
The error threshold to consider a point an inlier

pyransac.ransac.**find_inliers**(*points: List*, *model:* pyransac.base.Model, *params:* pyransac.ransac.RansacParams)

Find the inliers from a data set.

Finds the inliers from a given data set given a model and an error function.

Parameters

- **points** – data points to evaluate
- **model** – type of model to which the data should adhere
- **params** – parameters for the RANSAC algorithm

Returns inliers

5.1.2 Data Models

class pyransac.base.**Model**
ABC class for data models.

Derivative classes should extend this class and implement its interface.

abstract calc_error(*point*) → float
Calculates error between data point and model.

Parameters **point** – data point to test against

abstract make_model(*points: List*) → None
Makes a model from given data points.

Parameters **points** – list of data points with which to make model

class pyransac.line2d.**Line2D**(*slope=None*, *y_int=None*, *x_int=None*)
Model for a 2-dimensional line.

calc_error(*point:* pyransac.line2d.Point2D) → float
Calculate error between data point and 2D model.

Parameters **point** – data point to calculate error with

Returns calculated error

make_model(*points: List[*pyransac.line2d.Point2D*]*) → None
Makes equation for 2D line given two data points.

Model parameters are stored internally.

Parameters **points** – list of data points to make model (length must be 2)

Returns None

property slope
Gets the slope of the model.

Returns slope of line (None if model not made).

property x_int
Gets the x intercept of the model.

Returns x intercept of line (None if model not made).

property y_int
Gets the y intercept of the model.

Returns y intercept of line (None if model not made).

5.1.3 Helpers

class pyransac.line2d.**Point2D**(*x: float*, *y: float*)
2-dimensional point class.

This is a simple class to contain Cartesian coordinates of 2D point.

x: **float**
x coordinate of point.

y: **float**
y coordinate of point

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

INDEX