MIPS User Documentation

Version 1.1

Date:

14 October 2011

Authors:

Liam Gumley - Liam.Gumley@ssec.wisc.edu
Amato Evan
Willem Marais - Willem.Marais@ssec.wisc.edu

NSMC Contract # FY3-MAS-1001

Cooperative Institute for Meteorological Satellite Studies (CIMSS)
Space Science and Engineering Center (SSEC)
University of Wisconsin-Madison
USA

0. Document revision changes

Changes had been made to this document to reflect the updated version of MIPS. The changes that had been made for each section will now be described:
1.4 - The output resolutions of the polar mosaic output images that are produced by MIPS were modified. This section was updated to reflect the changes.
1.5 - The output resolution of the global mosaic output images were modified. This section was updated to reflect the changes.
4 - The MIPS package has now precompiled binaries included for the x86 32-bit and 64-bit systems. The compressed tar file names were added to this section.
5.2 - Instructions of how to install the precompiled binaries were added to this section.

In the following sections the MIPS version number was updated:
5., 9.1, 11., 12., A.1.

From section 6.6 to section 6.10 new MIPS configuration values were added:
Section 6.6 describes the new configuration details that were added in response of memo (18-08-2011) item number two:

```
Mosaic polar composite image can be generated in 250m resolution in
the range of 60 (or -60) latitude above.
```

Section 6.7 describes the new configuration value that had been added which allows the user to set the output resolution of the global mosaic images.

Section 6.8 describes the new configuration details that were added in response of memo (18-08-2011) item number one:

```
Polar tiles need add GeoTIFF format file
```

Sections 6.9 and 6.10 describe the new configuration details that were added in response of the memo (18-08-2011) item number three:

```
The curve of false color enhancement can be changed by user in case
when the color saturated
```

1. Introduction

MIPS is a Multi-Sensor Image Processing System, it can process both MERSI and MODIS L1B granules or overpasses. MIPS creates several types of images which include L1B granule images, subset images, global KML files, polar mosaic images and global mosaic images. For all the mentioned image types, MIPS has the capability to create true and false color images.

To process MERSI data MIPS requires the 1000m and 250m L1B granule files. To process MODIS data MIPS requires the 1KM, HKM, QKM and the geolocation L1B granule files.

The different types of images MIPS can produce, will be described in the following subsections.

1.1 L1B granule images

For each L1B granule MIPS process, it creates an unprojected 1000m resolution true color image. This type of image is a representation of what data are in the L1B granule.

1.2 Subset images

A subset image is a user defined geographical region over which an image is created. For example if a user wants MIPS to create a daily image - per satellite and sensor - of a city or province, MIPS can be configured to create such a subset which covers the desired geographical area of a city or province. The subset images have a pixel resolution of 250m and are produced in both GeoTIFF and JPEG file formats. The projection that is used is known as “equirectangular projection”. Subset images will be created for both true and false color images.

1.3 KML images

For all the processed data, MIPS creates 250m resolution true and false color image tiles for the global KML files. To to view the KML files please download the Google Earth program from the following website: http://www.google.com/earth/index.html. The KML can be served over an internal network or over the internet. In both cases, the MIPS KML product directory must be made accessible through a web-server.

1.4 Polar mosaic images

MIPS can create polar mosaic true and false color images.MIPS creates 250m polar mosaic images, and the area that the mosaic image cover can be configured. MIPS can also be configured to create either or both 1000m and 4000m resolution polar mosaic images. The 250m images are created in both GeoTIFF and JPEG formats. The other resolution images will only by created in JPEG format. The projection that is used for the polar images is known as “polar stereographic projection”.

1.5 Global mosaic images

MIPS can also create global mosaic true and false color images of all the L1B granules that had been processed. The pixel resolution of the global mosaic image can be configured - per default it is configured as 1000m, and the images are available in GeoTIFF and JPEG file formats.

2. Disclaimer

The University of Wisconsin-Madison Space Science and Engineering Center (SSEC) makes no warranty of any kind with regard to the MIPS software or any accompanying documentation, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. SSEC does not indemnify any infringement of copyright, patent, or trademark through the use or modification of this software.

There is no expressed or implied warranty made to anyone as to the suitability of this software for any purpose. All risk of use is assumed by the user. Users agree not to hold SSEC, the University of Wisconsin-Madison, or any of its employees or assigns liable for any consequences resulting from the use of the MIPS software.

3. System requirements

3.1 Hardware requirements

MIPS requires the following to be able to process either global MERSI or MODIS data:
- At least eight 2.5GHz CPU cores
- 24 GB of main memory
- 200 GB of disk space

Note: It is recommended to run MIPS on a 64-bit Intel or AMD based system.

3.2 Software requirements

MIPS was designed to be executed on a Linux variant OS (Operating System). Basic packages such as ‘coreutils’, ‘flex’ and ‘bison’ must be installed on the host OS. MIPS also requires the image software package ImageMagick, at least version 6.4 or later.

Note: MIPS was ***not*** designed to to be executed on a Windows OS.

4. Package content

The MIPS package, mips_version1.1_package, includes the following files:

DISCLAIMER
A text file with the MIPS software disclaimer inside of it.

00README_MIPS.txt
The MIPS user documentation in text format.

MIPS-1.1_User_Documentation.pdf
The MIPS user documentation in PDF format.

mips-1.1.tar.gz
The MIPS software, which includes the 32-bit and 64-bit FL IDL interpreters. It also includes 32-bit executable binaries.

mips-1.1_compile_binaries
The source code of the additional software, that MIPS depends on, are in this directory. Instruction of how to compile the software will be given in the installation instructions.

mips_1.1_precompiled_binaries_64bit.tar.gz
The precompiled binaries of the additional software, that MIPS depend on, are in this compressed file. These binaries were compiled on a 64-bit Ubuntu 10.04.03 Linux distribution.

mips_1.1_precompiled_binaries_32bit.tar.gz
The precompiled binaries of the additional software, that MIPS depend on, are in this compressed file. These binaries were compiled on a 32-bit Ubuntu 10.04.03 Linux distribution.

ImageMagick-6.7.1-2.tar.bz2
This file contains the source code of ImageMagick. Instructions of how to compile and install ImageMagick is given in the installation instructions.

The MIPS package also includes eight files which contains L1B granules that are to be used for testing MIPS. An explanation of how to use these files is given in section 11. The eight file names are:

mips-1.1_FY3A_nonpolar_test_data.tar.gz
mips-1.1_FY3A_polar_test_data.tar.gz
mips-1.1_FY3B_nonpolar_test_data.tar.gz
mips-1.1_FY3B_polar_test_data.tar.gz
mips-1.1_TERRA_nonpolar_test_data.tar.gz
mips-1.1_TERRA_polar_test_data.tar.gz
mips-1.1_AQUA_nonpolar_test_data.tar.gz
mips-1.1_AQUA_polar_test_data.tar.gz

5. Installation

5.1 Installing ImageMagick

To see if ImageMagick is already installed on the host system, execute the following command in a shell terminal:

```
[user@host]$ convert -version
```

The output that is printed to the shell terminal, should be something similar to this:

```
Version: ImageMagick 6.5.4-7 2010-02-26 Q16 OpenMP http://
www.imagemagick.org
Copyright: Copyright (C) 1999-2009 ImageMagick Studio LLC
```

If ImageMagick is not installed on the host system and the host OS is Ubuntu, the following command can be executed on in the shell terminal:

```
[user@host]$ sudo apt-get install imagemagick
```

If ImageMagick cannot be installed through the OS's package manager, ImageMagick must then be manually configured, compiled and installed. Refer to appendix section A.2 for such instructions.

5.2 Installing MIPS

To install MIPS the file mips-1.1.tar.gz has to be untarred. When this file is untarred in an empty directory, the empty directory will have a new sub-directory called mips.version.1.1.

Note: Throughout this document it is assumed that the MIPS package, mips_version1.1_package, is located under the user's home directory. It is also assumed that MIPS will be installed under the user's home directory.

To untar the file mips-1.1.tar.gz, execute the following command:

```
[user@host]$ cd $HOME
[user@host]$ tar xzf $HOME/mips_version1.1_package/mips-1.1.tar.gz
```

Next the software binaries MIPS depends on must be unpacked. If MIPS is installed on a 64-bit system, the `mips_1.1_precompiled_binaries_64bit.tar.gz` must be unpacked:

```
[user@host]$ tar xzf $HOME/mips_version1.1_package/
mips_1.1_precompiled_binaries_64bit.tar.gz $HOME/mips.version.1.1/bin
```

Otherwise if the system is a 32-bit system, the `mips_1.1_precompiled_binaries_32bit.tar.gz` must be unpacked:

```
[user@host]$ tar xzf $HOME/mips_version1.1_package/
mips_1.1_precompiled_binaries_32bit.tar.gz $HOME/mips.version.1.1/bin
```

If none of the precompiled binaries can be used, the software binaries have to be compiled on the host system. Refer to section A.1 for instruction to compile the binaries.

6. Configuring MIPS

MIPS requires that a BASH script file to be imported before it can process L1B granule from a terminal or a shell script. The BASH script file defines all the necessary environmental variables for MIPS. This file will be referred as the MIPS configuration file from hence forth.

Each sensor, MERSI and MODIS, has it's own configuration file. For the following examples it is assumed that MIPS was installed under the user's home directory:

```
[user@host]$ export MIPS_HOME=$HOME/mips.version.1.1
```

The configuration files for MERSI and MODIS are located at the following paths:

```
$MIPS_HOME/scripts/mips_mersi_env.bash
$MIPS_HOME/scripts/mips_modis_env.bash
```

Note: The environmental variable MIPS_HOME must always be defined as a **full** filesystem path. In other words, the first character of the path must start with the character '/'. And the path must end with mips.version.1.

The following sections explain what necessary configuration value changes have to be made in the configuration files before MIPS can process L1B granules. Use an editor such as vim, vi, emacs, joe or nano to edit the configuration files.

6.1 Setting image output directories

First the configuration variables for the output directories must be set where all the images will be archived. For each sensor's configuration file the values for these variables **must** be unique. MIPS will not function correctly if same output directory is used for some or all the image products.

The first two variables that have to be set, are the tile and polar tile archive directories. The tile images are used to create the subsets, KML files, polar mosaic images and global mosaic images. The two variables are:

```
export MIPS_TILE_DIR=$MIPS_HOME/tiles
export MIPS_POLR_TILE_DIR=$MIPS_HOME/polartiles
```

Next the subsets, KML and mosaic (polar and global) output directories have to be set. The output directories are set by the following three variables:

```
export MIPS_SUBSET_DIR=$MIPS_HOME/subsets
export MIPS_KML_DIR=$MIPS_HOME/kml
export MIPS_MOSAIC_DIR=$MIPS_HOME/mosaic
```

Lastly the directory where the unprojected L1B granule images are archived must also be set. The variable for this is:

```
export MIPS_GRAN_IMG_DIR=$MIPS_HOME/granuleimages
```

6.2 Setting KML preferences

If the KML files are going to be served on a web server, the HTTP address where from the KML files are going to be accessed needs to be set. The HTTP address must be equal to the KML product directory path. For example, if the URL to the server name is 'http://servername.com':

```
export MIPS_KML_URL="http://servername.com/products/modis/kml/
```

Note: The KML URL **must** always end with a forward slash ('/') character.

Next the variable that must be set is the geographical location where Google Earth must center the globe when a KML file is loaded. The name of the geographical site where upon Google Earth will center itself must also be set. Here is an example where these three variables are set:

```
export MIPS_SITE_NAME="BEIJING"
export MIPS_SITE_LAT="39.913889"
export MIPS_SITE_LON="116.391667"
```

Note: The site name **must** all be in capital letters.

6.3 Setting the IDL interpreter

MIPS requires an IDL interpreter. Either the commercial IDL package can be used, or a freeware version of IDL can be used. The name of the freeware version of IDL is Fawlty. The Fawlty binaries are included in the MIPS 'mips-1.1.tar.gz' file.

If the host system is a 32-bit system, the 32-bit version of Fawlty must be used. In this case, the configuration value will be as follows:

```
export FL_DIR=$MIPS_HOME/fl/fl_0.79.12_32bit
```

Otherwise if the host system is a 64-bit system, the configuration value will be as follows:

```
export FL_DIR=$MIPS_HOME/fl/fl_0.79.12_64bit
```

Note: For an Intel Itanium based system, the 32-bit version of Fawlty **must** be used. There is not a specific Fawlty version available the Itanium architecture.

Note: On Ubuntu 10.04 32-bit the library '/usr/lib/libsm6' is not necessarily installed. The 32-bit Fawlty IDL interpreter needs this libarary. The library can be installed by executing the following command:

```
[user@host]$ sudo apt-get install libsm6
```

6.4 Setting subset configuration file

The subset configuration file defines all the subset images that will be created by MIPS. There are two subset configuration files per sensor. One of the subset configuration files must be a readonly file, no user or only one user should be able to edit this configuration file. All the subsets that must be created on a daily basis irrespective of the time of year must be in the readonly subset configuration file. The second configuration file list subsets that can be modified whenever a user wishes to do so.

The readonly subset configuration files for MERSI and MODIS are:

```
$MIPS_HOME/config/mersi_subsets_readonly.conf
$MIPS_HOME/config/modis_subsets_readonly.conf
```

The second type of configuration files for MERSI and MODIS are:

```
$MIPS_HOME/config/mersi_subsets.conf
$MIPS_HOME/config/modis_subsets.conf
```

Each line, in a subset configuration file, is a subset configuration definition. The following is an example of such a definition:

```
Beijing;114.26 41.97 118.13 38.75
```

There are two fields and these two fields are separated by a semicolon (';') character. The first field is the name of the subset. The subset output images will begin with this name. The second field is the upper-left and lower-right corner coordinates of the subset geographical region. The first two coordinates are the upper-left longitude and latitude values. The last two coordinates are the lower-right longitude and latitude values.

6.5 Setting the sensor band combinations for false color images

Three of a sensor's solar reflection bands, for both MERSI or MODIS, can be used in a combination to create false color images. Each of the three bands represent the colors red, green and blue in an image. The red, green and blue color combination for a MERSI true color image is band number 3, 2 and 1. For MODIS the band combination for true color images are band numbers 1, 4 and 3.

If all the bands in the band combination are not of the same spatial resolution, a technique known as "band sharpening" can be used. With this technique the highest resolution band, of the three bands in the band combination, can be used to increase the spatial resolution of the lower resolution bands. In the case for a MERSI true color image all the bands have a maximum resolution of 250m, thus band sharpening is not necessary. But only one band for MODIS true color images has a resolution of 250m, while the rest have a maximum resolution of 500m. In this case band sharpening has to be done with band 1.

Different band combinations are used for polar and non-polar images. Therefore there are two different configuration files that are used to define the band combination for the false color images. For MERSI the two configuration files are:

```
$MIPS_CONF/mersi_tile_bands.conf
$MIPS_CONF/mersi_polar_tile_bands.conf
```

For MODIS the two configuration files are:

```
$MIPS_CONF/modis_tile_bands.conf
$MIPS_CONF/modis_polar_tile_bands.conf
```

Each configuration line has either three or four numbers. The first three numbers are the band numbers that must be used in the red, green and blue color combinations. For example, the following configuration for MERSI defines the band combinations for true color images:

```
3 2 1
```

The fourth number is the band number that must be used for band sharpening. For example the following configuration indicates that band number one must be used to sharpen band numbers six and seven:

```
1 6 7 1
```

The following table shows the solar reflection bands, for MERSI and MODIS, that are available in MIPS for false color configurations:

MERSI		**MODIS**	
Band number	**Center Wavelength (µm)**	**Band number**	**Center Wavelength (µm)**
1	0.470	3	0.469
2	0.550	4	0.555
3	0.650	1	0.645
4	0.865	2	0.858
N/A	N/A	5	1.240
6	1.640	6	1.640
7	2.130	7	2.130

The following table show the false color band combination that is enabled by default in MIPS.

Sensor	**Band combination**	**Purpose**
MERSI	3 2 1	True color image
MERSI	1 6 7 1	Highlights snow and ice

Sensor	Band combination	Purpose
MERSI	7 4 3 3	Highlights vegetation
MODIS	1 4 3 1	True color image
MODIS	3 6 7 3	Highlights snow and ice
MODIS	7 2 1 1	Highlights vegetation

6.6 Setting the polar mosaic image outputs

The default pixel resolution at which the polar mosaic images are created can be set by the following configuration value:

```
export MIPS_POLAR_MOSIAC_RESOL=250
```

The `MIPS_POLAR_MOSIAC_RESOL` value must be a natural number that is devisable by 250. The latitude from which the Northern or Southern polar mosaic start from can be configured by the following two variables:

```
export MIPS_NORTH_POLARM_LAT=80
export MIPS_SOUTH_POLARM_LAT=-80
```

For the variable `MIPS_NORTH_POLARM_LAT` the absolute minimum value is 41.0 degrees, and for `MIPS_SOUTH_POLARM_LAT` the absolute maximum value is -41.0 degrees.

MIPS can also create a 1000m and a 4000m JPEG image of each polar mosaic image. To enable this capability, the following two variables can be set to either `"YES"` or `"NO"`.

```
export MIPS_POLAR_MOSIAC_1KM="YES"
export MIPS_POLAR_MOSIAC_4KM="YES"
```

6.7 Setting the global mosaic output

The pixel resolution at which the global mosaic images are created, can be set by the following variable:

```
export MIPS_GLOBAL_MOSAIC_RESOL=1000
```

The value must be a natural number and must be devisable by 250.

6.8 Setting GeoTIFF image tile outputs

Both polar and non-polar tile and composite images are created as PNG images. If the following variable equals `"YES"`, MIPS will also create GeoTIFF image together with the PNG images:

```
export MIPS_TILE_GEOTIFF_BL="YES"
```

6.9 Setting image enhancement coefficients

MIPS uses a non-linear brightness enhancement technique to enhance the output images. The enhancement coefficients are set in a file for each satellite. For MERSI, the following variables points to the files which have the enhancement coefficients:

```
export MIPS_FY3A_LENCH_FILE=$MIPS_CONF/mersi_fy3a_lench.conf
export MIPS_FY3B_LENCH_FILE=$MIPS_CONF/mersi_fy3b_lench.conf
```

For MODIS the following variables points to the files which have the enhancement coefficients:

```
export MIPS_TERRA_LENCH_FILE=$MIPS_CONF/modis_terra_lench.conf
export MIPS_AQUA_LENCH_FILE=$MIPS_CONF/modis_aqua_lench.conf
```

If the enhancement coefficient file is empty, MIPS will use built in coefficients to do the non-linear brightness enhancement. The following is an example of what is in the file `mersi_fy3a_lench.conf`:

```
03_02_01:0 30 60 120 190 255:0 110 160 210 240 255
07_04_03_03:0 30 60 120 190 255:0 110 160 210 240 255
01_06_07_01:0 30 60 120 190 255:0 110 160 210 240 255
```

In each line there are three fields. The first field is the band combination to which the enhancement coefficients apply to. In the second field are the input brightness coefficient values, and in the last field are the output brightness coefficient values. The number of elements in the input and output coefficient arrays must be equal to each other. The coefficients are applied to all the RGB channels.

For the `03_02_01` band combination, all the pixel values between 0 and 30 are scaled to 0 and 110. Therefore, the pixel values between 0 and 30 are scaled to be brighter. Then all the pixel values between 30 and 60 are scaled to 110 and 160. This continuous until all the pixel values between 190 and 255 are scaled to 240 and 255.

6.10 Setting image sharpening parameters

MIPS use ImageMagick's `convert` command to do image sharpening on the polar and non-polar tile and composite images. The parameters that are given to the `convert` command can be modified in the MIPS configuration file by modifying the following variable:

```
export MIPS_IMG_SHARPEN_STR="-sigmoidal-contrast 3,50% -modulate
100,125,100 -gamma 1.1 -unsharp 0"
```

Refer to the ImageMagick documentation for more information about the parameters that are used.

7. Adding logo for KMLs

If during runtime a file named logo.jpg is found in the MIPS KML directory, it will be used as a logo image in the KML files generated by MIPS. The logo will be displayed at the top left of the Google Earth window. The logo image can be any size, however approximately 100x100 pixels is recommended. You can use the ImageMagick convert command to create a suitable image. For example:

```
[user@host]$ convert -geometry 100x100 SSEC_logo.jpg logo.jpg
[user@host]$ cp logo.jpg $MIPS_HOME/kml
```

8. Required input data

The required data for MIPS to process MERSI data, are the following HDF5 L1B files:

- Level-1B 1000m resolution
- Level-1B 250m resolution

Both these L1B files are required when processing a MERSI granule or overpass.

The required input data for MIPS when processing MODIS data, are the following HDF4 L1B files:

- Level-1B 1000m resolution (MOD021KM / MYD021KM)
- Level-1B 500m resolution (MOD02HKM / MYD02HKM)
- Level-1B 250m resolution (MOD02QKM / MYD02QKM)
- Level-1B geolocation (MOD03 / MYD03)

All for these L1B files are required when processing a MODIS granule or overpass.

9. Running the MIPS software

The MIPS package includes a couple of test cases which can be used to test MIPS. For this section, the L1B granule in the file ‘mips-1.1_FY3A_nonpolar_test_data.tar.gz’ will be used to explain how to execute the MIPS processing scripts. This test case only contains one MERSI L1B granule, thus this section will focus on explaining how to process a MERSI L1B granule. The different steps that need to be followed to create a subset image, KML files, a polar mosaic image and a global mosaic image will be described in following subsections. A short overview of what each each script does will also be given.

Before continuing with the rest of this section, first set the environmental variable MIPS_HOME equal to the filesystem path of the MIPS software. For example if MIPS is installed under the user home directory, the command will be:

```
[user@host]$ export MIPS_HOME=$HOME/mips.version.1.1
```

Next the configuration must be loaded which sets the processing environment. In this section MERSI data will be processed, thus the MERSI configuration file must be loaded:

```
[user@host]$ source $MIPS_HOME/scripts/mips_mersi_env.bash
```

9.1 Untar test file and locating the L1B granule files

First create a temporary directory, go into the new directory and copy the file 'mips-1.1_FY3A_nonpolar_test_data.tar.gz' into the temporary directory. Next untar the file. It is assumed that the user as a local temporary directory.

```
[user@host]$ mkdir -p $HOME/tmp/mips_test_case
[user@host]$ cd $HOME/tmp/mips_test_case
[user@host]$ tar xzf $HOME/mips_version1.1_package/
mips-1.1_FY3A_nonpolar_test_data.tar.gz
```

The MERSI L1B granule files are located under the directory
mips-1.1_FY3A_nonpolar_test_data/input:

```
[user@host]$ ls mips-1.1_FY3A_nonpolar_test_data/input
FY3A_MERSI_GBAL_L1_20101005_0305_0250M_MS.HDF
FY3A_MERSI_GBAL_L1_20101005_0305_1000M_MS.HDF
```

Copy the L1B files to a directory where in the data will be processed. The temporary directory can be removed if it is not going to be used as the processing directory.

```
[user@host]$ cd $HOME
[user@host]$ mkdir -p $HOME/tmp/mips_process_data
[user@host]$ cp $HOME/tmp/mips_test_case/
mips-1.1_FY3A_nonpolar_test_data/input/*  $HOME/tmp/mips_process_data
[user@host]$ rm -r $HOME/tmp/mips_test_case
[user@host]$ cd $HOME/tmp/mips_process_data
```

9.2 Processing L1B granule

There are two script which can be used to process a MERSI L1B granule. The names of the two script are:

```
$MIPS_HOME/scripts/process_granule_tiles_mersi_wrap.bash
$MIPS_HOME/scripts/process_granule_tiles_mersi.bash
```

If the MERSI L1B file names have the same naming convention as those given in section 9.1, the first script can be used. The first script will extract the satellite name, date, time and pixel resolution from the file names. Here is an example of how the first script is executed.

```
[user@host]$ $MIPS_HOME/scripts/process_granule_tiles_mersi_wrap.bash
FY3A_MERSI_GBAL_L1_20101005_0305_1000M_MS.HDF
FY3A_MERSI_GBAL_L1_20101005_0305_0250M_MS.HDF
```

If the L1B files do not have the same naming convention as the file names given in section 9.1, the second script must be used. The script requires that together with the L1B file names the satellite name (in small letters), date and time to be given to the script. The following is an example of how the second script is executed.

```
[user@host]$ $MIPS_HOME/scripts/process_granule_tiles_mersi.bash
FY3A_MERSI_GBAL_L1_20101005_0305_1000M_MS.HDF
FY3A_MERSI_GBAL_L1_20101005_0305_0250M_MS.HDF fy3a 20101005 0305
```

The scripts will first create corrected reflectance data from the given L1B files. This is done to improve the visual quality of the data. Thereafter the scripts determine on what 10 by 10 degree tiles the data can be projected on, using "equirectangular projection". This only applies to data that are 80 degrees south of the North pole and -80 degrees north of the South pole. These tile images are used by the subset, KML and global mosaic image processing scripts to create the output product images. Next the corrected reflectance data are projected onto the different tiles. If a previous L1B granule was processed with of the same date and the same tiles were created, the scripts will overlay the new tiles on the old tiles. This process is called image compositing. The whole process described in this paragraph is repeated for every false color band combination that is defined in tile bands configuration file.

If the L1B granule covers either the North or South polar region, the scripts determine on what polar tiles the data can be projected on. The size of a polar tile is 1024km by 1024km, and there are a total of 64 tiles per polar region. The projection that is used is called "polar stereographic projection". Next the corrected reflectance data are projected onto the different polar tiles. As with the previous case; If there are older polar tiles of the same day, the scripts will overlay the new polar tiles on the old polar tiles. The whole process described in this paragraph is repeated for every color band combination that is defined in the polar tile bands configuration file.

Together with the image tiles, true and false color images, the two processing scripts also create what is called event files. The purpose of the event files are to act as indicators for the subset, KML, polar mosaic and global mosaic processing scripts. These scripts search for specific event files, and based on the event files further processing is done.

9.3 Subset processing

To initiate the process for the subset image processing, the following script must be executed:

```
[user@host]$ $MIPS_HOME/scripts/process_subsets.bash
```

The script does not take any input argument. It uses the environmental variables defined by the configuration file 'mips_mersi_env.bash' and available event files to determine what subsets must be created.

The process_subsets.bash script loads both the readonly and the plain subset configuration files. For each subset configuration definition it lists what tiles overlap the defined subset region. For each tile in the list, per configuration definition, the script determines if there is an available event file. An event file indicates that there is a new composite that can be used to make the subset. If there is an event file which has the same tile number than what is in the list, the script creates a subset and use the available composite image. The script repeats the process for each tile number in the list that overlaps the subset region.

If more than 60% of a subset image is complete, the subset image is copied to the final subset output directory. Otherwise the file is temporarily archived in an intermediate directory until it can be copied to the final output directory.

9.4 KML processing

The KML processing script, like the subset processing script, does not take any input arguments. To initiate the KML data processing, the following script must be executed:

```
[user@host]$ $MIPS_HOME/scripts/process_kml.bash
```

For each available composite image, indicated by an event file, the KML processing script decimates the image into smaller tiles. This is called pyramid tiling: high resolution images are broken up in smaller images. These smaller images are indexed by individual KML files, and these KML files are again indexed by higher order KML files. There is for each tile a KML file which points to the smaller tile KML files that represent parts of the composite tile image.
Next the process_kml.bash script creates a KML script which acts as an index for each tile's index KML file. Google Earth use this KML file to load the underlaying KML files and the composite images.

9.5 Polar mosaic processing

The polar mosaic processing script does not take any input arguments. To initiate the polar mosaic processing, the following script must be executed:

```
[user@host]$ $MIPS_HOME/scripts/process_polar_mosaic.bash
```

The polar mosaic script first finds all the available event files in the polar tile archive directory. If there are any event files, for each event file the script determines for what day, satellite, band combination and region (North or South) a polar mosaic image can be created. Then for each day, satellite, band combination and polar region that was identified, the script creates a polar mosaic image. All the missing tiles for a specific polar mosaic image, are replaced by empty polar image tiles. Therefore whenever there is a missing tile, there will be a black tile in the polar mosaic output image.

9.6 Global mosaic processing

The global mosaic processing script does not take any input arguments. To initiate the global mosaic processing, the following script must be executed:

```
[user@host]$ $MIPS_HOME/scripts/process_global_mosaic.bash
```

The global mosaic script first finds all the available event files in the tile archive directory. If there are any event files, for each event file the script determines for what day, satellite, band combination a global mosaic image can be created. Then for each day, satellite and band combination that was identified, the script creates a global mosaic image. All the missing tiles for a specific global mosaic image are replaced by empty tile images. Therefore whenever there is a missing tile, there will be a black tile in the global mosaic output image.

10. Output directories

There are six directories where in MIPS product images are archived. Each of these directories corresponding to the image products, which was discussed in section 9, will be described in the following section.

10.1 Granule images

When the granule processing scripts (section 9.2) create corrected reflectance data from a L1B granule, a true color unprojected image is created from the corrected reflectance data. The granule images are archived in the directory $MIPS_HOME/granimg. Here is an example of a directory listing of this directory :

```
[user@host]$ ls $MIPS_HOME/granimg
fy3a.1km.20101005.0305.03_02_01.hdf.jpg
```

The naming convention that is used for the granule images is:
<satellite name>●<pixel resolution>●<year><month><day>●<band combination>.hdf.jpg

Given that the granule images always have a spatial resolution of 1km, the sharpening band number is never given in the band combination field in the file name.

10.2 Image tiles

The granule processing scripts (section 9.2) archives the non-polar tile images in the tiles directory. If a directory listing is done in the tiles directory, the following directories will be present:

```
[user@host]$ ls $MIPS_HOME/tiles
tile425
tile426
tile459
tile460
tile461
tile462
tile463
tile495
tile496
tile497
tile498
tile499
```

For each 10 by 10 degree tile there is a separate directory. In each tile directory the image composites are archived together with other data files that are used by MIPS. There will be a total of 648 tile directories if all the data from 90 degree North to -90 degrees South are processed. A map of the tiles overlaid on a projected image of the continents is given in appendix A.3.

10.3 Polar image tiles

The granule processing scripts (section 9.2) archived polar data in the polartiles directory. If a directory listing is done in the polartiles directory, the following directories will be present:

```
[user@host]$ ls $MIPS_HOME/polartiles
north
south
```

There are two sub-directories in the polartiles directory, one for the Northern polar region and one for the Southern polar region. If a directory listing is done for the Northern polar region polartiles directory, the following directory will be present:

```
[user@host]$ ls $MIPS_HOME/polartiles/north
ptile063
```

In each polar tile directory polar image composites are archived together with other data files that are used by MIPS. There will be a total of 64 polar tile directories in each of the Northern and Southern directories. The size of each tile is 1024km by 1024km.

10.4 Subset images

All subset images are archived under the directory $MIPS_HOME/subsets. Each subset file starts with the name that was specified in the subset configuration files. As it was described in section 9.3, the subset images that are not yet complete are archived under the directory $MIPS_HOME/subsets/intermediate. Here is an example of a directory listing of the subsets directory:

```
[user@host]$ ls $MIPS_HOME/subsets
Beijing.fy3a.20101005.03_02_01_00.dat
Beijing.fy3a.20101005.03_02_01_00.jpg
Beijing.fy3a.20101005.03_02_01_00.tif
intermediate
```

For each subset image, there is a GeoTIFF image together with a JPEG image. The naming convention that is used for the subset images is:
<subset name>●<satellite name>●<year><month><day>●<band combination>.<jpg, tif>

The band combination is a series of four two-digits. The first two-digit is the band number of the sensor that is used as the color red. The second and third two-digits are the band numbers for the colors green and blue. The fourth two-digit is the band number that was used for band sharpening for this image. If no band sharpening was done, this two-digit will be equal to “00”.

10.5 KML files

There are four different KML file types in the directory $MIPS_HOME/kml. The four types are called the Today, current, main index and global index KML files.

The first type of KML file has the keyword ‘Today’ in the file name. This is the KML file that a user must download if he or she wants to view the most current tile images. If MIPS processed L1B granules that are older than a day, this KML file type will not show any images. Only the most recent file images are displayed with this KML file. The Today-KML file also automatically refresh every 10 minutes the images that are displayed on Google Earth, therefore the user never has to re-download this KML file.

The second type of KML file that will be present has the keyword ‘current’ in the file name. The Today-KML file links to this KML file. Thus if MIPS processed L1B granules that are older than a day, this file will not exist for those particular granules. The current-KML file links to the main index KML file. The URL that is defined by the configuration variable MIPS_KML_URL (see section 6.2) is used in the current-KML file to link to the global index KML file.

For the third type of KML files there is a main index KML for every satellite name, day and band combination data set. This type of KML file will always be created. The main index KML links to a global index KML file which in turn links to the tile index KML files. The main index KML file links to the global index KML file using the URL given by the configuration value MIPS_KML_URL. If a user want to load a specific date’s KML file, this main index KML file must be loaded.

Here is an example of a directory list of the KML product directory:

```
[user@host]$ ls $MIPS_HOME/kml
20101005
BEIJING.20101005.fy3a.03_02_01_00.kml
BEIJING_MERSI_03_02_01_00_Today.kml
fy3a.20101005.update.jpg
intermediate
```

Here is an example the directory listing of a specific day and band combination’s KML directory:

```
[user@host]$ ls $MIPS_HOME/kml/20101005/03_02_01_00
20101005.03_02_01_00.fy3a.kml
tile425_fy3a
tile426_fy3a
tile460_fy3a
tile461_fy3a
tile462_fy3a
tile496_fy3a
tile497_fy3a
tile498_fy3a
tile499_fy3a
```

The first file in the above directory listing is the global index KML file. This KML file links to all the KML files in each tile directory. The global index KML does not use the URL given by the configuration variable MIPS_KML_URL, it uses a relative links to link to the tile KML files. Thus if directory '$MIPS_HOME/kml/20101005/03_02_01_00' is copied to a user's computer, the global index KML file can be loaded in Google Earth. If the global index KML file is loaded from a web server, Google Earth will not display any images.

The naming convention for the Today KML file type is:
<site name>_<sensor name>_<band combination>_Today.kml

The naming convention for the current-KML file type is:
<site name>●<current>●<satellite name>●<band combination>.kml

The naming convention for the main index KML file is:
<site name>●<year><month><day>●<satellite name>●<band combination>.kml

The naming convention for the global index KML file is:
<year><month><day>●<band combination>●<satellite name>.kml

10.6 Polar and global mosaic

The polar mosaic images and global mosaic images share the same archive directory. The polar mosaic images are clearly labeled with the keywords 'Arctic' and 'Antarctic'. The global mosaic images starts with the keyword 'Global'. Here is a directory listing of the mosaic directory:

```
[user@host]$ ls $MIPS_HOME/mosaic
Arctic.fy3a.20101005.01_06_07_01.1km.jpg
Arctic.fy3a.20101005.01_06_07_01.1km.tif
Arctic.fy3a.20101005.01_06_07_01.4km.jpg
Arctic.fy3a.20101005.03_02_01_00.1km.jpg
Arctic.fy3a.20101005.03_02_01_00.1km.tif
Arctic.fy3a.20101005.03_02_01_00.4km.jpg
Global.fy3a.20101005.03_02_01_00.4km.jpg
Global.fy3a.20101005.03_02_01_00.4km.tif
```

For the polar mosaic images there are three file types. The first is a GeoTIFF file and the other are two JPEG files. One of the JPEG files and the GeoTIFF file has a pixel resolution of 1000m. The second JPEG file has a pixel resolution of 4000m.
The global mosaic images are available in a GeoTIFF and a JPEG file format. With both these files the pixel resolution is 4000m.

The naming convention of the mosaic images is:
<Arctic, Antarctic, Global>●<satellite name>●<year><month><day>●<band combination>●<pixel resolution>●<jpg, tif>

11. Testing MIPS

There are eight tar files in the MIPS package that have L1B granule files of the sensors MERSI and MODIS. The eight tar file name are given in section 4. These files are test cases which can be used to make sure that MIPS was successfully installed. The content of the first test case is given below:

```
[user@host]$ cd $HOME/tmp
[user@host]$ tar xzf $HOME/mips_version1.1_package/
mips-1.1_FY3A_nonpolar_test_data.tar.gz
[user@host]$ find ./mips-1.1_FY3A_nonpolar_test_data -maxdepth 2  -
print | sort
./mips-1.1_FY3A_nonpolar_test_data
./mips-1.1_FY3A_nonpolar_test_data/compare_output
./mips-1.1_FY3A_nonpolar_test_data/compare_output/granimg
./mips-1.1_FY3A_nonpolar_test_data/compare_output/kml
./mips-1.1_FY3A_nonpolar_test_data/compare_output/mosaic
./mips-1.1_FY3A_nonpolar_test_data/compare_output/polartiles
./mips-1.1_FY3A_nonpolar_test_data/compare_output/
process_global_mosaic.log
./mips-1.1_FY3A_nonpolar_test_data/compare_output/
process_granule_tiles_mersi_wrap.log
./mips-1.1_FY3A_nonpolar_test_data/compare_output/process_kml.log
./mips-1.1_FY3A_nonpolar_test_data/compare_output/
process_polar_mosaic.log
./mips-1.1_FY3A_nonpolar_test_data/compare_output/process_subsets.log
./mips-1.1_FY3A_nonpolar_test_data/compare_output/scratch
./mips-1.1_FY3A_nonpolar_test_data/compare_output/subsets
./mips-1.1_FY3A_nonpolar_test_data/compare_output/tiles
./mips-1.1_FY3A_nonpolar_test_data/input
./mips-1.1_FY3A_nonpolar_test_data/input/
FY3A_MERSI_GBAL_L1_20101005_0305_0250M_MS.HDF
./mips-1.1_FY3A_nonpolar_test_data/input/
FY3A_MERSI_GBAL_L1_20101005_0305_1000M_MS.HDF
./mips-1.1_FY3A_nonpolar_test_data/output
```

There are three sub-directories under a test case directory which are called ‘input’, ‘output’ and ‘compare_output’. The L1B granule files that are used for the test cases are archived in the directory ‘input’. When a test case process is initiated the output of the test will be archived under the directory ‘output’. This includes all the MIPS output directories. After a test case process has finished processing the L1B granule, the data in the ‘output’ directory can be compared to the data in the ‘compare_output’ directory. The content of the two output directories must be the same.

The following steps, given in the list below, should be followed when a test case file is to be used:

1. Untar a test case file into a temporary directory.
2. Define the MIPS_HOME environmental variable.

3. Execute the script ‘execute_test_case_mersi.bash’ for MERSI data, or execute the script ‘execute_test_case_modis.bash’ for MODIS data.
4. Compare the images in the tiles and polar tiles directories.
5. Compare the subset images.
6. Compare the both the polar and global mosaic images.
7. Compare the global index KML files with Google Earth.

Examples of each step will now be given in the following subsections.

11.1 Setting up the test environment

First a temporary directory has to be created where in the the test case can untarred. For this example it is assumed to user has a temporary directory called ‘tmp’ under the home directory. Next the MIPS_HOME environmental variable has to be set. It is assumed that MIPS is installed under the user’s home directory. Here is an example of setting up the test environment:

```
[user@host]$ cd $HOME/tmp
[user@host]$ tar xzf $HOME/mips_version1.1_package/
mips-1.1_FY3A_nonpolar_test_data.tar.gz
[user@host]$ ls
mips-1.1_FY3A_nonpolar_test_data
[user@host]$ export MIPS_HOME=$HOME/mips.version.1.1
```

11.2 Initiating test case process

In this example a MERSI test case will be used. Thus the script that is going to be executed is called ‘execute_test_case_mersi.bash’. The script will load the MIPS configuration itself, therefore there is no need to do it prior to starting the test case process. Here is an example of how the test case process is initiated:

```
[user@host]$ $MIPS_HOME/scripts/utilities/execute_test_case_mersi.bash
$HOME/tmp/mips-1.1_FY3A_nonpolar_test_data
```

First the script will execute the processing script ‘process_granule_tiles_mersi_wrap.bash’. Thereafter the test case script execute the product scripts to create the KML files, subset and mosaic images.

The test case script creates all the output directories in the directory $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output. The script will also create five log files in the output directory. The log file names are:

- process_granule_tiles_mersi_wrap.log
- process_kml.log
- process_subsets.log
- process_polar_mosaic.log
- process_global_mosaic.log

To view the progress of the test case process, execute the following command:

```
[user@host]$ cd $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output
[user@host]$ tail -F process_granule_tiles_mersi_wrap.log
process_kml.log process_subsets.log process_polar_mosaic.log
process_global_mosaic.log
```

Any errors that occur during the processing of the L1B granule will be written one of the log file. An explanation of the error codes is given in appendix A.4.

11.3 Comparing the tile and polar tiles output directories

If no error was reported in one of the log files, the images have been successfully produced. First the individual tile image and the polar tile image in the directories ‘output’ and ‘compare_output’ must be compared to one another.

The two sets of images will be visually indistinguishable from each other, but numerically (**not** visually) the pixel color values might differ slightly. Given that test case could have originally been executed on a different type of system (32-bit or 64-bit) than the user’s system, the rounding of the floating point calculations might be different for each case. Given this reasoning, the images must be compared visually and not quantitatively.

To compare the two images sets, first create two directories where in the images can be viewed. In proceeding example the two directories will be created in the directory $HOME/Documents. The tiles and polar tile images will be copied to these two directories. The one directory called ‘output_tiles’ will have the output results of this test case. The second directory called ‘compare_output_tiles’ will have tile images to which the test case’s images must be compared to.

```
[user@host]$ mkdir -p $HOME/Documents/output_tiles $HOME/Documents/
compare_output_tiles
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output -
name 'tile*.jpg' -type f -exec cp {} $HOME/Documents/output_tiles \;
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/
compare_output -name 'tile*.jpg' -type f -exec cp {} $HOME/Documents/
compare_output_tiles \;
```

11.4 Comparing the subset images and mosaic images

The subset images and the mosaic images must also be compared visually. To compare the two mosaic and subset image sets, two directories must be created where to the image sets are to be copied. The one directory called ‘output_mosaic_subset’ will have the output results of the mosaic and subset images. The second directory ‘compare_output_mosaic_subset’ will have the mosaic images to which the test case’s mosaic and subset images must be compared to.

```
[user@host]$ mkdir -p $HOME/Documents/output_mosaic_subset $HOME/
Documents/compare_output_mosaic_subset
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output/
mosaic -name '*.jpg' -type f -exec cp {} $HOME/Documents/
output_mosaic_subset \;
```

```
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output/
subsets -name '*.jpg' -type f -exec cp {} $HOME/Documents/
output_mosaic_subset \;
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/
compare_output/mosaic -name '*.jpg' -type f -exec cp {} $HOME/
Documents/compare_output_mosaic_subset \;
[user@host]$ find $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/
compare_output/subsets -name '*.jpg' -type f -exec cp {} $HOME/
Documents/compare_output_mosaic_subset \;
```

11.5 Comparing the KML files

A HTTP address was not configured for the KML files in the test case. The KML data directories has to be copied to a directory where from the global index KML files can be loaded into Google Earth. The KML global index data directory that was generated in this test case will be copied to a directory called ‘output_kml’. The KML global index where to the test case’s KML files must be compared to, will be copied to a directory called ‘compare_output_kml’.

```
[user@host]$ mkdir -p $HOME/Documents/output_kml $HOME/Documents/
compare_output_kml
[user@host]$ cp -r $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/output/
kml/20101005/03_02_01_00/* $HOME/Documents/output_kml
[user@host]$ cp -r $HOME/tmp/mips-1.1_FY3A_nonpolar_test_data/
compare_output/kml/20101005/03_02_01_00/* $HOME/Documents/
compare_output_kml
```

To compare the two global index KML files, open the following two KML files in Google Earth:

```
$HOME/Documents/output_kml/20101005.03_02_01_00.fy3a.kml
$HOME/Documents/compare_output_kml/20101005.03_02_01_00.fy3a.kml
```

12. Credits

The Multi-Sensor Image Processing System (MIPS) package was developed by Liam Gumley, Amato Evan and Willem Marais at the Space Science and Engineering Center, University of Wisconsin-Madison. Support for MIPS development was provided by the National Satellite Meteorological Agency, of the China Meteorological Administration.

Portions of the Multi-Sensor Image Processing System package were obtained from the following freely available software packages:

Corrected Reflectance version 1.4.2 (crefl) was developed by Jacques Descloitres
and was obtained from the NASA GSFC Direct Readout Laboratory:
http://directreadout.sci.gsfc.nasa.gov/

HDF5 file dump version 1.6.10 (h5dump) and HDF5 to HDF4 converter version 2.1.1 (h5toh4) was developed by the NCSA HDF Team and was obtained from the HDF Group:
http://www.hdfgroup.org/

HDF file dump version 4.2r3 (ncdump) was developed by the NCSA HDF Team and was obtained from the HDF Group:
http://www.hdfgroup.org/

Fawlty Language version 0.79.12 (fl) was developed by Lajos Földy and was obtained from the FL website:
http://fl.net23.net/

The Geotiff copy version 1.3.0 (geotifcp) binary is part of the libgeotiff package. The original author of the libgeotiff package is Niles Ritter and it is currently maintained by Frank Warmerdam. It was obtained from the Geotiff website:
http://trac.osgeo.org/geotiff/

The Cartographic Coordinate System Filter version 4.7.1 (cc2cs) is part of the Proj.4 package. The original author of the software is Gerald Evenden and is currently maintained by Frank Warmerdam. It was obtained from the Proj.4 project website:
http://trac.osgeo.org/proj/

The latitude-longitude pair to column-row pair converter version 1.14 (ll2cr) and the forward navigation version 1.31 (fornav) binaries are part of the MODIS Swath-to-Grid Toolbox version 0.22 (MS2GT). MS2GT was developed at the National Snow & Ice Data Center, University of Colorado, by Terry Haran. It was obtained from the NSIDC website:
http://nsidc.org/data/modis/ms2gt/

The arbitrary precision calculator version 1.06 (bc) is part of the GNU operating system and was obtained from:
http://www.gnu.org/s/bc/

Appendix A

A.1 Compiling MIPS binaries

The configuring and compiling of the MIPS binaries are conveniently done by a script which is part of MIPS. The script was written and tested on a Ubuntu server version number 10.04.3 LTS. If the script is executed on a different system, modifications to the script might have to be made.

The software that are going to be configured and compiled depend on other external development files and programs. The following software must be installed prior to executing the script:

- gcc
- gfortran
- bison
- flex

This is done through executing the following command on Ubuntu:
sudo apt-get install build-essential gfortran bison flex

Next the script must be executed which compiles the software. The script is located in the directory $MIPS_HOME/scripts/utilities. The script takes three arguments. The first argument is the location of the packages that will be compiled. The second argument is the location where the compile binaries and libraries will be copied to. The third argument is the filesystem path to the directory where the MIPS binary directory is, which is $MIPS_HOME/bin. An example will now be given to show how the compile script is executed. It is assumed the directory of the MIPS packages and MIPS are located under the user's home directory.

```
[user@host]$ MIPS_PACKAGE_DIR=$HOME/mips_version1.1_package/
mips-1.1_compile_binaries
[user@host]$ MIPS_COMPILE_DIR=$HOME/tmp/mips_compile
[user@host]$ MIPS_HOME=$HOME/mips.version.1.1
[user@host]$ mkdir -p $MIPS_COMPILE_DIR
[user@host]$ $MIPS_HOME/scripts/utilities/compile_binaries_32bit.bash
$MIPS_PACKAGE_DIR $MIPS_COMPILE_DIR $MIPS_HOME/bin
```

On a 64-bit system, the command is as follows:

```
[user@host]$ $MIPS_HOME/scripts/utilities/compile_binaries_64bit.bash
$MIPS_PACKAGE_DIR $MIPS_COMPILE_DIR $MIPS_HOME/bin
```

On a IA-64, the command is as follows:

```
[user@host]$ $MIPS_HOME/scripts/utilities/compile_binaries_ia64.bash
$MIPS_PACKAGE_DIR $MIPS_COMPILE_DIR $MIPS_HOME/bin
```

If ImageMagick is not going to be compiled on the host server - it is already installed on the MIPS server, the compile directory (MIPS_COMPILE_DIR) can be removed.

A.2 Configuring and compiling ImageMagick

It is assumed that the host server does not have any TIFF, PNG or JPEG libraries installed. These libraries first have to be compiled before ImageMagick can be compiled. If the MIPS binaries have been compiled (appendix A.1), then these image libraries had already been compiled - unless the compile directory (MIPS_COMPILE_DIR) had been removed.

First ImageMagick has to be configured before it can be compiled. The location of the image libraries has to be set and the installation directory also has to be set. In this case, ImageMagick is going to be installed in the user home directory - the same user which will execute MIPS. It is assumed that the ImageMagick package file is available under the user's home directory. Create a temporary directory and go to that directory prior to executing the following instructions:

```
[user@host]$ IMG_MGCK_INSTLL_DIR=$HOME/Apps
[user@host]$ mkdir -p $HOME/tmp
[user@host]$ cd $HOME/tmp
[user@host]$ tar xjf $HOME/mips_version1.1_package/
ImageMagick-6.7.1-2.tar.bz2
[user@host]$ cd ImageMagick-6.7.1-2
[user@host]$ CC="gcc" CXX="g++" CPPFLAGS="-I$MIPS_COMPILE_DIR/local-
install/include" LDFLAGS="-L$MIPS_COMPILE_DIR/local-install/lib"  ./
configure --prefix=$IMG_MGCK_INSTLL_DIR --without-jbig --without-jp2
--without-x --without-magick-plus-plus
```

If it is preferred to install ImageMagick not in the user directory but in the /usr directory, set the variable IMG_MGCK_INSTLL_DIR equal to "/usr".

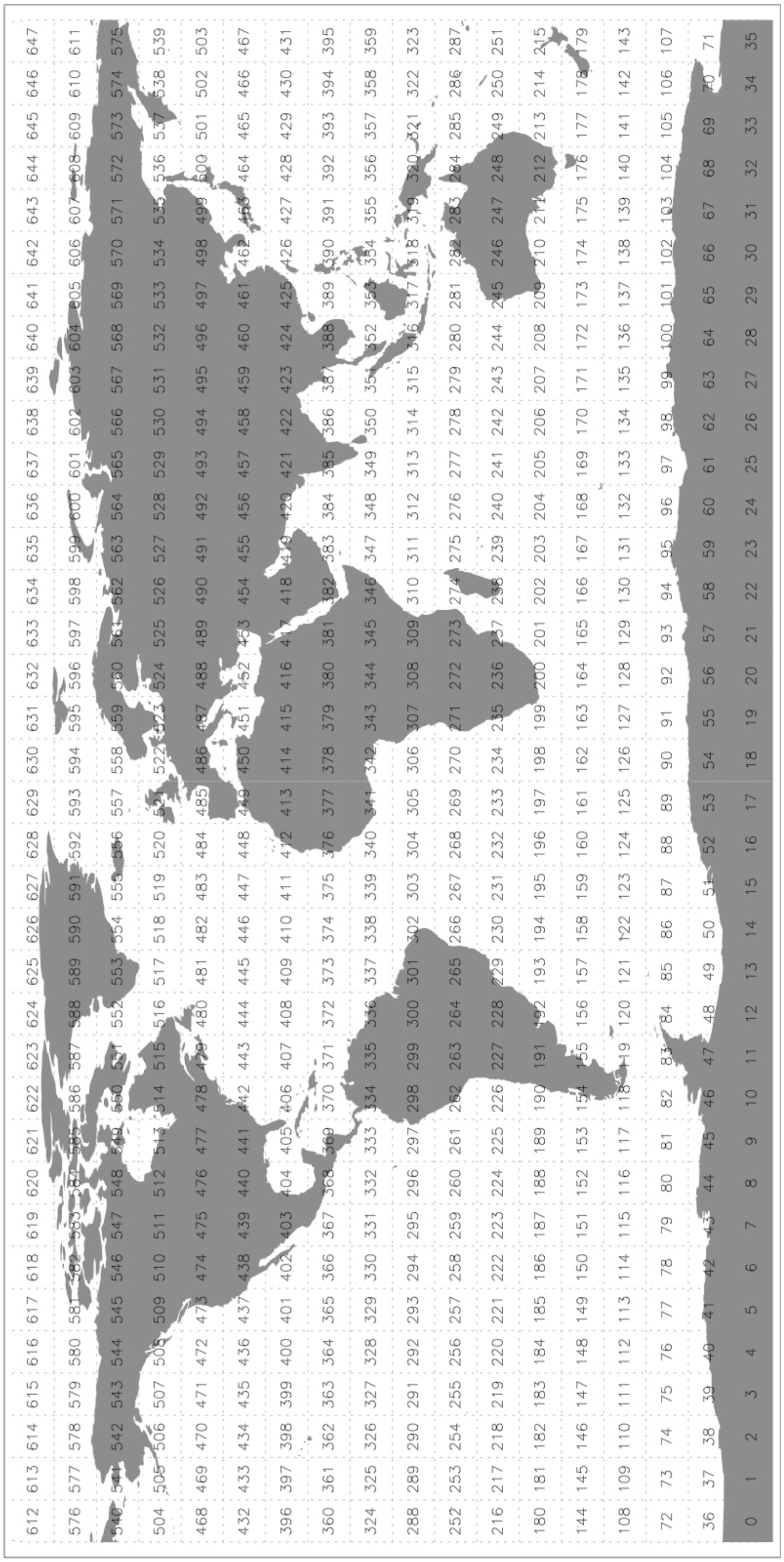
Next ImageMagick can be compiled and then installed:

```
[user@host]$ make -j3
[user@host]$ make install
```

If ImageMagick is installed under the directory $HOME/Apps, then MIPS needs to be configured to search for binary executables in a non-standard directory. Edit the MIPS configuration file - both $MIPS_HOME/scripts/mips_modis_env.bash and $MIPS_HOME/scripts/mips_mersi_env.bash or both, and set the PATH variable to the following:

```
export PATH=$HOME/Apps/bin:$MIPS_HOME/scripts:$MIPS_HOME/bin:/usr/
local/bin:$PATH
```

A.3 Globe tile map



A.4 Error numbers and comments

Error Nr	Script name	Comment
0	process_granule_tiles_modis.bash	The MODIS L1B files are not readable. Either the given files do not exist or the script does not have have permission to read the files.
1	run_mersi_crefl.bash	The CREFL binary failed to process the 1000m MERSI L1B. The L1B file’s content might be corrupt.
2	asc_desc_filt_binfile.bash	The IDL script asc_desc_filt_binfile.pro failed.
3	run_mersi_crefl.bash	The CREFL binary failed to process the 250m MERSI L1B. The L1B file’s content might be corrupt.
4	condition_szenangle_bin.bash	The IDL script condition_szenangle_bin.pro could not filter the sensor zenith angle data from the binary file.
5	run_modis_crefl.bash	The IDL script modis_saturation_fix.pro failed to process the MODIS HKM and QKM files. The L1B files might be corrupt.
6	run_modis_crefl.bash	The CREFL binary failed to process the 1KM MODIS L1B. The L1B file’s content might be corrupt.
7	run_modis_crefl.bash	The CREFL binary failed to process the HKM MODIS L1B. The L1B file’s content might be corrupt.
8	run_modis_crefl.bash	The CREFL binary failed to process the QKM MODIS L1B. The L1B file’s content might be corrupt.
9	process_granule_tiles_modis.bash	Either the MODIS L1B files are corrupt, or the order in which the L1B files were passed to the script is wrong.
10	process_granule_tiles_modis.bash	Either the satellite name, the date or UTC time that was passed to the script is invalid.
11	process_granule_img.bash	The IDL script make_granule_image_crefl.pro failed to process read the CREFL files to create an image. Either the CREFL files are corrupt, or the band numbers do not exist in the CREFL files.
12	process_granule_img.bash	The ImageMagick ‘convert’ executable failed to convert a PNG file into a JPEG file. Check to see if ImageMagick is properly installed.

Error Nr	Script name	Comment
13	run_metadata.bash	The IDL script write_global_metadata.pro failed to process the data in either the MERSI 1000m L1B or MODIS geolocation file.
14	run_metadata.bash	The script could not change the field value GRAN_NAME in a metadata file, the utility ‘sed’ failed.
15	process_granule_tiles.bash	The metadata file corresponding to the event file that was given to the script, does not exist. Either there is not enough disk space or another process has deleted the metadata file.
16	create_normal_tile_tiff.bash	The IDL script create_normal_tile_tiff.pro failed.
17	calc_grid_size.bash	The IDL script calc_grid_size.pro failed.
18	hdf_to_binary.bash	The IDL script hdf_to_binary.pro failed to extract a SDS array from a HDF4 file.
19	reproject_ms2gt_mix.bash	On of or all the binary files that the script created, where the data were copied from the given HDF4 files, do not exist.
20	reproject_ms2gt_mix.bash	The IDL script filter_high_sensor_zangle_bin failed.
21	reproject_ms2gt_mix.bash	The MS2GT ll2cr executable binary failed to convert longitude and latitude data into row and column values.
22	create_polartile_tiff.bash	The IDL scripts create_polartile_tiff.pro failed.
23	reproject_ms2gt_mix.bash	The MS2GT fornav executable binary failed to project the red, green and blue CREFL data.
24	reproject_ms2gt_mix.bash	The MS2GT fornav executable binary failed to project the CREFL sharpening band data.
25	reproject_ms2gt_mix.bash	The IDL script reproject_binary.pro failed to create an image from the projected data.
26	reproject_ms2gt_qkm.bash	On of or all the binary files that the script created, where the data were copied from the given HDF4 files, do not exist.
27	reproject_ms2gt_qkm.bash	The IDL script filter_high_sensor_zangle_bin failed.
28	reproject_ms2gt_qkm.bash	The MS2GT ll2cr executable binary failed to convert longitude and latitude data into row and column values.

Error Nr	Script name	Comment
29	reproject_ms2gt_qkm.bash	The MS2GT fornav executable binary failed to project the red, green and blue CREFL data.
30	reproject_ms2gt_qkm.bash	The IDL script reproject_binary_qkm.pro failed to create an image from the projected data.
31	run_reproject_ms2gt.bash	The sharpening band number is not defined. One of the following six scripts do not have a sharping band number in one of the configuration lines: • MODIS configuration file: mips_modis_env.bash • MERSI configuration file: mips_mersi_env.bash • MODIS polar tile band configuration file: modis_polar_tile_bands.conf • MERSI polar tile band configuration file: mersi_polar_tile_bands.conf • MODIS tile band configuration file: modis_tile_bands.conf • MERSI tile band configuration file: mersi_tile_bands.conf
32	run_reproject_ms2gt.bash	An output tile image was not successfully copied to the tiles archive directory.
33	run_reproject_ms2gt.bash	One of the MS2GT projection template files were not successfully duplicated. The if the files ‘normal_ms2gt_templ.gpd and ‘normal_ms2gt_templ.mpp’ exist in the directory $MIPS_HOME/templates. If these two files do not exist, re-install MIPS. Also check if there is enough disk space.
34	get_geo_bounds.bash	The script failed to calculate the corner coordinates of a non-polar mosaic image.
35	get_polar_bounds.bash	The script failed to calculate the corner coordinates of a polar mosaic image.
36	make_tile_composite_zangle_nr.bash	The IDL script image_szangle_composite.pro failed.
37	make_tile_composite_zangle_nr.bash	The script failed to change the new composite metadata file’s field GRAN_NAME.
38	make_tile_composite_zangle_nr.bash	The ImageMagick ‘convert’ executable failed to convert a PNG file into a JPEG file. Check to see if ImageMagick is properly installed.
39	process_granule_tiles_mersi.bash	The MERSI L1B files are not readable. Either the given files do not exist or the script does not have have permission to read the files.
40	process_granule_tiles_mersi.bash	Either the MODIS L1B files are corrupt, or the order in which the L1B files were passed to the script is wrong.
41	process_granule_tiles_mersi.bash	Either the satellite name, the date or UTC time that was passed to the script is invalid.

Error Nr	Script name	Comment
42	convert_mersi_h5toh4.bash	The executable update_fy3_l1b_calb could not copy the global MERSI L1B HDF5 attribute ‘VIR_Cal_Coeff’.
43	convert_mersi_h5toh4.bash	The executable h5toh4 failed to convert a MERSI 1KM L1B HDF5 file into a HDF4 file.
44	convert_mersi_h5toh4.bash	The executable h5toh4 failed to convert a MERSI 250m L1B HDF5 file into a HDF4 file.
45	process_subsets.bash	The subset configuration file is missing. Check in either the mips_modis_env.bash or mips_mersi_env.bash files if the variable MIPS_SUBSET_CONF points to a file that exist.
46	get_tileid_range.bash	The IDL script get_tileid_range.pro failed.
47	make_cut_subset.bash	A specific composite tile’s (non-polar) projection text file is corrupt.
48	make_cut_subset.bash	A specific composite tile’s (non-polar) projection text file is corrupt.
49	make_cut_subset.bash	The IDL script make_cut_subset.pro failed.
50	process_kml_bandcomb.bash	The IDL script make_image_tiles.pro failed.
51	process_kml_bandcomb.bash	The IDL script make_global_kml.pro failed.
52	process_kml_bandcomb.bash	The IDL script make_user_kml.pro failed.
53	run_reproject_ms2gt.bash	The IDL script l1b_geo_ul_lr.pro failed to extract the upper left and lower right coordinates from a given L1B file. Either a MODIS geolocation file is corrupt or a MERSI 1000m file is corrupt.
54	process_kml_bandcomb.bash	A corresponding metafile of an event file is missing. It could be that there is not enough disk space, or another process has deleted the metadata file.
55	run_polar_metadata.bash	The script could not change the field value GRAN_NAME in a metadata file, the utility ‘sed’ failed.
56	get_normal_tile_ul_lr_coord.bash	The script failed to calculate the corner coordinates of a non-polar tile.
57	get_polar_bounds.bash	The wrong polar region keyword was given to the script.
58	get_tileid_range.bash	The wrong projection type keyword was given to the script.

Error Nr	Script name	Comment
59	interp_colsrows.bash	The IDL scripts interp_colrow.pro failed.
60	interp_colsrows.bash	The column output file of IDL scripts interp_colrow.pro does not exist.
61	interp_colsrows.bash	The row output file of IDL scripts interp_colrow.pro does not exist.
62	make_image_mosaic.bash	The wrong mosaic type keyword was given to the script.
63	process_granule_tiles.bash	The event file that was given to the script does not exist. It could be that another process deleted the event file.
64	process_granule_tiles.bash	The event file that was given to the script has an unknown satellite name. The script will not be able to process the event file.
65	process_polar_subsets.bash	The ImageMagick ‘convert’ executable failed to convert the Northern polar mosaic GeoTIFF image into a JPEG image.
66	process_polar_subsets.bash	The ImageMagick ‘convert’ executable failed to convert the Northern polar mosaic GeoTIFF image into a lower resolution JPEG image.
67	reproject_ms2gt_mix.bash	The MS2GT fornav executable binary failed to project the sensor zenith angle data onto a 250m projection grid.
68	reproject_ms2gt_qkm.bash	The MS2GT fornav executable binary failed to project the sensor zenith angle data onto a 250m projection grid.
69	run_reproject_ms2gt.bash	The metadata file that was given to the script does not exist.
70	run_reproject_ms2gt.bash	The incorrect projection type parameters was given to the script. The script calling run_reproject_ms2gt.bash is feeding it incorrect parameters. If the script was executed manually, make double sure what parameters were passed to the script.
71	run_reproject_ms2gt.bash	The script band_comb_type.bash failed to determine highest resolution of the output imagery can be based on the given sensor band numbers.
72	process_granule_polar_tiles.bash	The metadata file corresponding to the event file that was given to the script, does not exist. Either there is not enough disk space or another process has deleted the metadata file.

Error Nr	Script name	Comment
73	run_polar_metadata.bash	The polar region that was given to the script is incorrect. Either a script is feeding incorrect parameters to the this script. If this script was executed manually, make double sure what parameters were passed to the script.
74	run_polar_metadata.bash	The IDL script write_polar_metadata.pro failed.
75	process_polar_subsets.bash	The ImageMagick ‘convert’ executable failed to convert the Southern polar mosaic GeoTIFF image into a JPEG image.
76	process_polar_subsets.bash	The ImageMagick ‘convert’ executable failed to convert the Southern polar mosaic GeoTIFF image into a lower resolution JPEG image.
77	make_image_mosaic.bash	The script “sort_comp_img_filelst.bash” failed to sort a given text file.
78	make_image_mosaic.bash	The ImageMagick ‘convert’ executable failed to scale a tile image.
79	make_image_mosaic.bash	The ImageMagick ‘convert’ executable failed to append a column image to a row mosaic image.
80	make_image_mosaic.bash	The ImageMagick ‘convert’ executable failed to scale a tile image.
81	make_image_mosaic.bash	The ImageMagick ‘convert’ executable failed to append a row image to a previous row image.
82	make_image_mosaic.bash	The script could not duplicate either the polar or non-polar GeoTIFF metadata file. Either there is not enough disk space, or the GeoTIFF template files do not exist. If the $MIPS_HOME/templates directory is empty, re-install MIPS.
83	make_image_mosaic.bash	The geotifcp executable binary failed to create a GeoTIFF image.
84	process_subsets.bash	The BASH script make_cut_subset.bash did not create a text file called ‘idl_output.txt’. The IDL script make_cut_subset.pro probably failed, and the script make_cut_subset.bash did not catch the error.
85	process_subsets.bash	The geotifcp executable binary failed to create a GeoTIFF image.
86	process_subsets.bash	The ImageMagick ‘convert’ executable failed to convert a GeoTIFF image into a JPEG image.
87	process_subsets.bash	The read-only subset configuration file is missing. Check in either the mips_modis_env.bash or mips_mersi_env.bash files if the variable MIPS_RO_SUBSET_CONF points to a file that exist.

Error Nr	Script name	Comment
88	process_global_mosaic.bash	The ImageMagick ‘convert’ executable failed to convert the global mosaic GeoTIFF image into a JPEG image.
89	filter_polar_data_edge.bash	The IDL script filter_polar_data_edge.pro failed.
90	run_reproject_ms2gt.bash	The script failed to duplicate the GeoTIFF metadata file.
91	run_reproject_ms2gt.bash	ImageMagick failed to convert the PNG image into a TIFF image.
92	run_reproject_ms2gt.bash	The geotifcp executable binary failed to create a GeoTIFF image.
93	process_polar_mosaic.bash	The script failed to create a 1000m resolution Arctic polar mosaic image.
94	process_polar_mosaic.bash	The script failed to create a 4000m resolution Arctic polar mosaic image.
95	process_polar_mosaic.bash	The script failed to create a 1000m resolution Antarctic polar mosaic image.
96	process_polar_mosaic.bash	The script failed to create a 4000m resolution Antarctic polar mosaic image.
97	make_tile_composite_zangle_nr.bash	The source GeoTIFF does not exist, even though MIPS is configured to create GeoTIFF files.
98	make_tile_composite_zangle_nr.bash	The binary listgeo failed to extract the GeoTIFF information from the source GeoTIFF file.
99	make_tile_composite_zangle_nr.bash	The binary geotifcp failed to create a GeoTIFF.