Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/7634--19-9858

Mighti Level 1 Data Analysis Summary

John Harlander

Space Systems Research Corporation
Alexandria, VA

Kenneth Marr and Christoph Englert

Geospace Science and Technology Branch
Space Science Division

May 15, 2019

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
15-05-2019

2. REPORT TYPE
Memorandum Report

3. DATES COVERED *(From - To)*
Jan 2017 - Feb 2019

4. TITLE AND SUBTITLE
Mighti Level 1 Data Analysis Summary

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
John Harlander, Kenneth Marr and Christoph Englert

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
4725

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Space Systems Research Corporation
Alexandria, VA 22314

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5344

8. PERFORMING ORGANIZATION REPORT NUMBER
NRL/MR/7634--19-9858

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
8800 Greenbelt Road
Greenbelt, MD 20771

10. SPONSOR / MONITOR'S ACRONYM(S)
NASA

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In this work we present the steps necessary to convert data from the MIGHTI sensors on the ICON Explorer mission from Level 0P to Level 1. Optical effects that are addressed include dark subtraction, thermal correction, and single emission line isolation, among many others. In addition to describing the analysis processes, we include associated mathematics, code flow charts, and code headers for reference.

15. SUBJECT TERMS
Interferometry
Image Analysis
Thermal Drift Correction

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(include area code)*
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	49	Kenneth Marr / (202) 767-2546

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

Contents

This page intentionally left blank.

Introduction

The Level 0´ output of the MIGHTI sensors are images of the Earth's limb recorded by a frame-transfer CCD detector from which neutral winds and temperatures of the Earth's upper Mesosphere/Lower Thermosphere can be measured as a function of altitude. An example image taken from a data simulation developed to test the level 1 software is shown in Figure 1. The wind measurements are obtained from a spatial fringe pattern produced by the atomic oxygen green (558 nm) and red (630 nm) line emissions at the left and upper right of Figure 1, respectively. The phase of these fringe patterns contains information about the atmospheric wind speed and the amplitude of the fringes contains information about the brightness of the emission. The separate temperature measurement (lower right of Figure 1) is determined from the relative brightness within three spectral channels which sample different parts of the near infrared molecular oxygen A-band. A more detailed description of the MIGHTI instruments can be found in Englert et. al. DOI: 10.1007/s11214-017-0358-4.

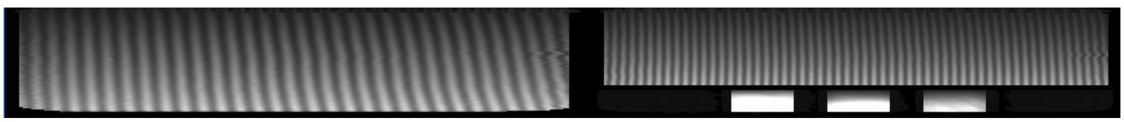


Figure 1. Representative image for MIGHTI. The green line fringe pattern is on the left half of the image, red line fringe pattern in the upper right, and five (3 illuminated) infrared channels for determining temperature on the lower right. The two non-illuminated IR channels (to the left and right of the three illuminated ones) provide a background measurement for the temperature derivation. Field of view tangent altitude is imaged vertically. The lowest green altitude is at a tangent height of 90 km and the highest is approximately 300 km.

1.0 Temperature Analysis

The Level 1 analysis of the temperature data is relatively straightforward. The signals from the infrared bands (3 signal channels and two background channels) are analyzed as follows:

- Correct for spikes due to hot pixels, star images, and/or cosmic ray events
- Dark subtract using the median value calculated from the four most recent dark exposures
- Correct for a small amount of pickup due to the smearing along rows as the image is shifted across the frame-transfer CCD sensor into the masked region
- Correct for changes in CCD gain if appropriate
- Convert to counts per second by dividing by the exposure time

Manuscript approved April 17, 2019.

These data are then passed to Level 2 where they are corrected for pixel to pixel response differences during the inversion process.

2.0 Analysis of Red and Green lines

The two primary outputs of the Level 1 wind analysis are the amplitude and phase of the fringe patterns (interferograms) as a function of pixel along a CCD row (horizontal direction in Figure 1) where each pixel in the row corresponds to the linearly varying optical path difference (OPD) of the interferogram. Along columns, the Earth's limb is imaged such that the CCD records a brightness profile of the emission from the Earth's limb covering an altitude range of approximately 90 – 300 km for the green line and 150 – 300 km for the red line. Atmospheric velocity information is contained in the phase of the interferogram whereas the amplitude carries information about the brightness of the emission but also has a weak dependence on atmospheric temperature. Emission brightness information is also contained in the mean signal value for each altitude after correcting for flat field and background signals. Both line-of-sight amplitude and mean signal values are saved at Level 1 but it is important to note that each have potential for systematic errors.

To obtain high-quality wind measurements the raw interferograms require processing to remove instrument effects such as pixel to pixel variations in sensitivity, instrument drifts with time, fringe distortions due to optical imperfections, CCD dark current, etc. During one orbit per 24 hour day, line emission from on-board calibration lamps produce a second set of fringes superimposed on the fringes from the atmosphere. The calibration lamp fringes are used to monitor instrument drifts both for the calibration orbit and subsequent orbits under the assumption that the instrument drifts are periodic with orbit. Special care must be used to separate the calibration and atmospheric signals in the analysis of fringe patterns obtained during the calibration orbit. Below we detail the Level 1 data analysis steps to extract the phase and amplitude of the fringe pattern for a variety of instrument modes. Note that there are two MIGHTI sensors, MTA and MTB, and each of them operate during both day and night. For calibration purposes, each of these four cases is treated as a different instrument. As a result, although the process for determining phase and amplitude is the same for all, the analyses are performed separately for each of them as they have different flat field, phase distortion, zero wind phases, and visibility correction calibration files.

2.1 Analysis for Interferograms Containing a Single Emission Line

The simplest images from which to extract the phase and amplitude of an emission line are those containing a single fringe pattern. The atmospheric exposures that are taken when the calibration lamps are off, which is the majority of them, fall into this category as do two per day calibration-only exposures (one in day mode and one in night mode) taken during the calibration orbit with the lamps on but with the aperture A1 closed, blocking the signal from the atmosphere. Here we describe the steps taken to analyze the data for phase and amplitude. Analysis of multi-line images will be addressed in Section 4.

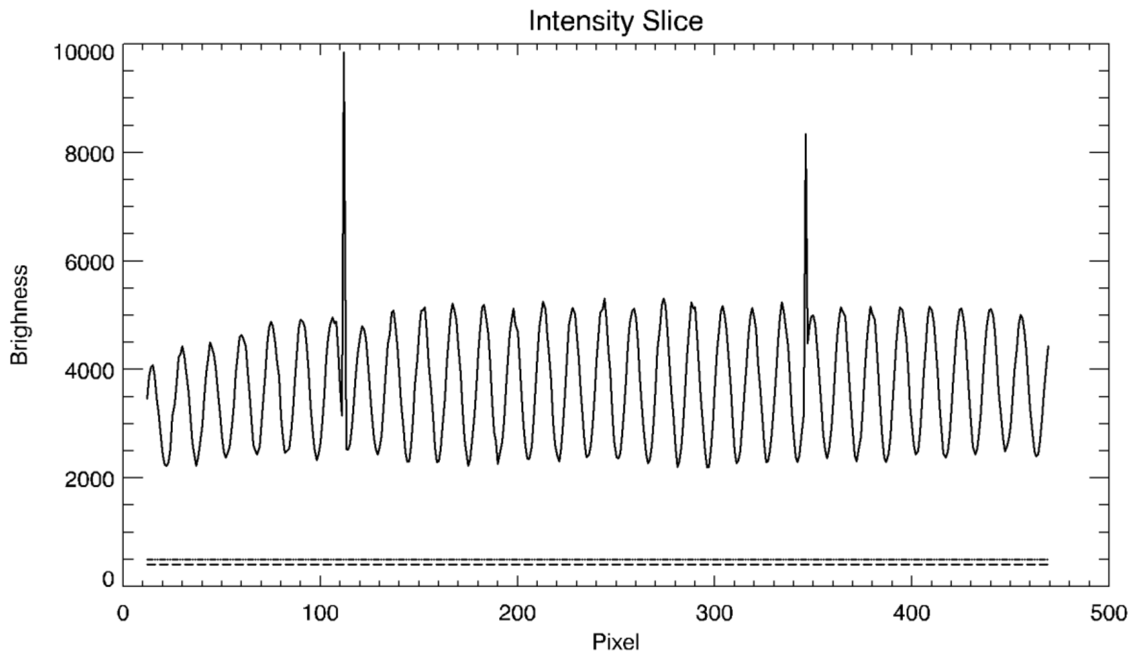


Figure 2. Brightness vs pixel for one altitude slice of green line atomic oxygen fringe (interferogram) from the simulated image in Figure 1. The primary data products of the Level 1 analyses are the phase of the interferogram which increases by 2π for every complete fringe and the fringe amplitude. There are numerous instrument artifacts evident in this plot that are removed in the Level 1 processing as described below.

2.1.1 **Dark and bias current subtraction.** Every image taken by the MIGHTI sensors has, in addition to light that falls on the detector, a DC bias and dark current, which is the spontaneous creation of electrons in the detector elements. To remove these effects from the images, 'dark frames' are taken and are subtracted from the raw images. To reduce noise contributions on the dark + bias frames a composite dark image is obtained from the mean of the most recent four dark images obtained during the calibration orbits. The dark + bias images are also spike corrected prior to subtraction. In Figure 2 the dark + bias signal is indicated by the dashed line at brightness value ~400. Each pixel will have slightly different dark + bias current.

2.1.2 **Pick-up subtraction.** The MIGHTI CCDs are frame transfer devices which means when the integration time is complete the image transfers from the active area of the CCD to a masked area for readout. This results in some smearing of the image as the data moves quickly through still illuminated pixels (no shuttering). The CCDs are oriented so this image transfer is along rows (constant altitude) rather than along columns (constant optical path difference). This results in a small constant term being added to each pixel in any given interferogram which is proportional to the total row brightness. The amount to subtract is calculated for each row using the mean signal along the row multiplied by the ratio of the frame transfer time (307.5 ms) to the exposure time (30 or 60 s). For illustrative purposes the pick-up in Figure 2 is indicated as the ~100 counts above the dark signal. It is typically much less than 100 counts.

2.1.3 **Divide by proper flat field.** Flat field calibration files obtained during TVAC testing are used to correct for pixel to pixel variations in sensitivity caused either by the CCD sensor itself or the optical system (e.g. vignetting, variations in transmittance, etc.). Note that due to their different wavelengths, the calibration and atmospheric line images have different flat fields. In this step the flat field calibration file corresponding to the line being analyzed is used. Some of the pixel to pixel variation evident in Figure 2 is due to the flat field response.

2.1.4 **Spike correct image.** Spike correction is performed to eliminate cosmic ray events from the images that would otherwise result in additional scatter in the phase and amplitude results. There are two simulated spikes in Figure 2. First, the image is normalized (flattened) by dividing each row (altitude) by its median brightness to remove the atmospheric brightness profile. Then spikes are identified by using a median filter along columns (constant optical path difference). Identified spikes are replaced by least-squares interpolating the rest of the flattened column onto the identified pixels. Finally, the normalization is removed, restoring the atmospheric brightness profile to the data.

2.1.5 **Calculate brightness.** Similar to the process in the spike correction, the mean brightness per row is calculated for each altitude for both the red and the green wind measurement. Using the appropriate day/night, MTA/MTB calibration factor, this relative brightness profile (in DN) is converted to a best estimate of absolute brightness in Rayleighs based on ground based calibrations during TVAC. Because of uncertainties in the calibrations this estimate is recorded as a relative measurement at Level 1.

2.1.6 **Remove DC component.** In order to correct the interferograms for instrument visibility variations (see below) the low frequency component must be removed. In Figure 2 this component is the distance between the dot-dash line at ~500 counts and the midpoint of the fringe modulation. In practice the low frequency component is removed by Fourier transforming, blocking low frequencies in spectral space and inverse transforming to obtain a fringe pattern with modulation symmetric about zero. During this step a Hann function apodization is applied row by row to mitigate edge effects of the Fourier transform due to the non-periodic boundaries of the interferogram and to facilitate later processing.

2.1.7 **Correct for phase distortion.** Ideally a plot of phase vs pixel for a MIGHTI interferogram is linear in phase which corresponds to equally spaced interferogram fringes, however, due to small variations in the ideal optical path difference caused by non-flat optical surfaces or index of refraction variations, the recorded fringes have unequal spacing. This is most evident in Figure 1 by the curvature of the fringes. To correct for this effect, the phase of a fringe was measured during TVAC with the instrument illuminated by a homogenous diffuse emission line source. A perfect linear phase was then subtracted from this phase to obtain the phase distortion calibration image which is then used to correct on-orbit images.

2.1.8 **Correct for instrument induced amplitude variations.** Due to slight differences in the optical efficiencies of the interferometer arms (e.g. grating reflectivity, transmittance, etc.) the recorded fringe amplitude varies across the interferogram (cf. Figure 2). To accurately invert the line of sight measurement for wind, the fringe amplitude should be constant vs path difference and altitude when viewing a homogenous diffuse source. To remove the instrument induced amplitude variations, a calibration fringe amplitude was measured during TVAC with the field of view illuminated with a diffuse emission line source. This calibration amplitude is divided into the interferogram row by row after removing the DC component to produce a corrected interferogram.

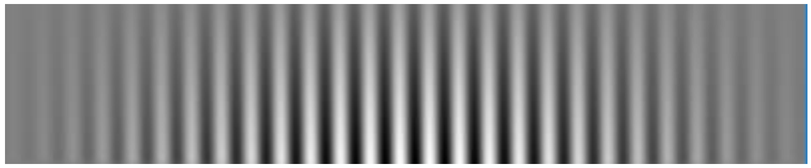


Figure 3. The green section of Figure 1 (left half of image) after correcting for dark, pickup, amplitude and phase distortion. Note the straightness of the fringes compared with Figure 1. Also evident is the Hann function apodization that tapers the amplitude to zero at the edges of the field.

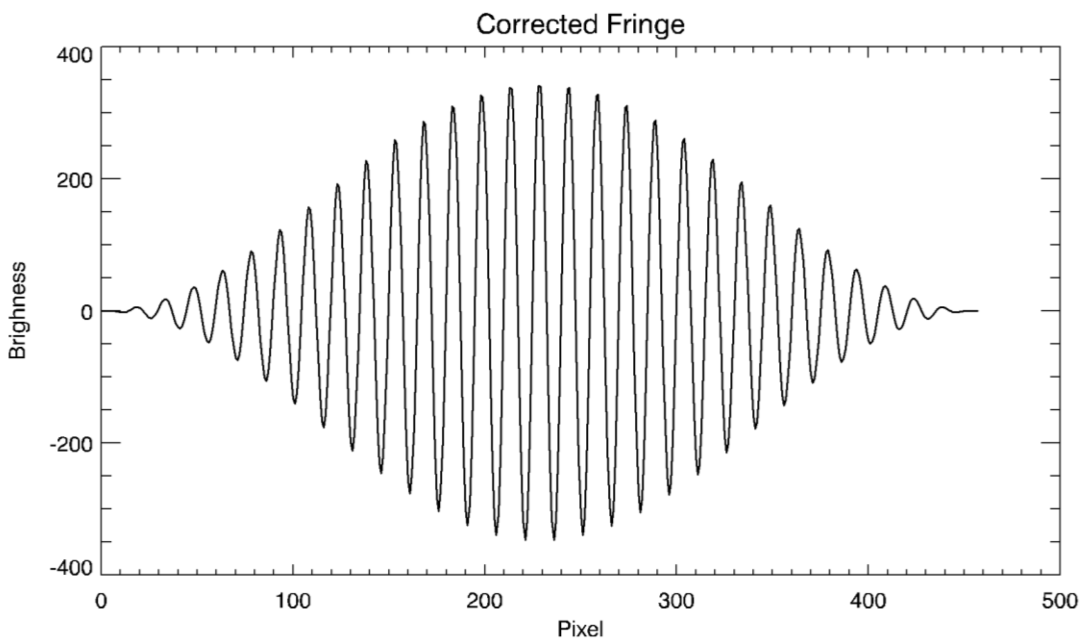


Figure 4. Intensity slice from Figure 3 after correction for instrument effects. This is the same row shown before correction in Figure 2.

2.1.9 **Isolate the spectral peak and inverse transform.** The phase vs pixel can be calculated by isolating one of the peaks in the spectrum (Fourier transform of the corrected interferogram), and inverse transforming to obtain a complex interferogram. Figure 5 shows the isolation in spectral space. In the case of spectra with both calibration and atmospheric lines (cf. Figure 9) the width of the spectral isolation function is chosen to isolate a single line.

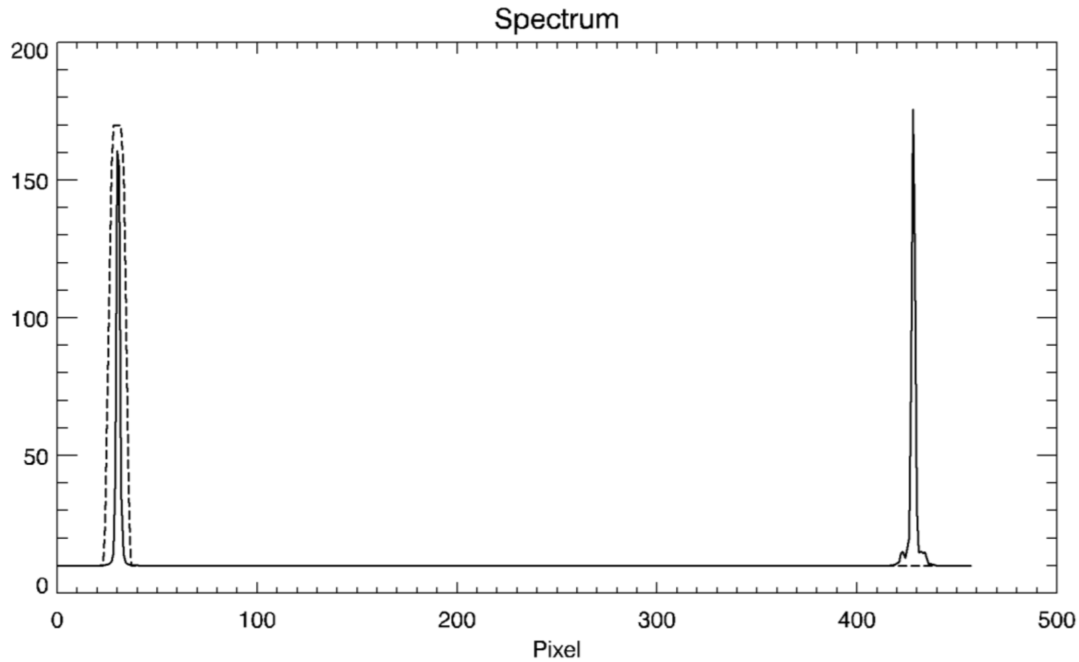


Figure 5. Isolation of the positive peak in the transform spectrum. The power spectrum of the plot from Figure 4 is shown (solid line) with the associated isolation function (scaled, dashed line). Note that a by-product of phase distortion correction is that the conjugate peak in the spectrum, on the right of the figure and removed with the isolation filter, receives twice the phase error as evidenced by its distorted line shape.

2.1.10 **Calculate phase and amplitude.** The phase is calculated by taking the inverse tangent of the imaginary part over the real part of the complex interferogram. A contiguous, or cumulative, linear phase is obtained by unwrapping the 2π jumps in the phase. Figure 6 shows the unwrapped phase vs pixel. The amplitude of the fringe, which is equal to the amplitude of the complex interferogram, is also saved at this step.

2.1.11 **Subtract reference phases.** In the final steps, as described below, a reference phase determined by calibration lamps is subtracted from the cumulative phase from step 2.1.10 in order to compensate for thermal effects. The resulting phase difference is then further subtracted from a phase difference derived from the “zero wind maneuver,” which is the calculated phase difference between the calibration lamps and an atmosphere at rest. This difference of differences is calculated along the entire optical path difference and passed to the Level 2 analysis.

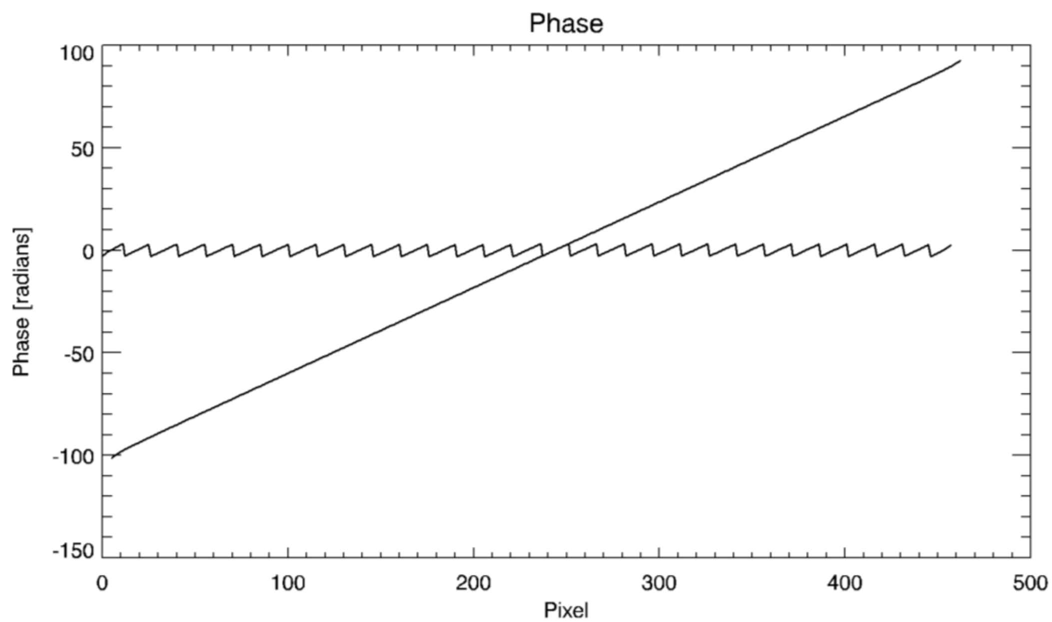


Figure 6. Phase vs pixel obtained from the inverse tangent. Each fringe corresponds to a change in phase of 2π. Also shown is the unwrapped phase where the 2π jumps have been removed.

3.0 Correction for Thermal Drifts and Zero Wind Determination

The line of sight atmospheric wind velocity is contained in the phase vs pixel (OPD, optical path difference) shown in Figure 6. Measuring this velocity requires that thermal drifts of the instrument be removed. This is accomplished by monitoring the fringes from an on-board calibration lamp with an emission line spectrally close to the atmospheric line. For the red line a neon emission at 630.4 nm is used and for the green line a krypton emission at 557.03 nm is used. As will be shown below, utilizing the calibration lamps requires that the phase difference between the calibration line and the atmospheric line for an atmosphere at rest (zero wind) be known. A special maneuver during which the field of view is oriented to view the limb towards the ram of the spacecraft, followed by a second observation where the field of view is oriented to view the same location on the limb towards the wake is referred to as the zero-wind calibration. To motivate the analysis, we start with a discussion of the calculations pertaining to thermal drift correction and zero wind calibration.

3.1 General atmospheric measurement

The phase vs pixel of an interferogram from an atmospheric emission line (cf. Figure 6) can be represented as follows:

$$\Phi_A = \Phi_{A0} + a^*(v_s + v_A) + \delta\phi_{DA} \quad (1A)$$

Φ_A is measured phase vs pixel of atmospheric line observed for a particular altitude (CCD row) on the scene

Φ_{A0} is phase vs pixel of atmospheric line for zero wind, no spacecraft velocity and zero interferometer drift
a is $2\pi\sigma\Delta/c$, where Δ is path the difference vs pixel, σ is the signal wavenumber, c is the speed of light
v_s is the projection of the spacecraft velocity along the line of sight, which is known
v_A is the atmospheric line of sight velocity
and
$\delta\phi_{DA}$ is the thermal drift for the atmospheric line which is a function of time.

$\delta\phi_{DA}$ has two possible components. This first is the drift of the interferometer that is due to thermal changes of the path difference inside the interferometer. Over small wavelength ranges the interferometer drift can be considered as independent of fringe frequency and is therefore the same for atmospheric and calibration lines. The second source of possible drift is due to a mechanical shift of the detector with respect to the interferometer caused by a differential thermal motion of the interferometer assembly, exit optics or detector. This drift scales linearly with fringe frequency and is monitored using registration marks ("notches") that are burned into one the interferometer gratings and imaged on the CCD. The notches can be seen along the top rows of Figure 1. $\delta\phi_{DA}$ can be expressed as:

$$\delta\phi_{DA} = \delta\phi_T + \delta\phi_{NA} = \delta\phi_T + f_A\Delta_N \qquad (1B)$$

where $\delta\phi_T$ is the interferometer drift assumed to be the same for both atmospheric and calibration lines (Littrow shift) and $\delta\phi_{NA}$ is the phase shift of the atmospheric line due to the shift of the grating registration marks on the detector.

As indicated on the right-hand side of equation 1B, $\delta\phi_{NA} = f_A\Delta_N$ where f_A is the measured fringe frequency for the atmospheric line in units of phase per pixel and Δ_N is the measured shift in pixels of the grating registration marks. Zero reference for this shift is the same predefined zero state of the interferometer used in the zero wind analysis (below).

The phase vs pixel for the calibration lamp fringe pattern taken during calibration orbit can be expressed as:

$$\Phi_C = \Phi_{C0} + \delta\phi_{DC} \qquad (2A)$$

Φ_C is the measured phase vs pixel of the on board calibration line
Φ_{C0} is the phase vs pixel of the calibration line at zero interferometer drift
$\delta\phi_{DC}$ is the phase drift of the calibration line. Following equation 1B $\delta\phi_{DC}$ can be written as

$$\delta\phi_{DC} = \delta\phi_T + \delta\phi_{NC} = \delta\phi_T + f_C\Delta_N \qquad (2B)$$

where f_C is the measured frequency of the calibration line in units of phase per pixel.

Combining equations (1A) through (2B) gives

$$\Phi_A - \Phi_C = (\Phi_{A0} - \Phi_{C0}) + a*(v_s + v_A) + (f_A - f_C)*\Delta_N \quad (3)$$

which is independent of the interferometer thermal drift, $\delta\phi_T$.

The first term on the right-hand side ($\Phi_{A0} - \Phi_{C0}$) is obtained from the zero-wind maneuver described in section 3.2.

3.2 Zero wind maneuver

The zero wind maneuver results in two exposures with opposite look directions, one viewing the limb towards the direction of the satellite velocity (ram) and the other following a 180° satellite yaw that views the same atmospheric volume on the limb but in the backward direction (wake). These exposures contain both atmospheric and calibration lines. Defining $\delta\phi_{DAZ1}$, $\delta\phi_{DCZ1}$ and $\delta\phi_{DAZ2}$, and $\delta\phi_{DCZ2}$ as the thermal drifts for both exposures (Z1 for ram and Z2 for wake) and for both sets of lines (subscript A for atmospheric and C for calibration), the phase vs pixel for each image and line can be expressed as

Atmospheric line:

$$\Phi_{ZA1} = \Phi_{A0} + a*(v_s + v_A) + \delta\phi_{DAZ1} \quad \text{: first measurement in ram} \quad (4)$$

$$\Phi_{ZA2} = \Phi_{A0} - a*(v_s + v_A) + \delta\phi_{DAZ2} \quad \text{: second measurement in wake} \quad (5)$$

Note the change of sign of the velocity terms.

Calibration line:

$$\Phi_{ZC1} = \Phi_{C0} + \delta\phi_{DCZ1} \quad \text{: first measurement in ram} \quad (6)$$

$$\Phi_{ZC2} = \Phi_{C0} + \delta\phi_{DCZ2} \quad \text{: second measurement in wake} \quad (7)$$

As before in equations 1B and 2B, each of the $\delta\phi_{Dxxx}$ terms can be expressed as the sum of a corresponding interferometer drift $\delta\phi_{Tx}$ and grating shift on the detector, $\delta\phi_{Nx} = f_x\Delta_{Nx}$.

Combining eqns (4), (5), (6), (7) and substituting for the drift terms gives:

$$\Phi_{A0} - \Phi_{C0} = ½[(\Phi_{ZA2} - \Phi_{ZC2}) + (\Phi_{ZA1} - \Phi_{ZC1}) - (f_A - f_C)*(\Delta_{N1} + \Delta_{N2})] \quad (8)$$

Where Δ_{N1} is the grating shift in pixels for exposure Z1 and Δ_{N2} is the corresponding value for exposure Z2 from a reference position for zero notch shift determined during instrument check-out on orbit.

The left side of equation 8 is the phase difference between a zero-wind atmospheric line and the calibration line. It is a fundamental property of the interferometer that is independent of thermal drift.

3.3 Combining general atmospheric measurement and zero wind

Equation (3) can be now be used to determine v_A as follows

$$a*(v_s + v_A) = (\Phi_A - \Phi_C) - (\Phi_{A0} - \Phi_{C0}) - (f_A - f_C)*\Delta_N \quad (9)$$

Note that v_s, the projection of the satellite velocity onto the line of sight, is removed in the level 2 analysis.

The first term in parentheses on the right side of equation 9 is obtained from the analysis of each atmospheric image while the second term in parentheses is from the zero-wind maneuver (cf. equation 8).

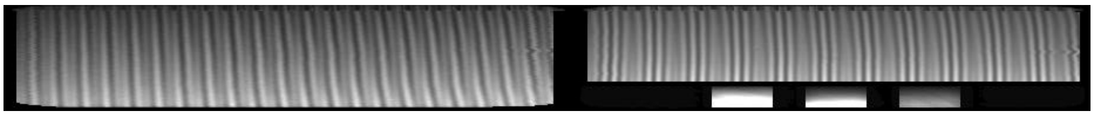


Figure 7. Example Level 0 image similar to Figure 1 but with the calibration lamps on. Notice the additional superimposed set of fringes from the calibration lamps.

4.0 Practical concerns

Equation 8 requires that the phase vs pixel be obtained for multiple images and for images containing superimposed fringe patterns of two spectral emission lines. There are several practical issues surrounding accurate determination of the phases:

- During normal operations the calibration lamps used to monitor the thermal drift will be on for only one orbit per 24-hour day. The measured thermal drift will need to be applied to data from subsequent orbits when the calibration lamps are not on. Most images during the calibration orbit will have both calibration and atmospheric lines, however, there are two calibration-only images (one for day and one for night) taken with aperture A1 closed, which blocks the signal from the atmosphere.

- As indicated above, the zero-wind maneuver consists of two images, one with the FOV pointed toward the spacecraft ram and one toward the wake. These images will contain both calibration and atmospheric lines.

- Analysis of images with both calibration and atmospheric lines are complicated by the fact that the two lines have different flat fields. This requires special processing to avoid the contamination of one line by the flat field of the other (see Figure 9).

- The fringe phase and amplitude in interferogram (fringe) space contains non-periodic boundary conditions, which creates artifacts near the edges of the transform window during discrete Fourier Transform processing. To reduce the effect of these artifacts on the derived winds, the phase and amplitudes are trimmed at each edge.

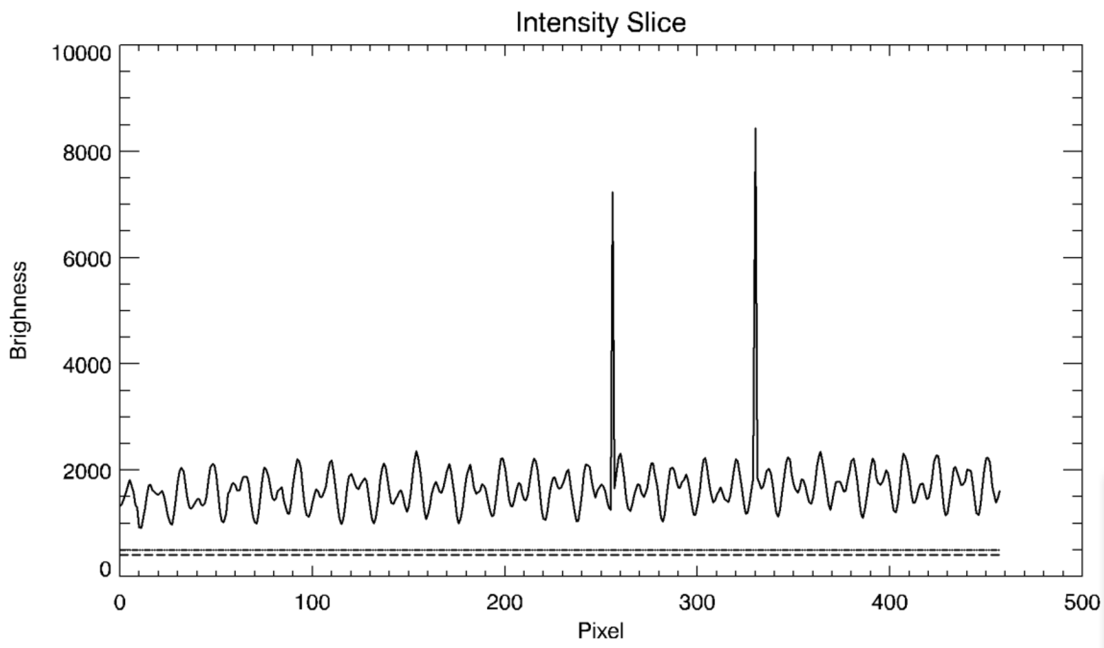


Figure 8. Brightness vs pixel for a slice through the green fringe pattern from Figure 7 when both the calibration and atmospheric lines are present. The double modulation is evident (compare with Figure 2).

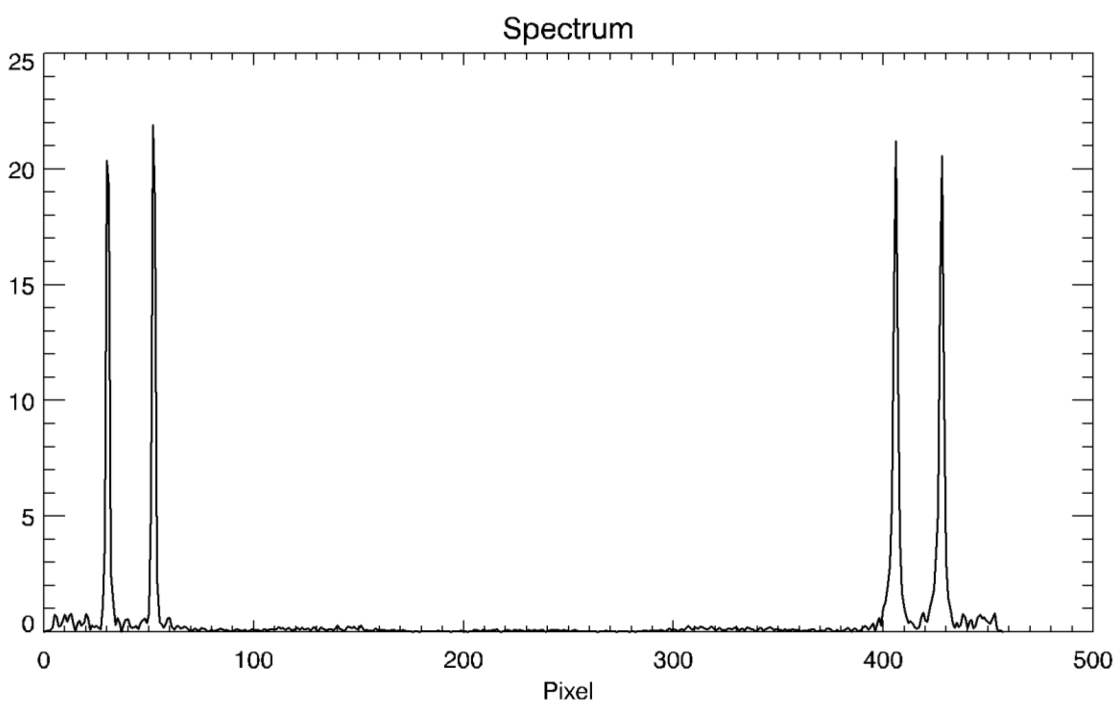


Figure 9. Amplitude Fourier transform of the slice shown in Figure 8 after correcting for dark, flat field, pickup, amplitude variation and phase distortion. The atmospheric line is near a frequency (fringes per window) of 23 while the calibration line is near a frequency of 43. The flat field for

the atmospheric line was applied to the image during the correction process. Note the fluctuations near the feet of the lines which is due to the atmospheric flat field applied to the calibration line. This contamination, which is a feature of all two-line interferograms, requires special processing to derive high-quality phase and amplitude information for both the calibration and atmospheric lines.

4.1 Extracting calibration phases from two-line data

To determine the calibration phase Φ_C for a given row for two-line interferograms while mitigating contamination from the atmospheric line, the following steps are used:

4.1.1 **Analyze calibration-only image.** Retrieve the closest-in-time calibration-only image and calculate phase vs pixel of calibration line using the standard technique detailed in Section 2. This provides a "high-quality" reference phase for the calibration line to be used in two-line analysis (Φ_{C0} in Equation 2).

4.1.2 **Reprocess calibration-only image and two-line image.** Divide each image by the *atmospheric line* flat field. Fourier transform the interferograms from both images. This results in a clean atmospheric spectral line in the two-line spectrum and creates an equally contaminated calibration spectral line for the calibration-only and two-line spectra. Two-line spectra using the atmospheric line flat field is shown in Figure 9.

4.1.3 **Isolate calibration line.** Using a spectral filter, remove the atmospheric line from the two-line spectra.

4.1.4 **Fit spectra for thermal shift.** Find the phase shift between the processed calibration-only and the isolated calibration spectra from the two-line image by fitting the first to the second in the spectral domain with free parameters of phase shift and amplitude. This phase shift is $\delta\phi_{DC}$ in equation 2A.

4.1.5 **Calculate calibration phase for two-line image.** Add the phase shift from step 4.1.4 to the calibration-only phase obtained in step 4.1.1 to obtain high-quality phase vs pixel for the calibration line Φ_C. A similar process is used to obtain Φ_{ZC1} and Φ_{ZC2} (cf. equations 6 and 7).

4.2 Extracting atmospheric phases from two-line data

To determine the atmospheric phase Φ_A for a given row the following steps are used:

4.2.1 **Get pure calibration-only phases.** Obtain the closest-in-time previous calibration-only image and dark subtract. Do not flat field. Also retrieve the thermal drift information by time in orbit. Add thermal shift to calibration-only image.

4.2.2 **Dark correct the two-line image to be analyzed.** Do not flat field.

4.2.3 **Remove calibration lines.** Subtract thermally shifted calibration-only fringe image from the two-line fringe image to obtain an atmosphere-only fringe image with a small calibration line residual component. Prior to subtraction, the brightness of the calibration-only fringe is scaled to the brightness of the calibration line in the two-line fringe. The scale factor is determined from the areas under the respective lines in frequency space. After the isolation function is applied, the residual calibration component will have a negligible effect on the determined atmospheric phases.

4.2.4 **Process the atmosphere-only fringe image.** Take the resulting image from step 4.2.3 and analyze using the single-line process detailed in Section 2. This method is also used to obtain Φ_{ZA1} and Φ_{ZA2} for the zero wind maneuver analysis (cf. equation 8).

4.3 Obtaining the calibration line phase, Φ_C for single-line atmospheric data

Except for the calibration orbit, all atmospheric data during normal operations will consist of single-line images. These are analyzed to obtain Φ_A using the process described in Section 2. However, to obtain wind speeds using equation 9, an appropriate calibration lamp phase Φ_C is required. Assuming that the thermal drift is periodic with the orbit the following process is used to obtain Φ_C for single-line atmospheric data.

4.3.1 **Determine time in orbit.** Obtain the time since the day-night spacecraft terminator crossing for the file in question from the ancillary file.

4.3.2 **Determine thermal phase shift.** Interpolate the time since terminator variable from the previous calibration orbit in the Thermal Drift file to the current time in orbit to determine a current thermal phase shift.

4.3.3 **Retrieve calibration-only phases.** Obtain the phases of the calibration-only image that is saved in the Thermal Drift file.

4.3.4 **Create expected calibration phase.** Calculate the calibration phase term Φ_C by adding the thermal phase shift to the calibration-only phases.

5.0 Noise estimation

Formulas to estimate the uncertainty of the line of sight level 1 amplitudes and phases have been developed and tested using a Monte Carlo approach for both a simplified single line with background calculation, where the various parameters were varied over a wide range of values, and the Level 1 data analysis simulation including the various processes/corrections described above. Englert et. al. (2007) details the phase uncertainty on a pixel by pixel basis. Here the analysis is expanded to include the amplitude and formulas for the uncertainties averaged along an entire row.

5.1 Uncertainty estimates for each pixel

As given by Englert et. al. (2007) the phase uncertainty for each pixel is given by:

$$\sigma_{ph} = \mathrm{sqrt}(2 * I_{Tot} * n)/I_{sig} \qquad (10)$$

where I_{Tot} is the total number of detected photons in the row n is the width in pixels of the spectral filter used when isolating the line of interest in Fourier space and I_{sig} is the total number of detected photons in the line of interest.

Similarly, the fringe amplitude uncertainty for each pixel is given by:

$$\sigma_{Amp} = \mathrm{sqrt}(I_{Tot} *n/2)/N \qquad (11)$$

where N is the total number of pixels in the row. Note that this depends only on the total number of photons, not the number of photons in the line of interest. This general property of Fourier transforms reflects the fact that each spectral element receives the same noise.

5.2 Uncertainty estimates averaged along a row

The calculation of the mean uncertainty along a row is complicated by the trimming applied to each row (described above) to mitigate the effect of large Fourier transform induced phase and amplitude deviations due to non-periodic boundary conditions at the beginning and end of each row. The effect of the trimming is to reduce the total number of detected photons. Formulas for the mean uncertainty and phase are:

$$<\sigma_{ph}> =\mathrm{sqrt}(2*I_{Tot}*N/N_{trim})/I_{sig} \qquad (12)$$

and

$$<\sigma_{Amp}>= \mathrm{sqrt}(I_{Tot}/2*N/N_{trim})/ N \qquad (13)$$

Where N_{trim} is the number of pixels in the row after trimming. Note that equations 12 and 13 are different from 10 and 11 by a factor of sqrt(1/n*N/ N_{trim}) and are thus independent of n, the width of the spectral filter applied to isolate the line of interest in Fourier space. The factor N/ N_{trim} simply reflects the reduction in numbers of photons and pixels along the row due to trimming.

6.0 Implementation in the L0´ to L1 code

Below are skeleton outlines of the IDL codes used to implement the above tasks. Specific routines are in **bold**. Explanatory comments follow a semicolon (;). Italics indicates processes or calculations performed at this location in the algorithm. Italics surrounded by parenthesis (e.g. *(2.1.3)*) are references to sections in the above document.

```
MIGHTI_L0P_to_L1_v1_0_4

      readNetCDF  ;reads ancillary file
      analyzeImage

END

analyzeImage

      readNetCDF ;reads image file
      Restore Parameters .dat File
      callDarks
      verifyDarks_v6 ;subtracts dark (2.1.1)
      pickUpRemove_v4 (2.1.2)
      removeCalLine_v2 ;if calibration lamps are on (4.2.3)
      callFlatField_v3 (2.1.3)
      spikeCorrect_v8 (2.1.4)
      calcBright  ;calculates mean brightness
      removeDC_v6 (2.1.5)
      vis_and_phase_correction_v5 (2.1.6, 2.1.7)
      Calculate Phase and Envelope row by row (2.1.8, 2.1.9)
      calcECEF_v3 ;finds ECEF vectors for each pixel
      deltaPhaseCalc_v2 ;Correct phase for thermal drift, notch shift
                         (3.0,3.1,4.3)
      Calculate Phase and Envelope Uncertainties (5.2)
      Extract IR signal (1.0)
      Prepare Output Variables
      addAttributesScience
      writeNetCDF_km

END
```

MIGHTI_create_thermal_drift_v1_0_3

Restore Parameters .dat File
ReadNetCDF ; Read file to set parameters
Find Cal Only Images
analyzeCalOnlyImage
Read calibration orbit files
analyzeThermDrift
Prepare thermal drift file
Write thermal drift file

End

analyzeCalOnlyImage

Restore Parameters .dat File
ReadNetCDF ;read image file and zero wind calibration file
callDarks
verifyDarks_v6 ;subtracts dark *(2.1.1)*
pickUpRemove_v4 *(2.1.2)*
callFlatField_v3*(2.1.3)*
spikeCorrect_v8 *(2.1.4)*
removeDC_v6 *(2.1.5)*
vis_and_phase_correction_v5 *(2.1.6, 2.1.7)*
Calculate Phase and Envelope row by row (2.1.8, 2.1.9)
Create_notch_position
sheetFitPhases ;fits 2d plane to phases

End

analyzeThermDrift *(4.1)*

Restore Parameters .dat File
callDarks
verifyDarks_v6 ;subtracts dark
callFlatField_v3 ;Flat cal+atm image with atm flat *(4.1.3)*
callFlatField_v3 ;Flat cal_only image with atm flat *(4.1.3)*
Fit for thermal drift in frequency space (4.1.4)
Create_notch_position
polyFitPHaseShift ;Fit phase vs OPD with quadratic

End

7.0 Flowcharts for L0P to L1 codes

ICON MIGHTI L0P – L1 Code Flow Charts

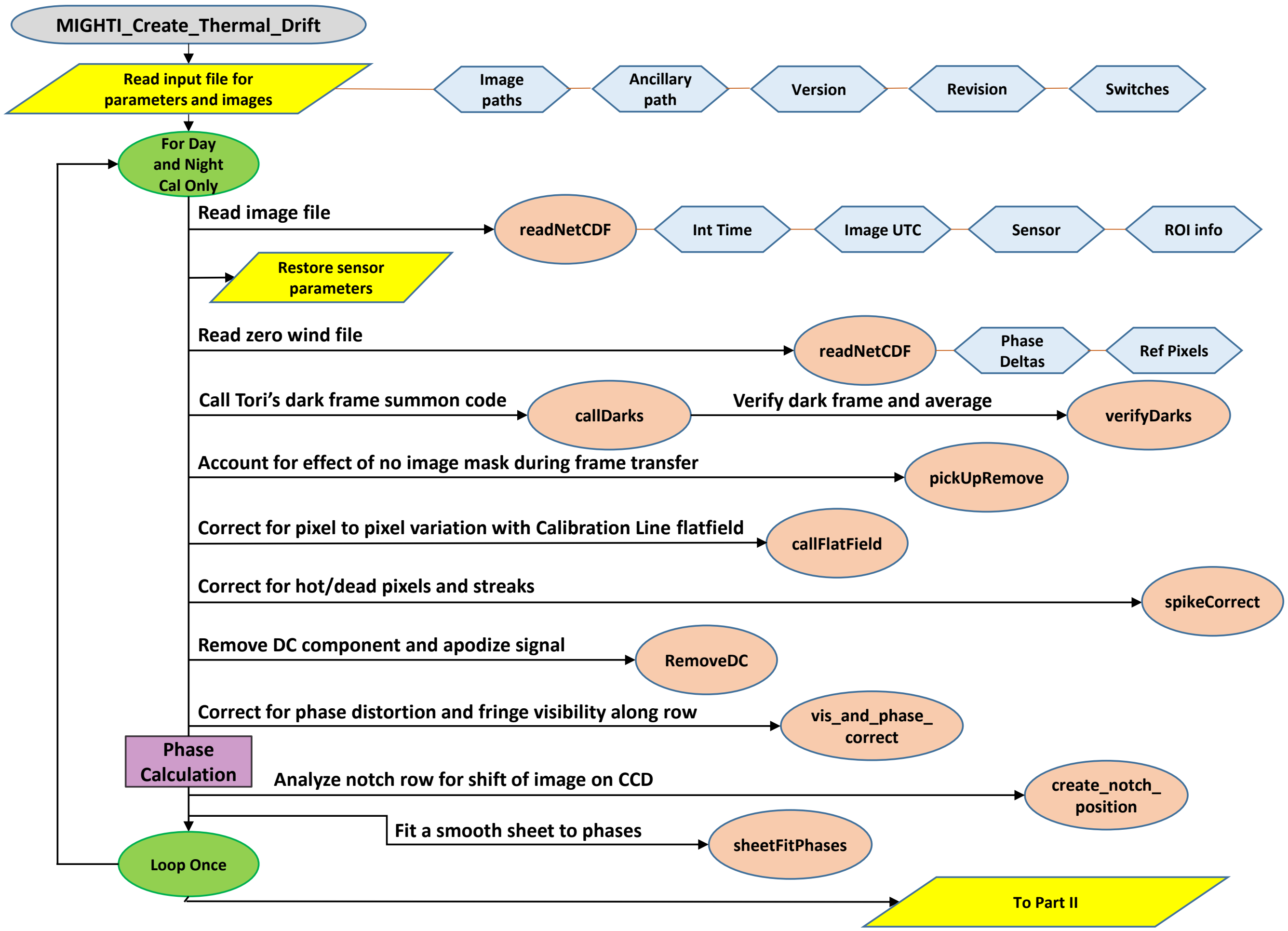
K. Marr
August 9, 2018

Key to Symbols

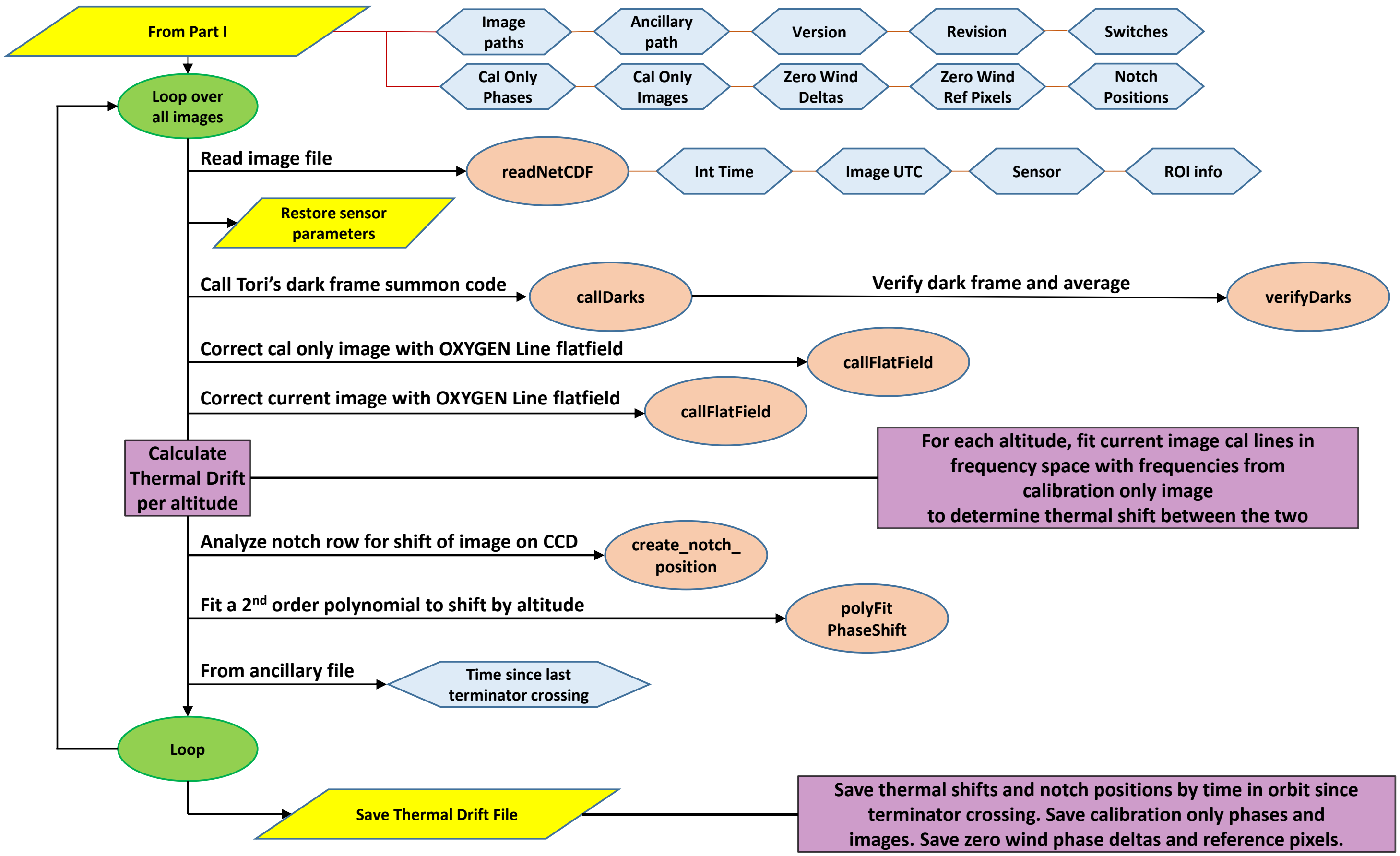
- **Tan circles indicate subroutines**
- **Green ovals are beginning and end of do-loops**
- **Yellow parallelograms represent input and output operations**
- **Purple rectangles are used to indicate any processing operations**
- **Gold diamonds are used to indicate questions asked or conditions**
- **Blue hexagons indicate parameters or constants**
- **Arrows indicate the direction to be followed in a flow chart**
- **Solid lines are completed or near complete.**
- **Dashed lines need more work before inclusion**

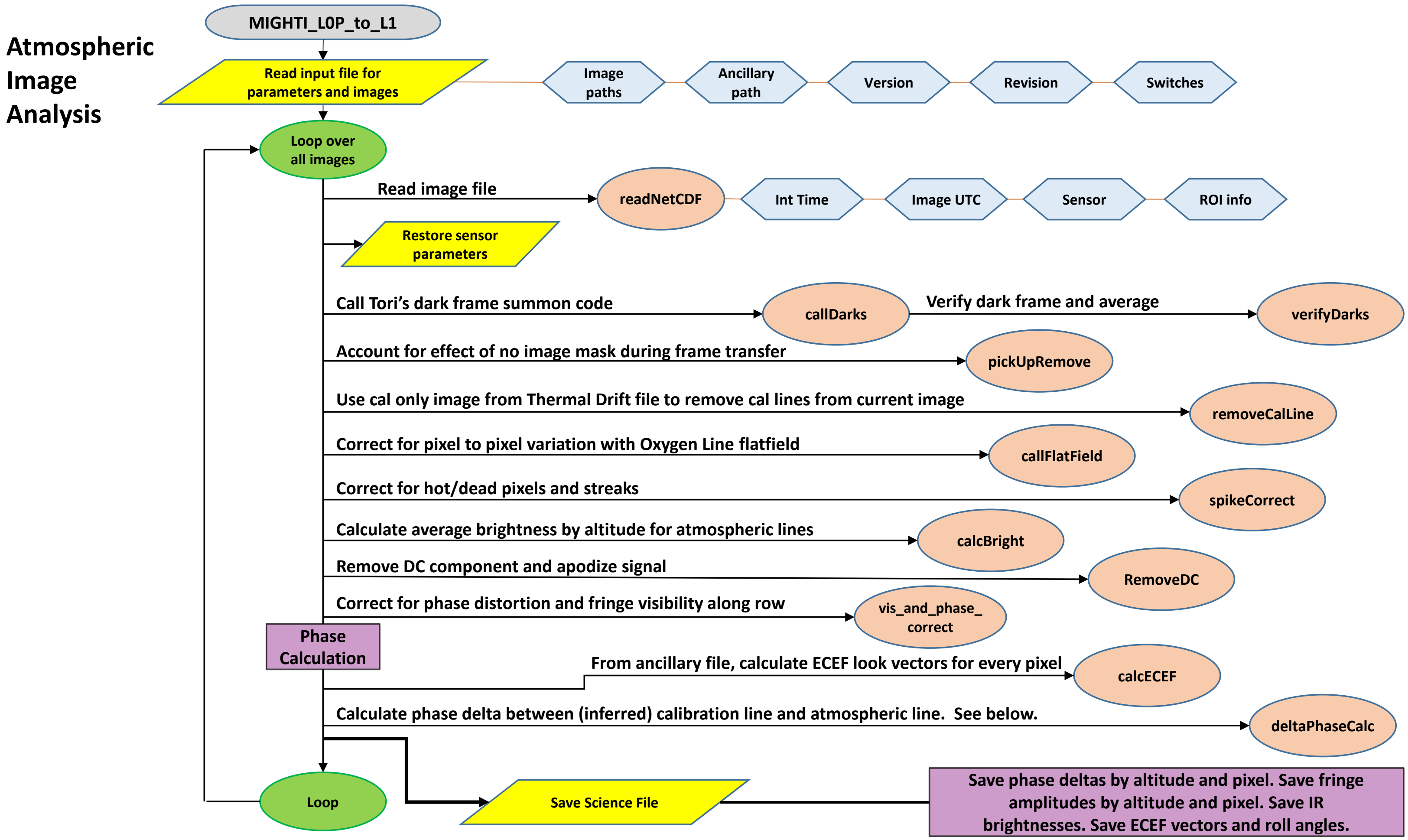
Codes will be run in the order indicated.

Cal Lamp Orbit Analysis Part I



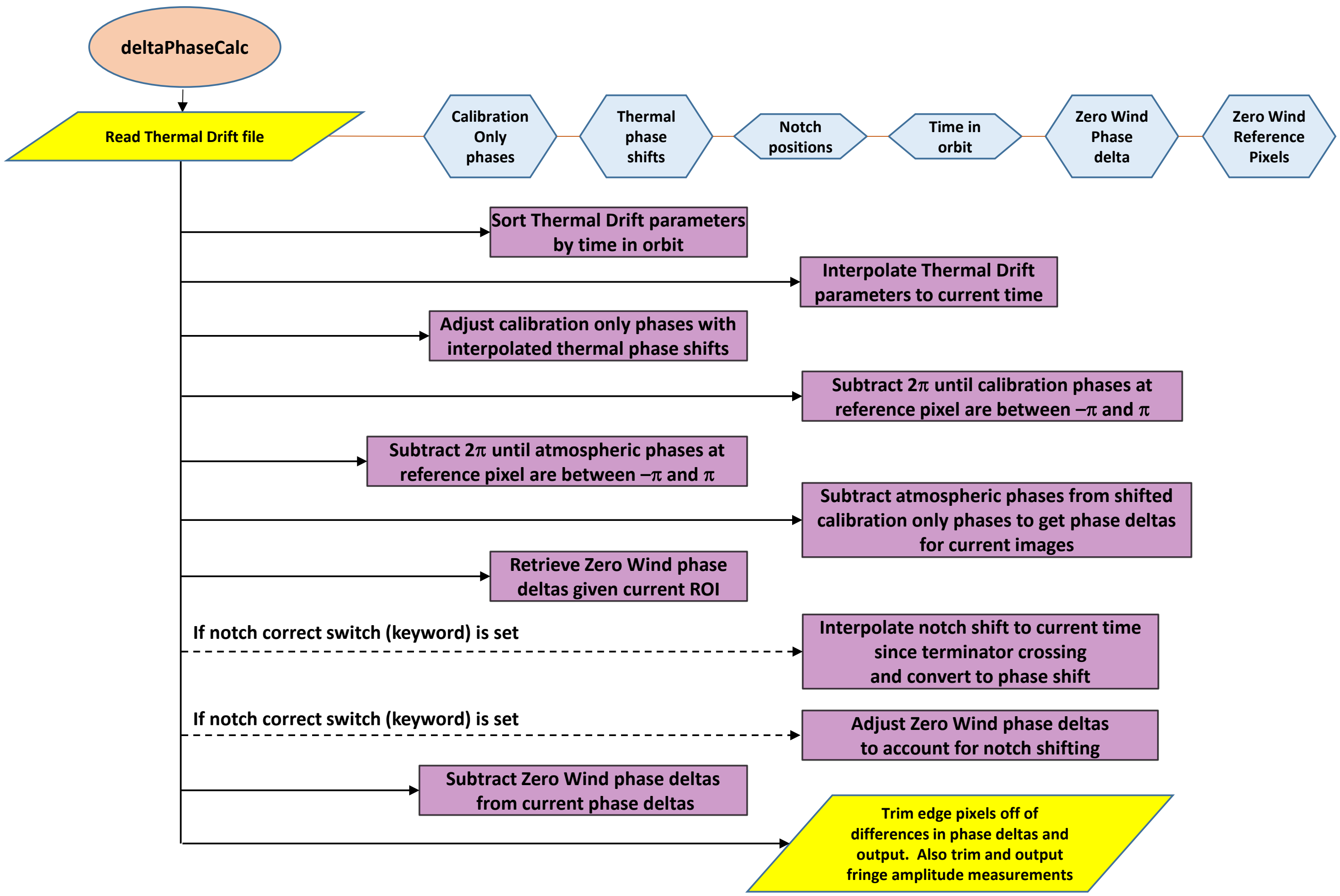
Cal Lamp Orbit Analysis Part II





Atmospheric Image Analysis
MIGHTI_L0P_to_L1
Read input file for parameters and images
Image paths
Ancillary path
Version
Revision
Switches
Loop over all images
Read image file
readNetCDF
Int Time
Image UTC
Sensor
ROI info
Restore sensor parameters
Call Tori's dark frame summon code
callDarks
Verify dark frame and average
verifyDarks
Account for effect of no image mask during frame transfer
pickUpRemove
Use cal only image from Thermal Drift file to remove cal lines from current image
removeCalLine
Correct for pixel to pixel variation with Oxygen Line flatfield
callFlatField
Correct for hot/dead pixels and streaks
spikeCorrect
Calculate average brightness by altitude for atmospheric lines
calcBright
Remove DC component and apodize signal
RemoveDC
Correct for phase distortion and fringe visibility along row
vis_and_phase_correct
Phase Calculation
From ancillary file, calculate ECEF look vectors for every pixel
calcECEF
Calculate phase delta between (inferred) calibration line and atmospheric line. See below.
deltaPhaseCalc
Loop
Save Science File
Save phase deltas by altitude and pixel. Save fringe amplitudes by altitude and pixel. Save IR brightnesses. Save ECEF vectors and roll angles.

Phase Delta Calculation Subroutine



8.0 IDL routine header information

The following pages contain the header information for each of the IDL routines developed for the L0 prime to L1 MIGHTI software. Included are a brief description and the input and output parameters for each routine.

8.1 addAttributesScience.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; addAttributesScience.pro - IDL procedure to create attributes
     pertaining to MIGHTI NetCDF files
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 1.1.6
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name           Action
;  ---------  ------------   -----------------------------------------
;  07 Jul 17  K. Marr          Began Code
;  11 Oct 17  J. Harlander     Reviewed and commented
;  05 Jan 18  K. Marr          Added a few flag variables
;  02 Feb 18  K. Marr          Removed Shuffle, Deflate Level and
                               Chunk(ing) Sizes.  These are set
                               in writeNetCDF
;  28 Mar 18  K. Marr          Added quality factor variable.
;  25 Apr 18  K. Marr          Removed MonoTon:False and edited
                               variable types to match variables.
;  07 May 18  K. Marr          Added conjugate maneuver register
;  17 Jul 18  K. Marr          Added brightness values
;  21 Aug 18  K. Marr          Improved Var Notes
;  14 Jan 19  K. Marr          Updated S/N flag attributes. Removed
                               near terminator flag.
;  30 Jan 19  K. Marr          Changed hyphen to underscore in
                               dimension names.  Var Notes for flags.
                               New variables
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Creates attributes for MIGHTI science NetCDF files
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;
; Required Input Parameters:
;
;           variableStruct: structure with variable information such
                            as size of IR, green and red regions
;           attStruct:      structure containing global attributes
;           sensor:         'A' or 'B'
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;
;           dataStructOut:  Composite output structure containing
                            global attributes, global dimensions, and
                            variable information
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.2 calcBright.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; calcBright.pro - IDL procedure to  determine the mean brightness
;                  of the red and green fringes and apply calibration
                   factor from the parameters file.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 1.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  -------------------------------------------
;
;  24 Jul 18       K. Marr  Started Coding
;  30 Jul 18  J. Harlander  Put in header and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;    Mean value of red and green fringes are calculated and a
     calibration factor read in from the parameters file is applied.
;    This is an attempt to carry the absolute brightness as determined
     from the DC signal value of red and green signals.
```

```
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;          Red and green brightnesses in R
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn: Input image to be analyzed
;           directory: path to working directory
;           Sensor:    'A' or 'B'
;           timeStr:   'Day' or 'Night'
;           redBotY:   Lowest row with red data
;           grnStart:  First green column
;           redStart:  Frist red column
;           grnStart:  First green column
;           redStart:  Frist red column
;           grnWidth:  Width of green fringe pattern (columns)
;           redWidth:  Width of red fringe pattern (columns)
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;          brightnessG:  Mean brightness of green by row used for DC
                         brightness estimate
;          brightnessR:  Mean brightness of red by row used for DC
                         brightness estimate
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.3 calcECEF_v4.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; calcECEF_v4.pro - IDL procedure to calculate ECEF vector locations
     for each pixel using polynomial fit detailed in the MIGHTI
     alignment document SSD-RPT-MI001
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 4.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
```

```
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  31 Mar 17  K. Marr         Began coding
;  17 Jan 18  J. Harlander    Reviewed and commented
;  21 Mar 18  K. Marr         Fixed 2x16 to 2x2 offsets.
;  01 Feb 19  K. Marr         Added local solar time calculation.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Determines ECEF vectors and latitude, longitdue and altitude of
     tangent point for each pixel in the MIGHIT FOV using the
     polynomial functions detailed in the MIGHTI alignment report SSD-
     RPT-MI001.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ancillaryStruct: structure containing ancillary file
            information
;           directory: directory containing data begin analyzed.  Must
            have Input, Output and Calibrations subdirectories
;           Sensor:  'A' or 'B'
;           timeStr:  'Day' or 'Night'
;           ancIndex:  ancillary index for file begin processed
;           redStart: column start for red fringes
;           grnStart: column start for green fringes
;           ROIStartY: lowest row of ROI
;           ROIStartX: lowest column of ROI
;           ROIHeight: maximum row of ROI
;
; Required external file:
;           Parameters file in Calibration subdirectory
;
; External functions or procedures called:
;           ECEF_to_wgs84
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:  All outputs are calculated for beginning, middle and end of
           integration.
;          Array format:  fltarr(x_pixel, y_pixel, 3, 3)
               where 3rd dimension contains vector data or Lat, Long,
               Alt and 4th dimension is time, beginning, middle, end
               of exposure
;
;           RedECEF:  ECEF for each red pixel
```

```
;            GreenECEF:  ECEF for each green pixel
;            greenLatLonAlt: Latitude, Longitude and Altitude for green
               pixel tangent point
;            redLatLonAlt: Latitude, Longitude and Altitude for red
               pixel tangent point
;            failFlag: 0 if successful, 5 if no parameters file
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.4 callFlatfield_v3.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; callFlatfield_v3.pro - IDL procedure to perform a flatfield
correction on MIGHTI A or B, on-orbit data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 3.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  -------------------------------------------
;  01 Jan 17  C. Englert    Started coding.
;  16 Nov 17  K. Marr       Changed to have NaNs instead of 0.
;  17 Jan 18  J. Harlander  Reviewed and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Reads the flatfield data that was created by
  Create_MIGHTI_Flatfields.pro
; and apply it to the measured data, depending on the
; sensor (A or B) and sensor mode (day or night).
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Data:
;            Flatfield data must be in Calibrations subdirectory of
             working directory:
;                    MIGHTI_A_DAY_Flatfield_Oxy.nc
;                    MIGHTI_A_NIGHT_Flatfield_Oxy.nc
;                    MIGHTI_B_DAY_Flatfield_Oxy.nc
;                    MIGHTI_B_NIGHT_Flatfield_Oxy.nc
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;          Flatfield corrected image data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn:    Input image to be analyzed
;           rowmin:     Starting row for analysis
;           colmin:     Starting column for analysis
;           Sensor:     'A' or 'B'
;           SensorMode: 'Day' or 'Night'
;           LineType:   'Oxy' or 'Cal'
;           directory:  path to working directory.  Must have Input,
Output and Calibrations                            subdirectories
;
; Required Input Data:
;           Flatfield data must be in Calibrations subdirectory of
            working directory:
;                   MIGHTI_A_DAY_Flatfield_Oxy.nc
;                   MIGHTI_A_NIGHT_Flatfield_Oxy.nc
;                   MIGHTI_B_DAY_Flatfield_Oxy.nc
;                   MIGHTI_B_NIGHT_Flatfield_Oxy.nc;
;
;imageIn, rowmin, colmin, Sensor, SensorMode, LineType,
FFCorrectedImage, directory, fileNameOut, failFlag
;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;          FFCorrectedImage:  Fringe image corrected flat field
           variations
;          fileNameOut:  Name of flat filed calibration file used for
           correction
;          FailFlag:  0 if successful, 5 if no flat field calibration
           file found
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.5 create_notch_position.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; create_notch_position.pro - IDL procedure to determine notch
     positions from the on-orbit calibration data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 1.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  22 Dec 17  C. Englert    First cut
;  04 JAN 18  C. Englert    added initial conditions for fake data and
SDL calibration data
;                           (SDL-cal data initial conditions are
commented out, but are
;                           expected to be viable for flight data.)
;  23 Jul 18 J. Harlander   Reviewed and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Routine processes one exposure.
; Depending on day or night mode, it uses
; the red or green side to find the notch position (shift). The notch
  position is determined by fitting an analytic function to a notch
  row.
;
; The initial conditions of the fit are highly important, due to the
  periodic nature of the analytic function that is fitted. If the
  initial conditions are not close to the correct solution, the fit
  will find a local minimum in Chi^2.
; If in doubt, the visual inspection of the data and fitted function
  makes it obvious whether the correct minimum is found or not.
; On orbit, any thermal changes are expected to be much too small to
  have any effect on the proper initial conditions, so that the
  predetermined initial conditions from the ground based calibration
  are expected to be robust.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn:   Input image to be analyzed
;           timeStr:   'Day' or 'Night'
;           Sensor:    'A' or 'B'
```

```
;           krOn:      Krypton lamp status
;           NeOn:      Neon lamp status
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;
;           A: Fit parameters as follows
;              A[0] = "phase offset" of the notch pattern
;              A[1] = phase shift of interferogram outside the notches
;              A[2] = scale factor for DC offset of interferogram
                     outside the notches
;              A[3] = scale factor for amplitude of interferogram
                     outside the notches
;              A[4] = phase shift of interferogram inside the notches
;              A[5] = scale factor for DC offset of interferogram
                     inside the notches
;              A[6] = scale factor for amplitude of interferogram
                     inside the notches
;           Notch position: A[0] parameter above
;           Failflag: As follows: (0=good, 1=frame not suitable for
                     notch drift determination, 2=notch drift
                     determination was attempted, but failed)
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.6 deltaPhaseCalc_v2.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; deltaPhaseCalc_v2.pro - IDL procedure to correct phases for thermal
;    drift and notch shift
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;codeVersion = '2.0.2'
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  01 Jan 18  K. Marr       Started coding.
;  12 Jul 18  J. Harlander  Reviewed and commented
;  28 Aug 18  K. Marr       Variable bug fixed.
;  01 Jan 19  K. Marr       Added ThermEpoch output.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Corrects phase data for thermal drift and notch shift. Also trims
;    phase and envelope images
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;
;           pathName       path to calibration data containing thermal
                           drift calibration file
;           timeInOrbit    time since most recent day/night terminator
                           crossing
;           redPhasesIn    red phase image
;           grnPhasesIn    green phase image
;           redEnvIn       red envelope image
;           grnEnvIn       green envelope image
;           grnStart       start column for green images
;           redStart       start column for red images
;           grnWidth       width of green image before trimming
;           redWidth       width of red image before trimming
;           trim           number of pixels to be trimmed from each
                           edge of image
;           timeStr        'Day' or 'Night'
;           redNeFreqDiff  frequency difference between red and Ne
                           lines - used in notch correction
;           grnKrFreqDiff  frequency difference between green and Kr
                           lines - used in notch correction
;           sensor         'A' or 'B'
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;           redDeltaPhasesOut corrected red phase image
;           grnDeltaPhasesOut corrected green phase image
;           redEnvOut         trimmed red envelope image
;           grnEnvOut         trimmed green envelope image
;           refPixelG         green reference pixel - read from
                              thermal drift calibration file
;           refPixelR         red reference pixel - read from thermal
                              drift calibration file
;           thermEpoch        time of first exposure in thermal drift
                              file.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  External routines called
;
;               readNetCDF
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.7 MIGHTI_Create_Thermal_Drift_v01r00.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; MIGHTI_Create_Thermal_Drift_v01r00.pro - IDL procedure to calculate
thermal drift calibration file
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;codeVersion = '1.0.3'
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  19 May 17  K. Marr       Started coding.
;  21 Feb 18  K. Marr       Switched to curvefit way of determining
                            phase shift
;  23 Mar 18  K. Marr       Swapped pickup remove to before anything
                            else.
;  26 Mar 18  K. Marr       Fixed error where calOnly Phases weren't
                            being saved.
;  01 May 18  K. Marr       Fixed cal Orbit guess issue.
;  03 May 18  K. Marr       Added code to skip any files taken during
                            non-science manuevers
;  08 May 18  K. Marr       Fixed non-orbit errors (too few files,
                            etc)
;  04 Jun 18  K. Marr       Update for spikeCorrect v8.
;  05 Jul 18  K. Marr       Added calibration file names in output
                            file
;  10 Aug 18  K. Marr       Changed how wrong ancillary is handled.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Required Input Data:
;           Text file containing file names and other information for
            the L0P files
;
;           Input text file has the following format:
;
;           Revision=944,944
;           Version=006
;           directory=C:\Users\John\Documents\research\proposals\ICON\
;           Ancillary=
            ICON_L0P_MIGHTI-A_Ancillary_2017-05-29_v01r002.nc
;           GPI=ICON_Ancillary_GPI_2016-335_v01r001.nc
```

```
;
;           [Files]
;           ICON_L0P_MIGHTI-A_NOMINAL_2017-05-29_013817_v01r944.NC
;           etc....
;
;           where
;
;           Revision indicates the revision of the output L1 file.
            There should be at least as many comma separated revision
            numbers as there are input L0P files.
;           Version indicates the version of the output L1 file.  Only
            one version number is used
;           Directory is the directory location for the L0P files and
            other information as follows
;              Note:  This directory must have subfolders
;              Input: Contains the L0P file data and associated
                      ancillary file (darks?)
;              Output: Subdirectory for output L1 files
;              Calibration:  Contains calibration files including
                       flatfield, visibility correction phase
                       distortion correction, thermal drift
                       and parameters files
;              Tohban: Subdirectory for output Tohban plots (if used)
;
;           Ancillary is the name of the ancillary file associated
            with the L0P files.  The file must be located in the Input
            subdirectory.
;           GPI is the name of the GPI file associated with the L0P
            files.  It is currently not used.
;           Files is a list of the filenames of the L0P files to be
            analyzed.
;
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           Input text file as described above
;           Variable for error codes
;
; Required Input and Calibration Data:
;           Calibrations subdirectory of working directory must
            contain the following files:
;                   MIGHTI_X_DAY_Flatfield_Oxy.nc
;                   MIGHTI_X_NIGHT_Flatfield_Oxy.nc
;                   MIGHTI_X_DAY_Flatfield_Cal.nc
;                   MIGHTI_X_NIGHT_Flatfield_Cal.nc
;                   ICON_L1_MIGHTI-X_Calibration-Instrument-
                    Parameters_YYYY-MM-DD_vVVrRRR
;                   ICON_L1_MIGHTI-X_Calibration-Phase-Night.nc
;                   ICON_L1_MIGHTI-X_Calibration-Phase-Day.nc
;                   ICON_L1_MIGHTI-X_Calibration-Visibility-Night.nc
```

```
;                    ICON_L1_MIGHTI-X_Calibration-Visibility-Day.nc
;
;             where X is 'A' or 'B', YYYY-MM-DD is year, month, date,
             VV is version number, RRR is revision number
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;           A Thermal Drift cal file. The file is placed in the Output
           subdirectory of the working directory. The filename has
           format:
;
;  ICON_L1_MIGHTI-X_Calibration-Thermal-Drift_YYYY-MM-DD_vVVrRRR.NC
;
;
;           Error Codes returned
;              0: Successful
;              1: Error reading input file
;              2: Input parameter error from input file or ancillary
                 missing
;              3: Not enough files for analysis or no Calibration Only
                 files
;              4: No sensor indicated
;              5: No calibration file (phase, visibility, inst.
                 parameter, flatfield)
;              6: Bad gain register
;              7: No Zero Wind file
;             8+: writeNetCDF errors
;             16: No good darks
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  External routines called
;
;                readNetCDF
;                verifyDarks_v6
;                removeCalLine
;                callFlatField_v3
;                spikecorrect_v8
;                pickupRemove_v4
;                removeDC_v6
;                vis_and_phase_correction_v5
;                calcECEF_v3
;                thermal_drift_correction_v5
;                addAttributesScience
;                writeNetCDF_km
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.8 MIGHTI_L0P_to_L1_v01r00.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; MIGHTI_L0P_to_L1_Launch.pro - IDL procedure generate level 1 (L1)
     files from level 0 prime (L0P) files.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; codeVersion = '1.0.9'
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  -------------------------------------------
;  01 Jan 17  K. Marr       Started coding.
;  16 Nov 17  K. Marr       Changed to have NaNs instead of 0.
;  16 Feb 18  J. Harlander  Reviewed and commented
;  14 Mar 18  K. Marr       Added deltaPhaseCalc changes.  Reduced
                            size of calcECEF variables.
;  23 Mar 18  K. Marr       Swapped pickup remove to before anything
                            else.
;  28 Mar 18  K. Marr       Added quality factor
;  30 Mar 18  K. Marr       Moved uncertainty calculation to after
                            delta calculation
;  10 Apr 18  K. Marr       Fixed trimming concerns
;  11 Apr 18  K. Marr       Increased trimming of uncertainty
                            calculation to 100 pixels and added
                            ability to set degree of poly fit.
;  16 Apr 18  J. Harlander  Modified low signal threshold to better
                            catch signalless altitudes.
;  23 Apr 18  K. Marr       Version increment for push to UCB.
                            Increased nRange minimum to 300.
;  25 Apr 18  K. Marr       Edited variable types.
;  27 Apr 18  K. Marr       Added check for closed atmospheric
                            aperture
;  01 May 18  K. Marr       Fixed error created if no
                            phases/amplitudes exist for a color
;  08 May 18  K. Marr       Added notch correct variables to
                            deltaPhaseCalc from params file.
;  10 May 18  K. Marr       Updates to deltaPhase routine,
                            uncertainty, skip unknown A1 files.
                            Set for Launch
;  31 May 18  K. Marr       Adjusted IRStarts from parameter files to
                            account for ROI trim. IRTopY added to
                            spikeCorrect
;  08 Jun 18  K. Marr       Added orbit number as a global parameter.
;  09 Jul 18  K. Marr       Removed factor of sqrt(2) in uncertainty
                            estimates
```

```
;  17 Jul 18  K. Marr       Added brightness profile output and v2 of
                            removeCalLine
;  10 Aug 18  K. Marr       Changed how wrong ancillary is handled.
;  27 Aug 18  K. Marr       Fixed calibration file names for
                            attribute.  Renamed file.  Minor
                            changes won't update revision value now.
;  20 Dec 18  K. Marr       Added thermal drift file to calibration
                            file output.
;  01 Jan 19  K. Marr       Improved quality flags.  Deleted
                            terminator flag.  Changed analysis
                            threshold.
;  14 Jan 19  K. Marr       Removed apodization of envelopes before
                            saving.
;  30 Jan 19  K. Marr       Added IR support data. Condensed into
                            IRCalc code.
;  08 Feb 19  K. Marr       Modified thresholding for analysis.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Reads file containing filenames of L0P files and outputs L1 files
     for the MIGHTI instrument
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Data:
;           Text file containing file names and other information for
            the L0P files
;
;
;           Input text file has the following format:
;
;           Revision=944,944
;           Version=011
;           useCalLines = 0
;           doNotchCorrect = 0
;           justPass = 1
;           useSpecificDark = 1
;           directory=C:\Users\John\Documents\research\proposals\ICON\
;           Ancillary=
                ICON_L0P_MIGHTI-A_Ancillary_2017-05-29_v01r002.nc
;           GPI=ICON_Ancillary_GPI_2016-335_v01r001.nc

;           [Files]
;           ICON_L0P_MIGHTI-A_NOMINAL_2017-05-29_013817_v01r944.NC
;           etc....
;
;           where
;
```

```
;           Revision indicates the revision of the output L1 file.
            There should be at least as many comma separated revision
            numbers as there are input L0P files
;           Version indicates the version of the output L1 file.  Only
            one version number is used
;           Directory is the directory location for the L0P files and
            other information as follows
;              Note:  This directory must have subfolders
;                   Input:  Contains the L0P file data and associated
                            ancillary file (darks?)
;                   Output: Subdirectory for output L1 files
;                   Calibration:  Contains calibration files including
                             flatfield, visibility correction phase
                            distortion correction, thermal drift and
                            parameters files
;                   Tohban: Subdirectory for output Tohban plots (if
                           used)
;           Ancillary is the name of the ancillary file associated
            with the L0P files.  The file must be located in the Input
            subdirectory
;           GPI is the name of the GPI file associated with the L0P
            files.  It is currently not used
;           Files is a list of the filenames of the L0P files to be
            analyzed
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;          L1 data product that includes amplitude and phase data for
           wind analysis, A-band data for temperature analysis and
           much of the L0P data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           Input text file as described above
;           Variable for error codes
;
; Required Input and Calibration Data:
;           Calibrations subdirectory of working directory must
            contain the following files:
;                   ICON_L1_MIGHTI-X_Calibration-Thermal-Drift_YYYY-
                     MM-DD_vVVrRRR.NC
;                   MIGHTI_X_DAY_Flatfield_Oxy.nc
;                   MIGHTI_X_NIGHT_Flatfield_Oxy.nc
;                   MIGHTI_X_DAY_Flatfield_Cal.nc
;                   MIGHTI_X_NIGHT_Flatfield_Cal.nc
;                   ICON_L1_MIGHTI-X_Calibration-Instrument-
                      Parameters_YYYY-MM-DD_vVVrRRR
```

```
;                   ICON_L1_MIGHTI-X_Calibration-Phase-Night.nc
;                   ICON_L1_MIGHTI-X_Calibration-Phase-Day.nc
;                   ICON_L1_MIGHTI-X_Calibration-Visibility-Night.nc
;                   ICON_L1_MIGHTI-X_Calibration-Visibility-Day.nc
;
;            where X is 'A' or 'B', YYYY-MM-DD is year, month, date,
            VV is version number, RRR is revision number
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;          An L1 file for each L0P input file. The file is placed in
           the Output subdirectory of the working directory. The
           filename has format:
;
;             ICON_L1_MIGHTI-X_SCIENCE_YYYY-MM-DD_TTTTTT_vVVrRRR.NC
;
;          where TTTTTT is the time stamp of the L0P file (see other
           conventions above)
;
;          Error Codes returned
;             0: Successful
;             1: Error reading input file
;             2: Input parameter error from input file or ancillary
                missing
;             3: No files for analysis
;             4: No sensor indicated
;             5: No calibration file (phase, visibility, inst.
                parameter, flatfield)
;             6: Bad gain register
;             7: No thermal file
;            8+: writeNetCDF errors
;            16: No good darks
;           999: Skip file but don't stop analysis of other files
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  External routines called
;
;               readNetCDF
;               verifyDarks_v6
;               removeCalLine
;               callFlatField_v3
;               spikecorrect_v7
;               pickupRemove_v4
;               removeDC_v6
;               vis_and_phase_correction_v5
;               calcECEF_v3
;               thermal_drift_correction_v5
;               addAttributesScience
;               writeNetCDF_km
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.9 pickupRemove_v4.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; pickupRemove_v4.pro - IDL procedure to correct images for signal
collected during
; image transfer.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 4.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  01 Mar 17  K. Marr       Started coding.
;  21 Nov 17  K. Marr       Handle NaNs.
;  16 Jan 18  J. Harlander  Reviewed and commented.
;  08 Mar 18  K. Marr       Switched to division method of removing
                            pickup.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Input image is corrected for signal pickup during image transfer
; Correction is done along rows (the direction of readout)
; Value of correction is determined by transfer time (307.5 ms),
     integration time and number of pixels.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;          imageIn:  Image to be corrected
;          intItme:  integration time of image in (msec)
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output variables:
;          ImageOut:  Pickup corrected image
;
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.10 removeCalLine_v2.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; removeCalLine_v2.pro - IDL procedure to remove calibration fringes
;     from images containing both atmospheric and calibration signals
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;codeVersion = '2.0.0'
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  01 Jan 18  K. Marr       Started coding.
;  28 Jun 18  J. Harlander  Narrowed fft window in scaling to reduce
                             edge effects, fixed thermal shift
                             correction
;  02 Jul 18  J. Harlander  Reviewed and commented
;  01 ??? 18  K. Marr       Added Day/Night flexibility
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Subtracts calibration fringes from images containing both
     atmospheric and calibration signals
; Calibration fringes are obtained from cal-only image and are
     adjusted for thermal drift and brightness change (if any) before
     subtraction.  Red and green regions are handled through separated
     calls to this routine.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;
;           imageIn       image to be corrected
;           thermalPath   path to thermal drift file
;           xStart        starting column for removal
;           xWidth        width in column for removal
;           yBot          lowest row (altitude) for removal
;           yTop          highest row for removal
;           timeStr       string indicating Day or Night
```

```
;           timeInOrbit   time since last day/night terminator
            crossing - used for thermal drift
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;            imageOut      return image
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  External routines called
;
;               readNetCDF
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.11 removeDC_v6.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; RemoveDC_v6.pro - IDL procedure to remove non-modulated part of
;     interferogram from fringe data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 6.1.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ------------------------------------------
;  31 Mar 17  K. Marr         Began coding
;  19 Oct 17  J. Harlander    Reviewed and commented
;  27 Oct 17  K. Marr         Added pixel size adjustment for lowest
                               green rows
;  07 Nov 17  K. Marr         Upgraded adjustment for rounded corners
;  21 Nov 17  K. Marr         Handle incoming NaNs
;  12 Dec 17  K. Marr         Returned the IR array into the image,
                               separated red and Green
;  16 Jan 18  J. Harlander    Reviewed and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
```

```
; Removes non-modulated part from fringe data using high pass filter
      in spectral domain.  Currently filters out lowest spatial
      frequencies from -3,-2,...,0,...2,3 where frequency 0 is the DC
      term.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn: Input image
;           Sensor:  'A' or 'B'
;           redStart: column start for red fringes
;           redWidth: width of red fringes
;           grnStart: column start for green fringes
;           grnWidth: width of green fringes
;           imageBoty: lowest row green fringes
;           redBoty: lowest row red fringes
;           imageTopY: top row with image data
;           IRTopY: top row of IR image
;
; Logical input flags:
;           hasAtmoLines:  If atmospheric lines are present
;           hasCalLines:   If calibration lines are present
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;
;           imageOut:  ImageIn with non-modulated part subtracted
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.12 spikeCorrect_v8.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; SpikeCorrect_v8.pro - IDL procedure to perform a spike correction
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 8.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  ----------------------------------------
;  12 Apr 18      K. Marr   Fixed median brightness calculation issue.
;  31 May 18      K. Marr   Adjusted IR spike analysis
;  17 Jul 18      K. Marr   Added output for brightness.
```

```
;  24 Jul 18 J. Harlander   Reviewed and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;  Corrects raw data for spikes using different methods for fringe
     data and IR data.
;
;  Fringe Data:  Normalize altitudes with brightness of row.
;                Add constant to offset from zero.
;                Calculate 5 pixel median along columns.
;                Identify spikes from difference between column and
                 median column using threshold.
;                Replace spikes with median value.
;                Median value at top and bottom two rows are set to
                 the median value third from top and third row.
;                Remove offset and normalization.
;
;  IR Data:      Normalize with IR flat field.
;                Find 3x3 2D median image.
;                Identify spikes from difference between image and
                 median image using threshold.
;                Replace spikes with median value.
;                Median value at top and bottom two rows are set to
                 the 1D 3 pixel median along row.
;                Remove offset and normalization.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Data:
;           IR Flatfield data must be in Calibrations subdirectory of
            working directory:
;                   ICON_L1_MIGHTI-A_Calibration-Flatfield_Day-IR.nc
;                   ICON_L1_MIGHTI-A_Calibration-Flatfield_Night-IR.nc
;                   ICON_L1_MIGHTI-B_Calibration-Flatfield_Day-IR.nc
;                   ICON_L1_MIGHTI-B_Calibration-Flatfield_Night-IR.nc
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Output:
;          Spike corrected image data.  DC value of interferogram
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn: Input image to be analyzed
```

```
;            imageBotY: Lowest row with data (green and IR)
;            redBotY:   Lowest row with red data
;            grnStart:  First green column
;            redStart:  Frist red column
;            grnWidth:  Width of green fringe pattern (columns)
;            redWidth:  Width of red fringe pattern (columns)
;            imageTopY: Highest row with data
;            IRStarts:  Starting columns for eacn IR signal region
;            IRWidths:  Width of each IR signal region
;            IRTopY:    Highest row for IR data
;            directory: path to working directory
;            Sensor:    'A' or 'B'
;            timeStr:   'Day' or 'Night'
;            ROIStartX: first column for region of interest
;            ROIStartY: first row for reqion of interest
;            ROIWidth:  number of columns in region of interest
;            ROIHeight: number of rows in reqgion of interest
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;           imageOut:     Spike corrected image
;           brightnessG:  Median brightness of green by row used for DC
brightness estimate
;           brightnessR:  Median brightness of red by row used for DC
brightness estimate
;           brightnessIR: Median brightness of IR by row after
flatfielding
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.13 verifyDarks.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
;   pro verifyDarks
;
;  Inputs:
;            darkNames - An array of strings with the path to dark
             frames to be verified
;            imageTime -
;            intTime -
;            sensor - A single character string indicating either
             MIGHTI A or B
;
; Outputs:
;             avgDark - an average of all verified dark images
;             failflag - an indicator in the case that all dark images
              are unusable
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
;
; Version : 6.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  -------------------------------------------
;  13 Jun 17  K. Marr         Stub created
;  21 Jun 17  J. Harlander    Added average and stddev checks,
                              aperture 1 position check,
                              calibration lamp check, Epoch check,
                              integration time check, sensor check,
                              TEC cold temperature check, Forward/Rear
                              bench temperature check
;  08 Jan 18  K. Marr         Changed averaging method to exclude
                              temporary spiked
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

8.14 vis_and_phase_correction.pro

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; vis_and_phase_correction.pro - IDL procedure to perform a visibility
;     and phase correction on MIGHTI A or B, on-orbit data
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Version : 5.0.0
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; History:
;
;  Date       Name          Action
;  ---------  ------------  -------------------------------------------
;  16 Jan 17  J. Harlander    Started coding.
;  02 Feb 17  J. Harlander    Added code to correct phase distortion
;  07 Nov 17  K. Marr         Removed zeroing of non-numbers
;  17 Jan 18  J. Harlander    Reviewed and commented
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Program Description
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Reads the visibility and phase distortion data that was created by
```

```
; Create_MIGHTI_vis_and_phase_correction.pro and apply it to the
     measured data, depending on the sensor (A or B) and sensor mode
     (day or night).
; Visibility correction done by dividing by visibility calibration
     file.
; Phase distortion correction done by multipliying interferogram by
; exp(-i*phase_dist) where phase_dist is obtained from the phase
     distortion calibration file.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Required Input Parameters:
;           ImageIn: Input image to be analyzed
;           startX:  Starting column for analysis
;           startY:  Starting row for analysis
;           Sensor:  'A' or 'B'
;           timeStr:  'Day' or 'Night'
;           directory:  path to working directory.  Must have Input,
            Output and Calibrations subdirectories
;
; Required external calibration files assumed to be in Calibrations
     subdirectory of working folder:
;           Visibility data:
;                   MIGHTI_A_DAY_Visibility.nc
;                   MIGHTI_A_NIGHT_Visibility.nc
;                   MIGHTI_B_DAY_Visibility.nc
;                   MIGHTI_B_NIGHT_Visibility.nc
;           Phase Distortion data:
;                   MIGHTI_A_DAY_Phase.nc
;                   MIGHTI_A_NIGHT_Phase.nc
;                   MIGHTI_B_DAY_Phase.nc
;                   MIGHTI_B_NIGHT_Phase.nc
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Outputs:
;          CorrectedImage:  Fringe image corrected for visibility and
                            phase distortions
;          fileNameOutVis:  Name of visibility calibration file used
                            for corrections
;          fileNameOutPhase:  Name of phase distortion calibration
                            used for correction
;          FailFlag:  0 if successful, 5 if no visibility or phase
                            correction calibration files found
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

9.0 Acknowledgements

The MIGHTI team would like to acknowledge the support and contributions of Aidan Thayer, Logan Storm, Brian Harding, Mike Stevens, Tori Fae, Scott England, Colin Triplett, and the rest of the ICON team. This work was supported by NASA grants NNG12FA45C and NNG12FA42I.