4D Use Case: Phase 2: Keypoint Generation

## Phase 2: Keypoint Generation

The execution of finding keypoints between image tiles. (It should be noted that many of the terms here refer to the ASIFT algorithm, although other algorithms are now also being used either in conjunction with or instead of ASIFT. Specific references to ASIFT are legacy.) The following steps are included.

## ASIFT Executable

A standalone set of functions for matching keypoints between two individual tiles. This is the code that does the majority of the work. Results are saved in the Scratch sub-Directory of the “source” image, including a “data\_matches.csv” file. This version of the code uses the “fast\_imas\_IPOL” executable, contained in the “fast\_imas\_IPOL” directory; however, other implementations (such as CUDA functions on a GPU node) can be swapped out here.

Usage in Python:

```

```

Usage standalone:

```

```

## ASIFT Class Definitions

A utility of class definitions used by the ASIFT Executable script.

Usage in Python:

```

```

Usage standalone:

```

```

## ASIFT Scheduler

A set of functions to handle the execution of all the “run cases” output by the “Run Case Planner” in the last phase. On a desktop workstation these could be run in serial, for instance. In an HPC environment, individual cases would be farmed out to individual nodes until execution is complete.

Usage in Python:

```

```

Usage standalone:

```

```

## ASIFT Performance Monitor

Measures performance metrics of the ASIFT Executable (time, memory), useful for internal optimization

Usage in Python:

```

```

Usage standalone:

```

```

## Dependencies

The code requires Python 2.7+ to execute. (A Python 3 version may be forthcoming soon.)

Code within the phases depends upon the following external Python libraries:

* GDAL v2.2+ : the GeoSpatial Data Abstraction Library ([www.gdal.org](http://www.gdal.org)), and associated Python “gdal” and “osgeo” bindings
* OpenCV v3.4+ : Open Computer Vision libraries and associated Python “cv2” bindings

## License

(TODO: Fill in the open-source license to be used here.)

## Acknowledgements

This project is funded by the NSF’s ICEBERG Cyber-Infrastructure grant.

Acknowledgements to Mariano Rodriguez for the “fast\_imas\_IPOL” executable (<http://github.com/rdguez-mariano/fast_imas_IPOL>), which CPU-versions of this code rely upon.

## Revisions:

1.0 – 2019.02.10 – Draft document by Mike MacFerrin