
lic

Release 0.4.2

Jul 06, 2020

Contents

1	Quick start	3
2	What is line integral convolution?	5
3	Documentation	7
3.1	Installation	7
3.2	Usage	8
3.3	API	8
3.4	Command line tool	8
4	Project links	11
5	External links	13

This package provides line integral convolution (lic) algorithms to Python.

There are functions which can be imported and are highly configurable for the power user. Moreover, there is a command line tool to generate lic images without having to code a single line.

Table of Contents

- : *Line Integral Convolution for numpy Arrays*
 - *Quick start*
 - *What is line integral convolution?*
 - *Documentation*
 - *Project links*
 - *External links*

CHAPTER 1

Quick start

Install by typing

```
pip install lic
```

Run the command line tool by

```
lic data_x.npy data_y.npy -v -l 30 -c
```

Or use it in your program:

```
import lic
import matplotlib.pyplot as plt

# ... get x and y arrays from somewhere ...

lic_result = lic.lic(x, y, length=30)

plt.imshow(lic_result, origin='lower', cmap='gray')
plt.show()
```

What is line integral convolution?

Line integral convolution is a way of visualizing complete vector fields in an intuitive way.

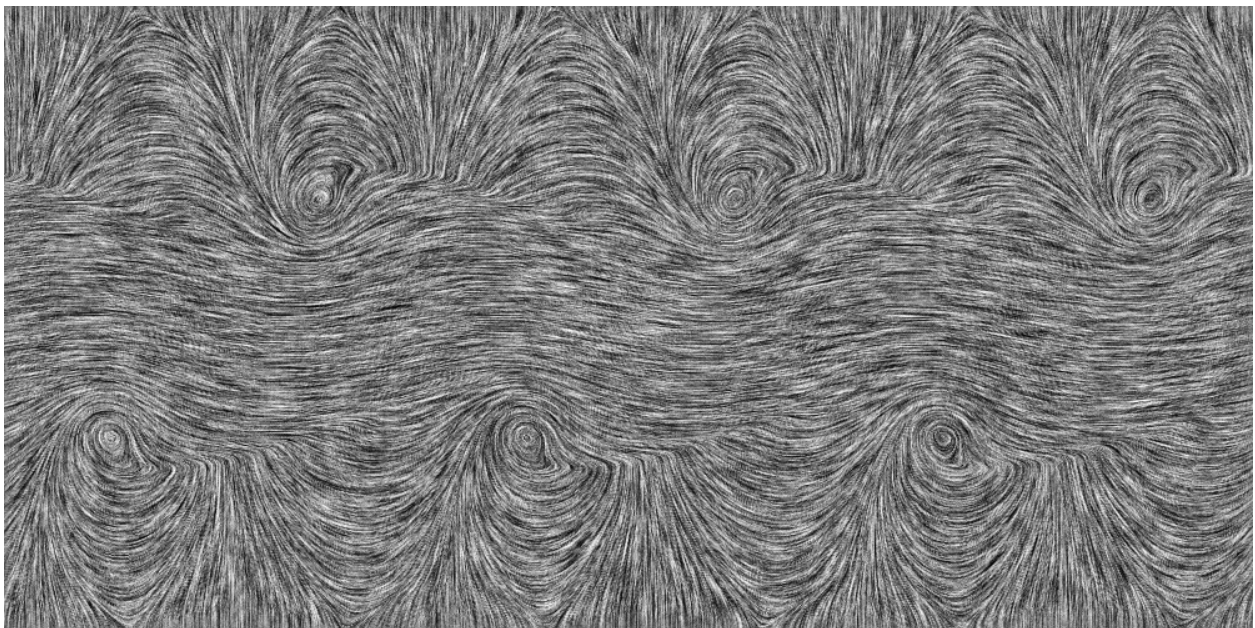


Fig. 1: lic image of the Kelvin-Helmholtz instability

A lic image is generated by “smearing out” a random noise pattern along the flow lines of a vector field. As a result, it shows the entire flow field including every detail, while the common visualizations using arrows or discrete lines will always lose fine details.

The disadvantage of (static) lic images is that the direction of the flow is not displayed. To overcome this problem, one could either overlay an array of arrows or generate a “moving” lic image. Such a moving image is generated by calculating several lic images with shifted “smearing” kernels. When animated, this gives the impression of a flow.

Fig. 2: animated lic image of the Kelvin-Helmholtz instability

3.1 Installation

The installation is straight forward. You can install the package via `pip`, `pipenv`, `poetry` and alike or by downloading the source from the gitlab repository.

3.1.1 From pypi.org (recommended)

Install by typing

```
pip install lic
```

or

```
pip install --user lic
```

if you do not have root access.

Please check the documentations for `pipenv`, and `poetry` for information on how to install packages with these tools.

Once the package is successfully installed, you can *import it* into your program or use the *command line tool*.

3.1.2 From gitlab.com

To get the latest features or contribute to the development, you can clone the whole project using `git`:

```
git clone https://gitlab.com/szs/lic.git
```

Now you can, for instance, copy `lic.py` over to your project and *import it directly* or use it as a *command line tool*.

3.2 Usage

Once the package is *installed* where Python can find it, you can use the function `lic` directly.

```
import lic
import matplotlib.pyplot as plt

# ... get x and y arrays from somewhere ...

lic_result = lic.lic(x, y, length=30)

plt.imshow(lic_result, origin='lower', cmap='gray')
plt.show()
```

Find out more about the options by reading the source documentation:

```
pydoc lic.lic
```

You can also control the seed, i.e., the underlying texture for the `lic`:

```
pydoc lic.gen_seed
```

You can run the example from the root folder to see the result:

```
PYTHONPATH="." python3 examples/ex1.py
```

3.3 API

There are two functions in the module `lic` that you might want to use, `lic.lic()` and `lic.gen_seed()`.

3.3.1 `lic`

3.3.2 `gen_seed`

3.4 Command line tool

You will need `numpy` data files (saved using `numpy.save`) to use `lic` from the command line:

```
lic data_x.npy data_y.npy -v -l 30 -c
```

See `lic --help` for a full list of options:

```
usage: lic [-h] [-l LENGTH] [-o FILE] [-c] [-q] [-v] [-d] [--version]
          data_x_file data_y_file

Line integral convolution (lic) algorithm. Please have a look at the
documentation (https://pypi.org/project/lic/) for further information on how
to use this software.

positional arguments:
  data_x_file          an numpy file containing a 2d numpy.ndarray with the x
```

(continues on next page)

(continued from previous page)

```
data_y_file          component of the vector field
                    an npy file containing a 2d numpy.ndarray with the y
                    component of the vector field

optional arguments:
-h, --help           show this help message and exit
-l LENGTH, --line-length LENGTH
                    the length of the line of the lic (default: 20)
-o FILE, --output-file FILE
                    the name of the output file, If it is not set, a name
                    will be generated from the names of the input data
                    files. (default: None)
-c, --enhance-contrast
                    enhance the contrast of the resulting lic image
                    (default: False)
-q, --quiet          switch off text output except for error messages. This
                    will overwrite -v. (default: False)
-v, --verbose        more verbose text output (default: False)
-d, --debug          switch on debug mode. This will show intermediate
                    results and plots, as well as log a lot of debugging
                    information. (default: False)
--version            show the version of this software
```


CHAPTER 4

Project links

- [Repository](#)
- [Documentation](#)
- [pypi page](#)

CHAPTER 5

External links

- <http://www.zhanpingliu.org/Research/FlowVis/FlowVis.htm>
- <https://www3.nd.edu/~cwang11/2dflowvis.html>