Fast Image Dehazing Using Dark Channel Prior For Python 3.6+

An implementation of the algorithm described in *Single Image Haze Removal Using Dark Channel Prior* [He et al. 09] (page), with the modifications proposed in *Guided Filtering* [He et al. 10] for faster transmission refinement.

Running

In order to run the program one needs:

- *Python 3.6+* installed
- *NumPy* installed.
- *scikit-image* installed.
- *numba* installed.

The `requirements.txt` file enumerates the requirements (`pip install -r requirements.txt`). The preferred way to manage dependencies is with Poetry, however. Just run `poetry install` for a venv for the project. If part of a bigger project, you may want a global install based on `requirements.txt`.

Leveraging the GPU with CUDA

For CUDA-based speedups, edit `pyproject.toml` to uncomment the `cupy` line, then run:

```
poetry update
poetry export --format requirements.txt -o requirements.txt
```

As an import

The preferred way to run this is by importing your preferred top-level functions from `dehazer.py`, eg,

```
from dehazer import dehazeDirectory
dehazeDirectory("dehazing/test_images", "dehazing/test_results", verbose= True,
report= True, checkSections= True)
```

`dehazeImage` and `dehazeDirectory` are the two main top-level functions; the convenience functions `dehazeDirectorySet` and `dehazeFolderOfDirectories` are shims around those.

Because of the just-in-time nature of Numba, the biggest speed benefits occur when the internals are run as a loop in a single environment instance, rather than many calls to single images. Numba compilation will induce an

approximately 2-5s delay per major step for the first processed image, which is avoided for all subsequent images if the Python environment doesn't have to be reloaded.

From the command line

With the performance caveat noted above, one should be able to run the program with the following command line (considering one is in the *src* folder):

```
$ python main.py -i ../images/cones.jpg -o ../results/cones_res.jpg
```

This programs calls the *main* module of the program to receive the arguments. The first argument *-i* is the path to the input image that will be dehazed. While the *-o* argument is the path to the output image, that is, the dehazed version of the input image. These are the only two required arguments.

For optional arguments, one can type:

```
$ python main.py -h
```

This will display the set of arguments available.

Importantly, do not run this command-line call as a loop. You will lose much of the performance benefits otherwise brought to the table by Numba.

Benchmarks and Results

A set of benchmark images can be found under the folder *images*. Most were taken from the main base paper page, but some were taken from the page of *Dehazing Using Color-Lines* [Fattal 14].

Results of applying the program to some of the benchmark images can be found under the folder *results*.

References

There is a document under *references* listing all the papers used in the development of this project. However, the two main references for this project were:

- *Single Image Haze Removal Using Dark Channel Prior* [He et al. 09], CVRP;
- *Guided Filtering* [He et al. 10], ECCV.

About

This project was developed as a Final Project for the "INF01050 - Computational Photography" class, 2016, at UFRGS by Carlo S. Sartori.

Migrated to Python 3 and enhanced by Philip Kahn