|  |  |
| --- | --- |
|  |  |
| **QRaFT( )** | Main module of the QRaFT package, called to process an image file |
| **read\_const\_string( )** | Reads a vector of processing parameter from a formatted string |
| **extract\_const( )** | Reads a single processing parameter from a string |
| **radial\_detrending**( ) | Removes an average radial intensity trend from a solar image |
| **rect\_to\_polar** | Transforms solar image from Cartesian to polar coordinates |
| **get\_coordinates( )** | Prepares an empty structure used by *rect\_to\_polar* |
| **patch\_image\_holes( )** | Patches small (1-3) groups of empty pixels using NN interpolation |
| **smooth\_polar( )** | Smoothens polar image using periodic BC in azimuthal direction |
| **azimuthal\_diff( )** |  |
| **detrend\_azimuthal( )** | Detrends polar image in azimuthal direction using periodic BC |
| **smooth\_polar( )** | See above |
| **linspace( )** | Generates a 1D array of uniformly spaced values within a given range |
| **trace\_blobs( )** | Returns geometric characteristics of the detected blobs in plane polar coordinates based on several combinations of the minimum radial distance used for tracing and the percentile detection threshold |
| **blob\_labeler( )** | Returns a structured array of labeled blobs and their basic statistics for a given combination of and the detection threshold |
| **blob\_analyzer( )** | Computes geometric characteristics of the blobs and the polynomial fits describing their average shape |
| **blob\_validator( )** | Returns filtered blob statistics and labels meeting validity conditions    NOT used in current version, conditions moved to feature\_validator() |
| **adapt\_thresh\_prob( )** | Generates an array of thresholds for a given data array based on a set of probabilities.    The thresholds are calculated using a cumulative probably distribution constructed for either positive or negative array elements. |
| **blob\_stat\_merger( )** | Merges two sets of blob characteristics into one set. Used to combine blob detection results obtained with different tracing parameters. |
| **blob\_stat\_to\_features( )** | Converts blob tracing results obtained by by *trace\_blobs()* describing blob location and geometric in polar coordinates into an array of structures containing the parameters of the detected blobs in Cartesian coordinates. The output parameters include:   1. The number of the interpolation nodes in each blobs; 2. Rectangular coordinates of the blob nodes; 3. The characteristic linear size of the feature; 4. Median value of feature intensity computed based on the original solar image with subtracted radial trend; 5. Orientation polar angles describing local orientation (clock angle) of each feature segment; 6. Interpolated rectangular coordinates corresponding to the point at which the local angles are evaluated; 7. The standard deviation of the local angles in a given feature.   The output is compatible with the structure “features” use in the previous implementations of the QRaFT package (versions 1.0 and 2.0). |
| **feature\_angles( )** | Computed local orientation angles of the detected features based on the rectangular coordinates of their interpolated nodes. |
| **feature\_validator( )** | Returns a validated set of features satisfying the filtering conditions      (sufficiently large length and intensity, sufficiently low characteristic curvature) |
| **feature\_aggregator( )** | Creates a set of 2D arrays containing aggregated parameters of local orientation angles for each location containing a detected feature. The arrays have the same dimensions as the original solar image.   1. The array containing the number of features passing through a given pixel. This number shows how many features detected using different combinations of tracing parameters overlap at a given location. 2. The array of ensemble-averaged orientation angles resulting from multiple tracing runs with different settings. 3. The array of standard deviations of the ensemble-averaged angles characterizing the disagreement of the tracing results at each location. 4. The array of relative orientation angles showing the departures of the feature orientation from the local radial direction. The relative angle is 0 if the feature is strictly radial at a given pixel. |
| **feature\_angles( )** | See above |
| **angles\_vs\_b( )** | Computes the plane-of-sky (POS) misalignment angles between the detected features and the magnetic field from a coronal simulation. At each location, the misalignment angle is zero if the feature is perfectly aligned with the POS projection of the magnetic field vector at that point. |
|  |  |
|  |  |
|  |  |