SPOC v.3.02A

Generated by Doxygen 1.8.9.1

Fri May 29 2015 10:39:58

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 COMMON/ATTACH.f File Reference

Functions/Subroutines

- subroutine attach (LMKNM0, LMKNM1, NTMP, AUTOFLAG)

Procedure.

2.1.1 Function/Subroutine Documentation

2.1.1.1 subroutine attach (character∗6 *LMKNM0,* character∗6 *LMKNM1,* integer∗4 *NTMP,* logical *AUTOFLAG*)

Procedure.

Parameters

ntmp	Abstract

```
This subroutine determines the relative central vectors of two
overlapping maplets and populates the MAP OVERLAPS portion of the
.LMK file.

     ------------------         ------------------
     |  *         o   |         |  o         ~   |
     |          x     |         |x                |
     |     V .        |     V1 .|    V1 .    *   |
     | @        O     |         |O       o        |
     |                |         |                 |
     ------------------         ------------------
           LMK0                       LMK1
We are working with a landmark LMK0 and we determine that its central
vector in the body-fixed frame is V.

There may be another landmark, LMK1 that shares common topography with
LMK0 and whose central vector is determined to be V1.

In order to compare these directly, we project LMK1 into the coordinate
system of LMK0 (V,Ux,Uy,Uz) with GET_MAP(LMK1,NTMP,QSZ,V,UX,UY,UZ,S0,..).

In a perfect world, the topograpny of LMK1 in this frame would match up
perfectly with the toography of LMK0, and the relative positions V and
features in the second box would be the same as V and features in the
first box.

The subroutine offers a choice of three correlation schemes to determine
how many LMK0 pixels and lines the window surrounding the LMK1 projection
must be moved in order for the topographi!!features to line up.  Then V
is changed to V + S0(dpUy+dlUx).  This is where LMK1 thinks that the
center of LMK0 really is.

The new V is then used as the LMK0 center to repeat the process several
times.   The solved shift between the two landmarks is then inserted into
LMKNM0.LMK as a MAP OVERLAP with the subroutine INSERTINLMK_LMK.  The
shift is in the LMKNM1 frame.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM0 I Name of landmark to be populated LMKNM1 I Name of overlapping landmark NTMP I Largest allowable maplet pixel range -NTMP,NTMP. AUTOFLAG I If .T./.F. will perform automatic/manual determination

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I tmpl.gray I tmpl.pgm O

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELETEINLMK_LMK GET_MAP INSEZRZTINLMK_LMK RAW2PGM READ_MAP

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called VEQU VSUB

Called_by_SPC_Programs LITHOS

.....

..... GRADIENT CORRELATION .....

.....

..... MANUAL CORRELATION .....

.....

2.2 COMMON/ATTACH_LO.f File Reference

Functions/Subroutines

- subroutine attach_lo (LMKNM0, LMKNM1, NTMP)

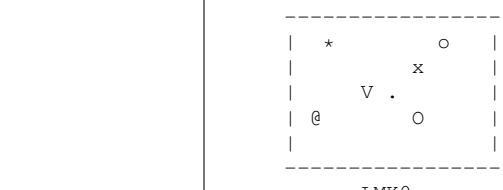
  Procedure.

2.2.1 Function/Subroutine Documentation

2.2.1.1 subroutine attach_lo (character∗6 *LMKNM0,* character∗6 *LMKNM1,* integer∗4 *NTMP*)

Procedure.

Parameters

ntmp	Abstract This subroutine determines the relative central vectors of two overlapping maplets and populates the MAP OVERLAPS portion of the .LMK file. It uses a LOCKOUT file to prevent reading/writing to any open LMKFILE.  We are working with a landmark LMK0 and we determine that its central vector in the body-fixed frame is V. There may be another landmark, LMK1 that shares common topography with LMK0 and whose central vector is determined to be V1. In order to compare these directly, we project LMK1 into the coordinate system of LMK0 (V,Ux,Uy,Uz) with GET_MAP(LMK1,NTMP,QSZ,V,UX,UY,UZ,S0,..). In a perfect world, the topograpny of LMK1 in this frame would match up perfectly with the toography of LMK0, and the relative positions V and features in the second box would be the same as V and features in the first box. The subroutine offers a choice of three correlation schemes to determine how many LMK0 pixels and lines the window surrounding the LMK1 projection must be moved in order for the topographi!!features to line up. Then V is changed to V + S0(dpUy+dlUx). This is where LMK1 thinks that the center of LMK0 really is. The new V is then used as the LMK0 center to repeat the process several times. The solved shift between the two landmarks is then inserted into LMKNM0.LMK as a MAP OVERLAP with the subroutine INSERTINLMK_LMK. The shift is in the LMKNM1 frame.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM0 I Name of landmark to be populated LMKNM1 I Name of overlapping landmark NTMP I Largest allowable maplet pixel range -NTMP,NTMP. AUTOFLAG I Alwasy .TRUE. (automati!!mode)

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I ./TESTFILES/<LMKNM0>.MAP O Lockout file for <LMKNM0>.MAP ./tmpl.gray I ./tmpl.pgm O

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY DELETEINLMK_LMK_LO READ_MAP GET_MAP_LO INSERTINLMK_LMK_LO

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called VEQU VSUB

Called_by_SPC_Programs LITHOSP

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.3 COMMON/BRT1.f File Reference

Functions/Subroutines

- double precision function brt1 (Z, Z0, KK, DN, ESZ)

  Procedure.

2.3.1 Function/Subroutine Documentation

2.3.1.1 double precision function brt1 (double precision, dimension(2) *Z,* integer, dimension(2) *Z0,* integer *KK,* integer, dimension(-1024:1024,-1024:1024) *DN,* integer *ESZ*)

Procedure.

Parameters

dn	Abstract

This function uses the image array extracted by the subroutine PICINPT to determine the brightness value at pixel/line location [Z(1),Z(2)] in the original image. It determines the four surrounding pixels in the image array dn and finds the value BRT1 at the pont of interest through bilinear interpolation.

Disclaimer

Abondon hope, ye who enter here.

Required_Reading

NONE

Declarations Variable_I/O

Variable I/O Description

DN I Array of integer brightness values from PICINPT Z0 I PICINPT extraction defining parameter KK I PICINPT extraction defining parameter ESZ I PICINPT extraction defining parameter Z I Location of point of interest in original image PICNM BRT1 O Brightness value at point of interest

File_I/O

Filename I/O Descrription

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

2.4 COMMON/CLEAR.f File Reference

Functions/Subroutines

- subroutine clear (PICNM)

  Procedure.

2.4.1 Function/Subroutine Documentation

2.4.1.1 subroutine clear (character∗12 *PICNM*)

Procedure.

Abstract This subroutine will clear out all LANDMARKS and LIMB entries in the image PICNM.SUM summary file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Base name of the picture (image) summary file to be modified or cleared. File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM O Image summary file

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs

2.5 COMMON/COMPARE.f File Reference

Functions/Subroutines

- subroutine compare (NTMP, PMX, DN, SP, QSZ, NSZ, K1, K2)

  Procedure.

2.5.1 Function/Subroutine Documentation

2.5.1.1 subroutine compare (integer *NTMP,* integer *PMX,* real∗4, dimension(5000,5000) *DN,* double precision, dimension(3,pmx) *SP,* integer *QSZ,* integer *NSZ,* integer *K1,* integer *K2*)

Procedure.

Parameters

ntmp	Abstract This subroutine creates an image file of two chosen maplet image extractions to be used to determine the surface normal by fitting a plane through three common points. The image ectraction can be as a regular brightness display of as an elevation independent gradient display. The calling procedure in LITHOS (option N/4) is rarely used but is very valuable when needed.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

pmx	Abstract This subroutine creates an image file of two chosen maplet image extractions to be used to determine the surface normal by fitting a plane through three common points. The image ectraction can be as a regular brightness display of as an elevation independent gradient display. The calling procedure in LITHOS (option N/4) is rarely used but is very valuable when needed.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP I PMX I DN I SP QSZ NSZ K1 K2

File_I/O

Filename I/O Description

COMPARE.gray I&O

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called

SPC_subroutines_called HGT2SLP RAW2PGM

SPICELIB_functions_called

SPICELIB_subroutines_called

Called_by_SPC_Programs LITHOS

2.6 COMMON/CORNERS.f File Reference

Functions/Subroutines

- subroutine corners (SZ, I1, J1, F1, I2, J2, F2, I3, J3, F3)

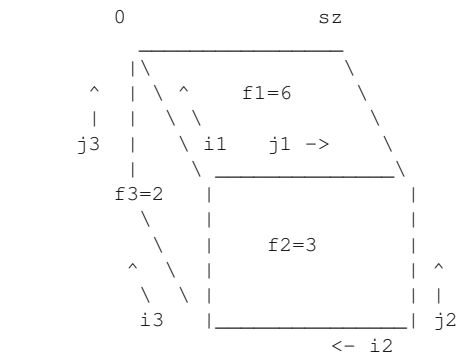
  Procedure.

2.6.1 Function/Subroutine Documentation

2.6.1.1 subroutine corners (integer∗4 *SZ,* integer∗4 *I1,* integer∗4 *J1,* integer∗4 *F1,* integer∗4 *I2,* integer∗4 *J2,* integer∗4 *F2,* integer∗4 *I3,* integer∗4 *J3,* integer∗4 *F3*)

Procedure.

Parameters

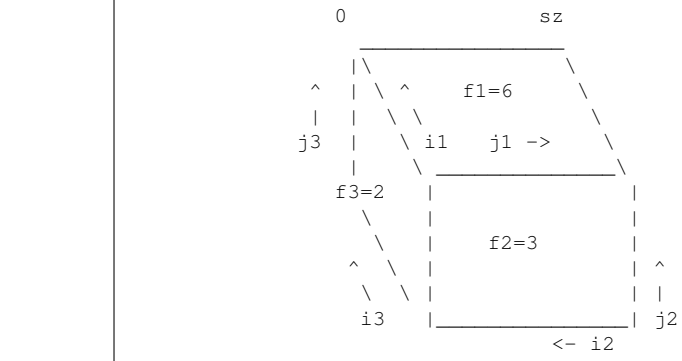
sz	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

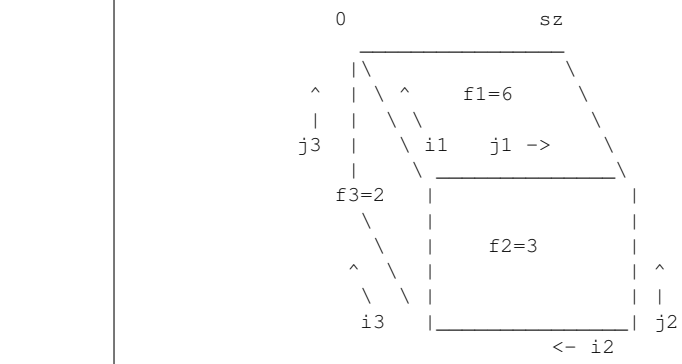
i1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

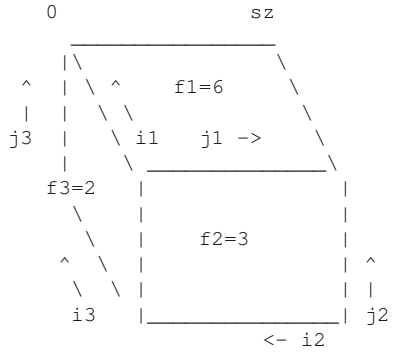
j1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

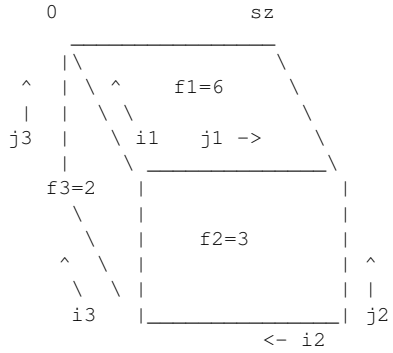
f1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

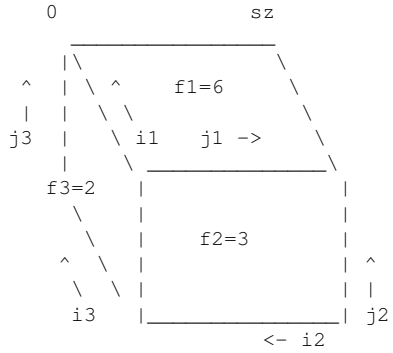
i2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

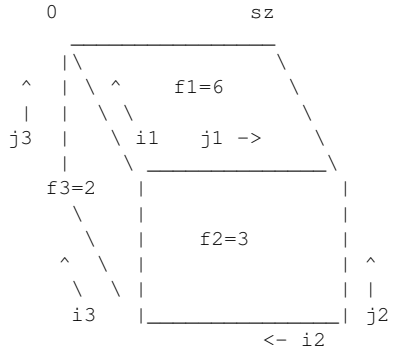
j2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

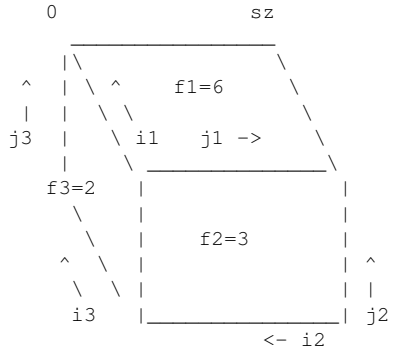
f2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

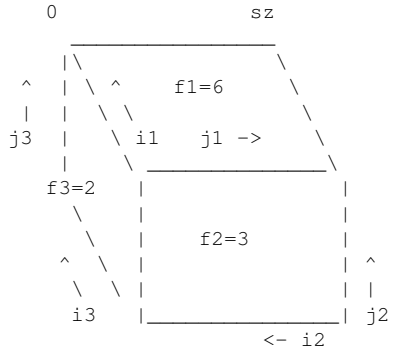
i3	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

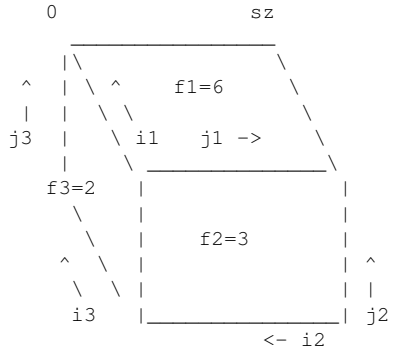
j3	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

f3	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each corner of the cube contains common points from three faces. The purpose of this subroutine is to identify which points on faces sharing a common corner match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0, j=0 on face 6, CORNERS(SZ, 0,0,6,I2,J2,F2,I3,J3,F3) would return I2=SZ, J2=SZ on face F2=3 and I3=0, J3=SZ on face F3=2 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

SZ I1 J1 F1 I2 J2 F2 I3 J3 F3

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called

SPC_subroutines_called

SPICELIB_functions_called

SPICELIB_subroutines_called

Called_by_SPC_Programs BLEMISHES VEC2SHAPE DENSIFY DENSIFYA SUBROUTINE RELAXATION

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.7 COMMON/CREATE_LMFILE.f File Reference

Functions/Subroutines

- subroutine create_lmfile (NTMP, LMKNM)
  Procedure.

2.7.1 Function/Subroutine Documentation

2.7.1.1 subroutine create_lmfile (integer *NTMP*, character∗6 *LMKNM*)

Procedure.

Parameters

ntmp	Abstract This subroutine creates an initial landmark (LMKNM.LMK) file. The location of the landmark center is specified by a px/ln location in an image, by a latitude and (west) longitude on the body, or by its location in a high resolution local map (bigmap). The initial orientation is determined by the shape model in the first two cases and from the bigmap in the last. The landmark's scale (km/px) and qsz (maplet size = 2∗qsz+1 x 2∗qsz+1) are also specified. Once the landmark file has been created, it must be input (I) into LITHOS for extraction of imaging data and creation of an associated maplet.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP I LMKNM I Landmark to be created.

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I LMRKLIST.TXT O LMRKLIST0.TXT O LMRKLIST1.TXT O LMKFILES/<LMKNM>/.LMK O SUMFILES/<PICNM>.SUM I

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called GET_HEIGHTS IMGPL2VN ORIENT PATCH_COORDS READ_MAP V2IMGPL U2VN

SPICELIB_functions_called VDOT RPD

SPICELIB_subroutines_called LATREC VMINUS

Called_by_SPC_Programs LITHOS

2.8 COMMON/DARNER.f File Reference

Functions/Subroutines

- subroutine darner (NTMP, QSZ, HT, ALB)

  Procedure.

2.8.1 Function/Subroutine Documentation

2.8.1.1 subroutine darner (integer *NTMP,* integer *QSZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

Procedure.

Parameters

ntmp	Abstract If there is missing data at some point in a maplet, this subroutine will fill it by interpolating from neighboring points. For example, if data is missing at point 1, the average value from points 0 and 3 will be assigned to the central point (.) and the value at 1 will be chosen to make the average of 1 and 2 be the same. 0 1 x o . x x 2 3 Apparently, this subroutine is no longer used.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ HT ALB

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.9 COMMON/DEBLEMISH.f File Reference

Functions/Subroutines

- subroutine deblemish (PICNM, Z0, KK, DN, ESZ)

  Procedure.

2.9.1 Function/Subroutine Documentation

2.9.1.1 subroutine deblemish (character*12 *PICNM,* integer, dimension(2) *Z0,* integer *KK,* integer, dimension(-1024:1024,-1024:1024) *DN,* integer *ESZ*)

Procedure.

Abstract

This subroutine locates pixels in an image that have been flagged as bad in the image's blemish (.BLM) file and sets those to zero (no data) in the image extraction process. In this way, the original image data is not altered. Some blemish files can be in the form of a 12 character template, defined in INIT_LITHOS.TXT by a record BL↩ EMISH='template'. For example, DAWN has two framing cameras with image names starting with FC1 and FC2. FC2 images all have several blemishes in the same places. A template specified by BLEMISH='FC2#########' will correct all of these images without having thousands of individual blemish files. If an image contains additional blemishes it can have its own file with just those additional blemishes. The template option has been tested with ROSETTA images of Lutetia.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Z0 I KK I DN I ESZ I

File_I/O

Filename I/O Description

BLEMISHES/<PICNM>.BLM I

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS LITHOSP AUTOREGISTER AUTOREGISTERP History 2013_09_15↩ : Changed to handle both LITHOS and AUTOREGISTER. 2014_02_12: Changed to include common TEM↩ PLATE.BLM file.

2.10 COMMON/DEFU.f File Reference

Functions/Subroutines

- subroutine defu (U)

Procedure.

2.10.1 Function/Subroutine Documentation

2.10.1.1 subroutine defu (integer∗4, dimension(3,3,6) *U*)

Procedure.

Parameters

u	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. This subroutine defines the orientation of the labeled points on the cube. U(i,j,f) defines the equivalence and direction of the ith component of face f of the cube in terms of spatial components j. U(i, j, f) = sign of grid space face space component component component For example: U(1,1,2)=1 U(2,3,2)=-1 U(3,2,2)=-1 says that for face 2 of the cube, the first component (x) is in the +X spatial direction, the second component (y) is in the -Z spatial direction and the third component (z), normal to the plane is in the -Y spatial direction. _____ x = X /| / | / | face f / | / | z=-Y y=-Z Notice that the grid coordinates form a left-handed coordinate system.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

U O Data matrix that is initialized by routine.

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs TRIAX VECS2SHAPE

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.11 COMMON/DELAY.f File Reference

Functions/Subroutines

- subroutine delay ()
  Procedure.

2.11.1 Function/Subroutine Documentation

2.11.1.1 subroutine delay ( )

Procedure.

Abstract

This subroutine is used in parallel routines to determine when a process should check to see whether it is safe to open a previously locked out file. The signal to look again is tied to the system clock. It is also tied to the USR designation of the process, so two competing processes willl not emerge from hibernation at the same time and try to open the same file.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

USRMX I J1 I J2 I J3 I

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File that contains the SP!! toolkit initialization data

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

History

2.12 COMMON/DELETE_LMFILE.f File Reference

Functions/Subroutines

- subroutine delete_lmfile (LMKNM)

  Procedure.

2.12.1 Function/Subroutine Documentation

2.12.1.1 subroutine delete_lmfile (character∗6 *LMKNM*)

Procedure.

Abstract This routine will delete the landmark file specified by the input variable <LMKNM> and any references to the landmark in the summary and map files.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Name of landmark to be deleted.

File_I/O

Filename I/O Description

LMRKLIST I List of sum and maplet files LMRKLIST0 I List of sum and maplet files LMRKLIST1 I List of sum and maplet files <LMKNM>.LMK 0 Landmark file to be deleted.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELETEINPIC_LIM DELETEINPIC_LMK

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.13 COMMON/DELETEINLMK_LIM.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_lim (LMKNM, PICNM)

  Procedure.

2.13.1 Function/Subroutine Documentation

2.13.1.1 subroutine deleteinlmk_lim (character∗6 *LMKNM,* character∗12 *PICNM*)

Procedure.

Abstract This subroutine will delete image entry <PICNM> from the limb fit section of the landmark file <LMKN↩ M>.LMK

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark name PICNM I Image name

File_I/O

Filename I/O Description

<LMKNM>.LMK O Landmark file to be edited

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP SUBROUTINE DETACH SUBROUTINE DISCONNECT_LMFILE SUBRO↩
UTINE LIMB_HEIGHTS SUBROUTINE TUCK

2.14 COMMON/DELETEINLMK_LIM_LO.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_lim_lo (LMKNM, PICNM)

  Procedure.

2.14.1 Function/Subroutine Documentation

2.14.1.1 subroutine deleteinlmk_lim_lo (character∗6 *LMKNM,* character∗12 *PICNM*)

Procedure.

Abstract This subroutine will delete image entry <PICNM> from the limb fit section of the landmark file <LMKN↩
M>.LMK

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark name PICNM I Image name

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK O Landmark file to be edited

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE DISCONNECT_LMFILE_LO SUBROUTINE LIMB_HEIGHTS_LO

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.15 COMMON/DELETEINLMK_LMK.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_lmk (LMKNM1, LMKNM2)

  Procedure.

2.15.1 Function/Subroutine Documentation

2.15.1.1 subroutine deleteinlmk_lmk (character∗6 *LMKNM1,* character∗6 *LMKNM2*)

Procedure.

Abstract This subroutine will delete the map overlap landmark entry LMKMN2 from the landmark file LMKNM1.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM1 I Base Name of the Landmark File LMKNM2 I Name of the Map Oeverlay landmark entry to be deleted

File_I/O

Filename I/O Description

<LMKNM1>.LMK O Lankmark file to be updated

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS LITHOSP SUBROUTINE ATTACH SUBROUTINE DETACH DISCONNECT↩ _LMFILE

2.16 COMMON/DELETEINLMK_LMK_LO.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_lmk_lo (LMKNM1, LMKNM2)

  Procedure.

2.16.1 Function/Subroutine Documentation

2.16.1.1 subroutine deleteinlmk_lmk_lo (character∗6 *LMKNM1,* character∗6 *LMKNM2*)

Procedure.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM1 I Base Name of the Landmark File LMKNM2 I Name of the Map Oeverlay landmark entry to be deleted

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM1>.LMK O Lockout file for <LMKNM1>.LMK ./LMKFILES/<LMKNM1>.LMK O Lankmark file to be updated

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.17 COMMON/DELETEINLMK_PIC.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_pic (LMKNM, PICNM)

  Procedure.

2.17.1 Function/Subroutine Documentation

2.17.1.1 subroutine deleteinlmk_pic (character∗6 *LMKNM,* character∗12 *PICNM*)

Procedure.

Abstract This subroutine searches for the image <PICNM> in the Pictures section for the landmark file <LMKN↩ M>.LMK and deletes it if it exists.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Basename of the landmark file PICNM I Image name to be deleted from the landmark file

File_I/O

Filename I/O Description

<LMKNM>.LMK O Landmark file name to edit

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP

2.18 COMMON/DELETEINLMK_PIC_LO.f File Reference

Functions/Subroutines

- subroutine deleteinlmk_pic_lo (LMKNM, PICNM)

  Procedure.

2.18.1 Function/Subroutine Documentation

2.18.1.1 subroutine deleteinlmk_pic_lo (character∗6 *LMKNM,* character∗12 *PICNM*)

Procedure.

Abstract This subroutine searches for the image <PICNM> in the Pictures section for the landmark file <LMKN↩ M>.LMK and deletes it if it exists.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Basename of the landmark file PICNM I Image name to be deleted from the landmark file

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK O Landmark file name to edit

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP

History 2013_10_17: Capitalization, compiler warnings fixed and header added.

2.19 COMMON/DELETEINNOM_TRJ.f File Reference

Functions/Subroutines

- subroutine deleteinnom_trj (PICNM)

  Procedure.

2.19.1 Function/Subroutine Documentation

2.19.1.1 subroutine deleteinnom_trj (character∗12 *PICNM*)

Procedure.

Abstract The subroutine will delete the last four rows of the nominal file <PICNM>.NOM. These rows consist of the information of the surrounding four images of the image <PICNM>.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Base name of the nominal file to be edited

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be edited

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DYNAMICS

2.20 COMMON/DELETEINPIC_LIM.f File Reference

Functions/Subroutines

- subroutine deleteinpic_lim (PICNM, LMKNM)

  Procedure.

2.20.1 Function/Subroutine Documentation

2.20.1.1 subroutine deleteinpic_lim (character∗12 *PICNM,* character∗6 *LMKNM*)

Procedure.

Abstract The subroutine will delete the landmark <LMKNM> from the Limb Fits section of the image summary file <PICNM>.SUM.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICMN I Image name of the summary file LMKNM I Landmark to be removed from summary

File_I/O

Filename I/O Description

<PICNM>.SUM O Image summary file to be edited

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS LITHOSP SUBROUTINE DELETE_LMFILE SUBROUTINE DETACH SUBR↩ OUTINE DISCONNECT_LMFILE LIMB_HEIGHTS

2.21 COMMON/DELETEINPIC_LIM_LO.f File Reference

Functions/Subroutines

- subroutine deleteinpic_lim_lo (PICNM, LMKNM)

  Procedure.

2.21.1 Function/Subroutine Documentation

2.21.1.1 subroutine deleteinpic_lim_lo (character∗12 *PICNM,* character∗6 *LMKNM*)

Procedure.

Abstract The subroutine will delete the landmark <LMKNM> from the Limb Fits section of the image summary file <PICNM>.SUM.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICMN I Image name of the summary file LMKNM I Landmark to be removed from summary

File_I/O

Filename I/O Description

./TESTFILES/<PICNM>.SUM O Lockout file for <PICNM>.SUM ./SUMFILES/<PICNM>.SUM O Image summary file to be edited

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE DISCONNECT_LMFILE_LO SUBROUTINE LIMB_HEIGHTS_LO

History 2013_10_16: Capitalization, compiler warnings fixed and header added.

2.22 COMMON/DELETEINPIC_LMK.f File Reference

Functions/Subroutines

- subroutine deleteinpic_lmk (PICNM, LMKNM)

  Procedure.

2.22.1 Function/Subroutine Documentation

2.22.1.1 subroutine deleteinpic_lmk (character∗12 *PICNM,* character∗6 *LMKNM*)

Procedure.

Abstract The subroutine will delete the landmark <LMKNM> from the LANDMARKS section of the image summary file <PICNM>.SUM.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Base name of the summary LMKNM I Landmark to be removed from the Landmark section

File_I/O

Filename I/O Description

<PICNM>.SUM O Image summary to be edited

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP SUBROUTINE DELETE_L↩
MFILE

2.23 COMMON/DENOISE.f File Reference

Functions/Subroutines

- subroutine denoise (PICNM, DN)

  Procedure.

2.23.1 Function/Subroutine Documentation

2.23.1.1 subroutine denoise (character∗12 *PICNM,* integer, dimension(-128:128,-128:128) *DN*)

Procedure.

Abstract

This subroutine removes speckle from the images. If the central pixel in a 3x3 square deviates from the average by more than two standard deviations, it replaces that pixel's dn (data number) with the average value. The subroutine is not used by default, but can be turned on from INIT_LITHOS.TXT with a DENOISE=.TRUE. entry.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I DN I

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS SUBROUTINE LIMB_HEIGHTS

2.24 COMMON/DETACH.f File Reference

Functions/Subroutines

- subroutine detach (LMKNM, XNAME)

  Procedure.

2.24.1 Function/Subroutine Documentation

2.24.1.1 subroutine detach (character∗6 *LMKNM,* character∗12 *XNAME*)

Procedure.

Parameters

lmknm	Abstract This subroutine will detach or disassociate a particular landmark or image <XNAME> with the landmark file <LMKNM>.LMK

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Base name of the landmark file to be modified. XNAME I Either the name of the landmark or image to be detached or removed from landmark file.

File_I/O

Filename I/O Description

<LMKNM>.LMK O Landmark file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELETEINLMK_LIM DELETEINLMK-LMK DELETEINPIC_LIM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.25 COMMON/DISCONNECT_LMFILE.f File Reference

Functions/Subroutines

- subroutine disconnect_lmfile (LMKNM, SEARCH)

  Procedure.

2.25.1 Function/Subroutine Documentation

2.25.1.1 subroutine disconnect_lmfile (character∗6 *LMKNM,* character∗72 *SEARCH*)

Procedure.

Abstract

This procedure removes all MAP OVERLAPS from a LMKFILE and removes the landmark as a MAP OVERLAP from all other LMKFILEs. The procedure also clears out the LIMB FITS portion of the LMKFILE that contains images in which the landmark maplet appears on a limb and clears LMKNM from the LIMB FITS portion of the corresponding SUMFILE.

The routine reads through the input file <SEARCH> which is either the LMRKLIST.TXT or OVERLAPS.TXT containing a list of landmarks that will be disconnected from the input landmark <LMRKNM> using the subroutine DELETEINLMK_LMK. DELETEINLMK_LIM_LO and DELETEINPIC_LIM_LO are used to clear all images from the LIMB FITS portion of LMKNM.LMK and LMKNM from the LIMB FITS portion of all SUMFILES that contained it.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark Name SEARCH I Name of file to search, either LMRKLIST.TXT or OVERLAPS.TXT

File_I/O

Filename I/O Description

<SEARCH> I Summary or landmark file that is searched

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELETEINLMK_LIM DELETEINLMK_LMK DELETEINPIC_LIM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS LITHOSP

2.26 COMMON/DISCONNECT_LMFILE_LO.f File Reference

Functions/Subroutines

- subroutine disconnect_lmfile_lo (LMKNM, LDIM, LMKMX, LNAME, LSZ, LSCL, LV)

  Procedure.

2.26.1 Function/Subroutine Documentation

2.26.1.1 subroutine disconnect_lmfile_lo (character∗6 *LMKNM,* integer *LDIM,* integer *LMKMX,* character∗6, dimension(ldim) *LNAME,* integer, dimension(ldim) *LSZ,* double precision, dimension(ldim) *LSCL,* double precision, dimension(3,ldim) *LV*)

Procedure.

Parameters

ldim	Abstract This procedure removes all MAP OVERLAPS from a LMKFILE and removes the landmark as a MAP OVERLAP from all other LMKFILEs. The procedure also clears out the LIMB FITS portion of the LMKFILE that contains images in which the landmark maplet appears on a limb and clears LMKNM from the LIMB FITS portion of the corresponding SUMFILE. It uses a LOCKOUT file to prevent reading/writing to any open LMKFILE or SUMFILE. LITHOSP, that uses this subroutine, reads the file LMRKLISTX.TXT that contains ancillary data for each landmark. Because this file can be processed quickly, it is possible to determine which landmarks might overlap LMKNM the input file and to use the subroutine DELETEINLMK_LMK_LO to remove LMKNM from the MAP OVERLAPS section of their LMKFILES if it exists. All entries in the MAP OVERLAPS section of LMKNM.LMK are removed. DELETEINLMK_LIM_LO and DELETEINPIC_LIM_LO are used to clear all images from the LIMB FITS portion of LMKNM.LMK and LMKNM from the LIMB FITS portion of all SUMFILES that contained it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Landmark Name LDIM I Maximum number of landmarks (set in calling program) LMKMX I Size of landmark array LNAME I Name array for all landmarks LSZ I Landmark size array for all landmarks LSCL I Scale of each landmark in array LV I Central vector for each landmark in array

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK I Landmark file that is searched

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY DELETEINLMK_LMK_LO DELETEINLMK_LIM_LO DELETEINPIC_LIM_LO

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs LITHOSP

History 2013_10_16: Capitalization, compiler warnings fixed and header added

2.27 COMMON/DISPLAY1.f File Reference

Functions/Subroutines

- subroutine display1 (DN, QSZ, NSZ)

  Procedure.

2.27.1 Function/Subroutine Documentation

2.27.1.1 subroutine display1 (real∗4, dimension(5000,5000) *DN,* integer *QSZ,* integer *NSZ*)

Procedure.

Abstract

The extracted imaging data for the npix images containing a maplet and the corresponding predicted data are held in a square, PICSZ X PICSZ image array (real∗4, [0,1]) where PICSZ=2∗(QSZ+1)∗NSZ and where NSZ = 2∗(INT(SQRT((NPIX-0.5)/2))+1). Note that NSZ is computed and returned by the subroutine EXTRACT_DATA.

This subroutine produces an 8 bit/pixel .pgm image file to allow the user to examine the current state of the fit. We usually take QSZ=49. Since there is a one pixel gap between each block of the display, identical features are then separated by multiples of 100 pixels, making manual alignment trivial.

The subroutine now produces two displays, LMRK_DISPLAY0.pgm with the data as used in LITHOS and LMRK↩ _DISPLAY1.pgm in which the data for each image is maximally stretched. Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

DN I QSZ I NSZ I

File_I/O

Filename I/O Description

LMRK_DISPLAY1.gray O

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called RAW2PGM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs STEREO

History 2014_02_22: Produces both unstretched and stretched displays.

2.28 COMMON/DN2PGM.f File Reference

Functions/Subroutines

- subroutine dn2pgm (DN, OUTFILE, KB)

  Procedure.

2.28.1 Function/Subroutine Documentation

2.28.1.1 subroutine dn2pgm (integer, dimension(-1024:1024,-1024:1024) *DN,* character∗72 *OUTFILE,* integer∗4 *KB*)

Procedure.

Parameters

dn	Abstract `This subroutine is specifi!!to the program AUTOREGISTER. It produces a .pgm copy of the image data so the user can have a quick reference to it.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

DN I OUTFILE I KB I

File_I/O

Filename I/O Description

<OUTFILE> O TEMPFILE.pgm from AUTOREGISTER

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER

2.29 COMMON/EXTEND.f File Reference

Functions/Subroutines

- subroutine extend (SZ, I1, J1, F1, I2, J2, F2)

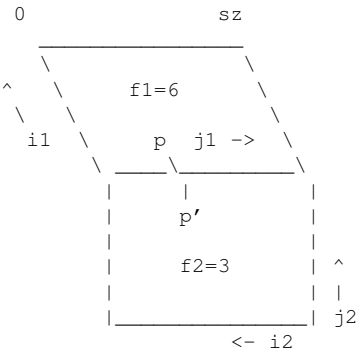
  Procedure.

2.29.1 Function/Subroutine Documentation

2.29.1.1 subroutine extend (integer∗4 *SZ,* integer∗4 *I1,* integer∗4 *J1,* integer∗4 *F1,* integer∗4 *I2,* integer∗4 *J2,* integer∗4 *F2*)

Procedure.

Parameters

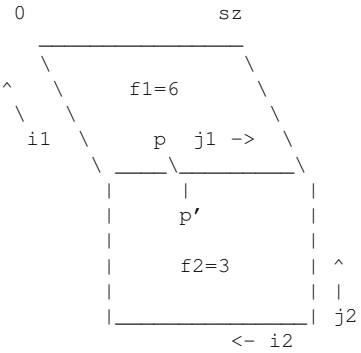
sz	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

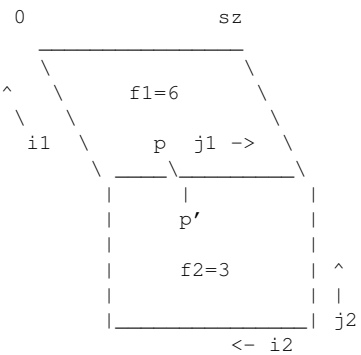
i1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

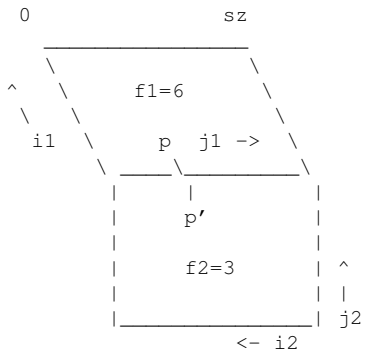
j1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

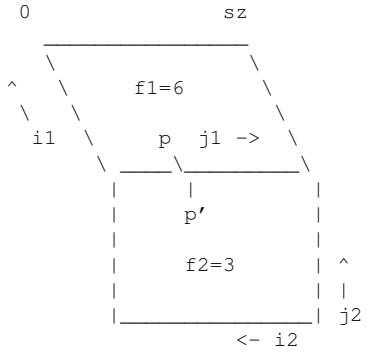
f1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

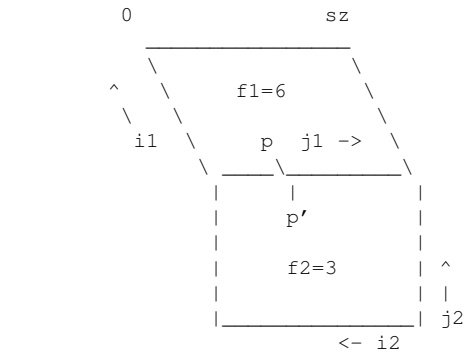
i2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

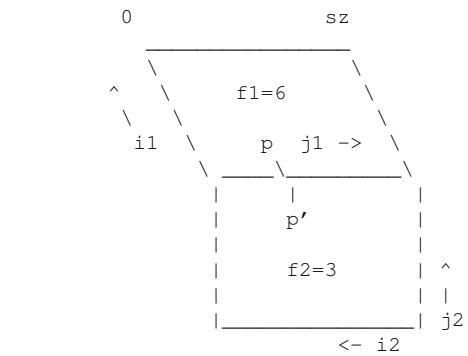
j2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

f2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. SZ is the size of a grid, with i,j ranging from 0 to SZ. If an index moves slightly outside that range on one face, the point will correspond to a grid point on another face. For example, if the point p at i1,j1 on face 6 below is moved to a negative value of i1, The new point p' will be on face 3 at coordinates i2=sz-j1, j2=sz+i1: 

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

SZ I I1 I J1 I F1 I I2 O J2 O F2 O

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DENSIFY LITHOS LIMBER SUBROUTINE RELAXATION

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.30 COMMON/EXTRACT_DATA.f File Reference

Functions/Subroutines

- subroutine extract_data (PMX, NTMP, QSZ, SCALE, NPIX, PICID, IPL, V, UX, UY, UZ, HUSE, HT, CP, SP, EUSE, ETOL, DPIC_DLOC, DIDH, DJDH, NSZ, Z0, KK, ESZ, DNK, DN1)

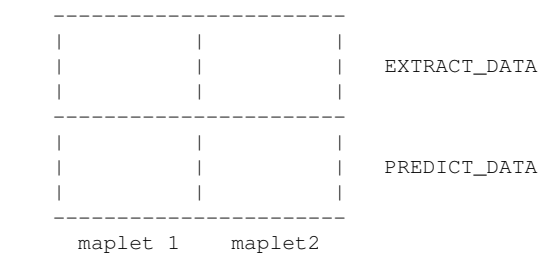
  Procedure.

2.30.1 Function/Subroutine Documentation

2.30.1.1 subroutine extract_data (integer *PMX,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* integer *NPIX,* character∗12, dimension(pmx) *PICID,* double precision, dimension(2,pmx) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(3,pmx) *CP,* double precision, dimension(3,pmx) *SP,* logical *EUSE,* double precision *ETOL,* double precision, dimension(2,3,pmx) *DPIC_DLOC,* double precision, dimension(pmx) *DIDH,* double precision, dimension(pmx) *DJDH,* integer *NSZ,* integer, dimension(2,pmx) *Z0,* integer, dimension(pmx) *KK,* integer *ESZ,* integer, dimension(-128:128,-128:128,pmx) *DNK,* real∗4, dimension(5000,5000) *DN1*)

Procedure.

Parameters

pmx	Abstract This subroutine is a wrapper that takes the images from subroutine EXTRACT_DATA_PI!!and puts orthorectified imaging data into an array for comparison with with the illuminated maplet data supplied by the PREDICT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX O NTMP O QSZ O SCALE O NPIX O PICID I IPL O V O UX O UY O UZ O HUSE O HT O CP O SP O EUSE O ETOL O DPIC_DLO!! O DIDH O DJDH O NSZ O Z0 O KK O ESZ O DNK O DN1 O

File_I/O

Filename I/O Description

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called EXTRACT_DATA_PIC FILLER

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

2.31 COMMON/EXTRACT_DATA_LO.f File Reference

Functions/Subroutines

- subroutine extract_data_lo (PMX, NTMP, QSZ, SCALE, NPIX, PICID, IPL, V, UX, UY, UZ, HUSE, HT, CP, SP, EUSE, ETOL, DPIC_DLOC, DIDH, DJDH, NSZ, Z0, KK, ESZ, DNK, DN1)

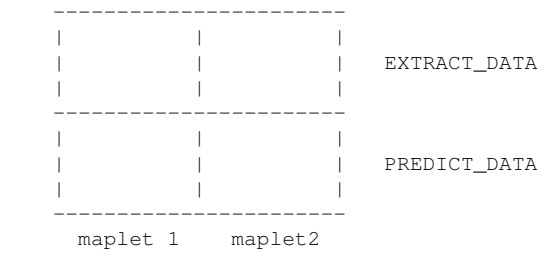
  Procedure.

2.31.1 Function/Subroutine Documentation

2.31.1.1 subroutine extract_data_lo (integer *PMX,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* integer *NPIX,* character∗12, dimension(pmx) *PICID,* double precision, dimension(2,pmx) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(3,pmx) *CP,* double precision, dimension(3,pmx) *SP,* logical *EUSE,* double precision *ETOL,* double precision, dimension(2,3,pmx) *DPIC_DLOC,* double precision, dimension(pmx) *DIDH,* double precision, dimension(pmx) *DJDH,* integer *NSZ,* integer, dimension(2,pmx) *Z0,* integer, dimension(pmx) *KK,* integer *ESZ,* integer, dimension(-128:128,-128:128,pmx) *DNK,* real∗4, dimension(5000,5000) *DN1*)

Procedure.

Parameters

pmx	Abstract This subroutine is a wrapper that takes the images from subroutine EXTRACT_DATA_PI!!and puts orthorectified imaging data into an array for comparison with with the illuminated maplet data supplied by the PREDICT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

This subroutine uses a lockout feature to prevent read/write of some files while they are opened by other processes that may be writing to them.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX O NTMP O QSZ O SCALE O NPIX O PICID I IPL O V O UX O UY O UZ O HUSE O HT O CP O SP O EUSE O ETOL O DPIC_DLO!! O DIDH O DJDH O NSZ O Z0 O KK O ESZ O DNK O DN1 O

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called EXTRACT_DATA_PIC_LO FILLER

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_15: Capitalization, compiler warnings fixed and header added.

2.32 COMMON/EXTRACT_DATA_PIC.f File Reference

Functions/Subroutines

- subroutine extract_data_pic (NTMP, QSZ, SCALE, PICNM, IPL, V, UX, UY, UZ, HUSE, HT, CP, SP, EUSE, ETOL, FINV, DPIC_DLOC, DIDH, DJDH, Z0, KK, ESZ, DN, DNX)

  Procedure.

2.32.1 Function/Subroutine Documentation

2.32.1.1 **subroutine extract_data_pic (integer *NTMP,* integer *QSZ,* double precision *SCALE,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(3) *CP,* double precision, dimension(3) *SP,* logical *EUSE,* double precision *ETOL,* double precision *FINV,* double precision, dimension(2,3) *DPIC_DLOC,* double precision *DIDH,* double precision *DJDH,* integer, dimension(2) *Z0,* integer *KK,* integer *ESZ,* integer, dimension(-1024:1024,-1024:1024) *DN,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *DNX*)**

Procedure.

Parameters

ntmp	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP O QSZ O SCALE O PICNM I SUMFILES/<PICNM>.SUM IPL O V O UX O UY O UZ O HUSE O HT O CP O SP O EUSE O ETOL O FINV O DPIC_DLO!! O DIDH O DJDH O Z0 O KK O ESZ O DN O DNX O

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I Name of summary file to extract data from. Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called BRT1 SLEN

SPC_subroutines_called FIND_PICFILE HGT2SLP LOC2PIX V2IMGPL

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called UCRSS VADD VEQU VHAT History 2014_04_04: Local lower fractional threshold LOCT1 added. Controlled by INIT_LITHOS.TXT. Default is no operation. 2014_04_17: IFF saved.

2.33 COMMON/EXTRACT_DATA_PIC_LO.f File Reference

Functions/Subroutines

- subroutine extract_data_pic_lo (NTMP, QSZ, SCALE, PICNM, IPL, V, UX, UY, UZ, HUSE, HT, CP, SP, EUSE, ETOL, DPIC_DLOC, DIDH, DJDH, Z0, KK, ESZ, DN, DNX)

  Procedure.

2.33.1 Function/Subroutine Documentation

2.33.1.1 subroutine extract_data_pic_lo (integer *NTMP,* integer *QSZ,* double precision *SCALE,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(3) *CP,* double precision, dimension(3) *SP,* logical *EUSE,* double precision *ETOL,* double precision, dimension(2,3) *DPIC_DLOC,* double precision *DIDH,* double precision *DJDH,* integer, dimension(2) *Z0,* integer *KK,* integer *ESZ,* integer, dimension(-1024:1024,-1024:1024) *DN,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *DNX*)

Procedure.

Parameters

ntmp	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP O QSZ O SCALE O PICNM I SUMFILES/<PICNM>.SUM IPL O V O UX O UY O UZ O HUSE O HT O CP O SP O EUSE O ETOL O FINV O DPIC_DLO!! O DIDH O DJDH O Z0 O KK O ESZ O DN O DNX O

File_I/O

Filename I/O Description

./TESTFILES/<PICNM>.SUM I Lockout file for <PICNM>.SUM ./SUMFILES/<PICNM>.SUM I Name of summary file to extract data from.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY FIND_PICFILE HGT2SLP LOC2PIX_LO V2IMGPL

SPICELIB_functions_called BRT1 VDOT

SPICELIB_subroutines_called VADD VHAT

Called_by_SPC_Programs LITHOSP SUBROUTINE EXTRACT_DATA_LO SUBROUTINE STEREO_LO

History 2013_10_15: Capitalization, compiler warnings fixed and header added. 2014_04_04: Local lower fractional threshold LOCT1 added. Controlled by INIT_LITHOS.TXT. Default is no operation. 2014_04_17: IFF saved.

2.34 COMMON/EXTRACT_GRAD.f File Reference

Functions/Subroutines

- subroutine extract_grad (PMX, NTMP, QSZ, SCALE, NPIX, PICID, IPL, V, UX, UY, UZ, HUSE, HT, CP, SP, EUSE, ETOL, DPIC_DLOC, DIDH, DJDH, NSZ, Z0, KK, ESZ, DNK, DN1)

  Procedure.

2.34.1 Function/Subroutine Documentation

2.34.1.1 **subroutine extract_grad (integer *PMX,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* integer *NPIX,* character∗12, dimension(pmx) *PICID,* double precision, dimension(2,pmx) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(3,pmx) *CP,* double precision, dimension(3,pmx) *SP,* logical *EUSE,* double precision *ETOL,* double precision, dimension(2,3,pmx) *DPIC_DLOC,* double precision, dimension(pmx) *DIDH,* double precision, dimension(pmx) *DJDH,* integer *NSZ,* integer, dimension(2,pmx) *Z0,* integer, dimension(pmx) *KK,* integer *ESZ,* integer, dimension(-128:128,-128:128,pmx) *DNK,* real∗4, dimension(5000,5000) *DN1*)

Procedure.

Parameters

pmx	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX NTMP QSZ SCALE NPIX PICID IPL V UX UY UZ HUSE HT CP SP EUSE ETOL DPIC_DLOC DIDH DJDH NSZ Z0 KK ESZ DNK DN1

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called EXTRACT_DATA_PIC FILLER

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.35 COMMON/FILLER.f File Reference

Functions/Subroutines

- subroutine filler (NTMP, QSZ, SIZE, HUSE, HT)

  Procedure.

2.35.1 Function/Subroutine Documentation

2.35.1.1 subroutine filler (integer∗4 *NTMP,* integer∗4 *QSZ,* integer∗4 *SIZE,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

Procedure.

Parameters

ntmp	Abstract `This subroutine is used to patch small holes in maplets where insufficient data has made it impossble to produce topography. It is almost never used, since the prefered alternative is to fill missing data with slopes from the globel thpography model (shape) during the slope to height integration.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract This subroutine is used to patch small holes in maplets where insufficient data has made it impossble to produce topography. It is almost never used, since the prefered alternative is to fill missing data with slopes from the globel thpography model (shape) during the slope to height integration.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

size	Abstract This subroutine is used to patch small holes in maplets where insufficient data has made it impossble to produce topography. It is almost never used, since the prefered alternative is to fill missing data with slopes from the globel thpography model (shape) during the slope to height integration.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ SIZE HUSE HT

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs EXTRACT_DATA EXTRACT_GRAD

2.36 COMMON/FIND_ALBEDO.f File Reference

Functions/Subroutines

- subroutine find_albedo (PMX, NTMP, NPIX, QSZ, NSZ, DN1, SP, CP, PUSE, LAMBDA, PHI, SK, TUSE, TMPL)

  Procedure.

2.36.1 Function/Subroutine Documentation

2.36.1.1 subroutine find_albedo (integer *PMX,* integer *NTMP,* integer *NPIX,* integer *QSZ,* integer *NSZ,* real∗4, dimension(5000,5000) *DN1,* double precision, dimension(3,pmx) *SP,* double precision, dimension(3,pmx) *CP,* logical, dimension(pmx) *PUSE,* double precision, dimension(pmx) *LAMBDA,* double precision, dimension(pmx) *PHI,* double precision, dimension(pmx) *SK,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL*)

Procedure.

Parameters

pmx	Abstract This subroutine is called by LITHOS and LITHOSP if ALPAD=.TRUE. in INIT_LITHOS.TXT. It is intended to give a better estimate for the albedo at each pixel of the maplet by more appropriate weighting of the data. The input weighting factor SK = SCALE**2/(SCALE**2+PICRES**2) approaches 1 for very high resolution images and 0 for very low resolution images. It is then multiplied by a factor [1-0.75cos(i)] that de-emphasizes high local sun angles and by B^4/(B^4+nu^4) that de-emphasizes brightness less than nu=.05 (5% of full scale). The sum of the weighted squared residuals between observed and predicted brightness for each image contributing to each pixel of the maplet are minimized to obtain a correction to the albedo.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX I Maximum number of images contributing to a maplet NTMP I Maximum maplet pixel range NPIX I Number of images contributing to maplet. QSZ I Maplet pixel range from -qsz to qsz. NSZ I Number of maplets in one row of data grid DN1 I Data grid with extracted and predicted maplet brightness SP I Sun vector for each image in maplet frame CP I Camera vector for each image in maplet frame PUSE I .TRUE. if image used in maplet construction LAMBDA I Current brightness coefficient for each image PHI I Current background brightness for each picture SK I A weighting factor emphasizing higher res images TUSE I .TRUE. means maplet data exists at that pixel TMPL I/O Slope and albedo deviation at each maplet pixel

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called ILLUM

SPC_subroutines_called None

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.37 COMMON/FIND_LAMBDA.f File Reference

Functions/Subroutines

- subroutine find_lambda (PMX, NTMP, NPIX, QSZ, NSZ, DN1, SP, CP, TUSE, TMPL, LAMBDA, PHI, CHI, CHI0)

  Procedure.

2.37.1 Function/Subroutine Documentation

2.37.1.1 subroutine find_lambda (integer *PMX,* integer *NTMP,* integer *NPIX,* integer *QSZ,* integer *NSZ,* real∗4, dimension(5000,5000) *DN1,* double precision, dimension(3,pmx) *SP,* double precision, dimension(3,pmx) *CP,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* double precision, dimension(pmx) *LAMBDA,* double precision, dimension(pmx) *PHI,* double precision, dimension(pmx) *CHI,* double precision *CHI0*)

Procedure.

Parameters

pmx	Abstract `This subroutine is a wrapper that calls FIND_LAMBDA_PI!!to determine the parameter LAMBDA for all images containing a given maplet. This parameter sets the scale for the conversion from brightness distribution to topography and back that is used in the stereophotoclinometry (SPC) process. The imaging data used is read from the array set up by the EXTRACT_DATA subroutine.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX NTMP NPIX QSZ NSZ DN1 SP CP TUSE TMPL LAMBDA PHI CHI CHI0

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called FIND_LAMBDA_PIC

SPICELIB_functions_called None

SPICELIB_subroutines_called None

2.38 COMMON/FIND_LAMBDA_PIC.f File Reference

Functions/Subroutines

- subroutine find_lambda_pic (NTMP, QSZ, DNX, SP, CP, TUSE, TMPL, LAMBDA, PHI, CHI, CHI0)

  Procedure.

2.38.1 Function/Subroutine Documentation

2.38.1.1 subroutine find_lambda_pic (integer *NTMP,* integer *QSZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *DNX,* double precision, dimension(3) *SP,* double precision, dimension(3) *CP,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* double precision *LAMBDA,* double precision *PHI,* double precision *CHI,* double precision *CHI0*)

Procedure.

Parameters

ntmp	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ DNX SP CP TUSE TMPL LAMBDA PHI CHI CHI0

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called ILLUM

SPC_subroutines_called None

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER SUBROUTINE FIND_LAMBDA

2.39 COMMON/FIND_PICFILE.f File Reference

Functions/Subroutines

- subroutine find_picfile (PICNM, PICFILE, EX)

  Procedure.

2.39.1 Function/Subroutine Documentation

2.39.1.1 subroutine find_picfile (character∗12 *PICNM,* character∗72 *PICFILE,* logical *EX*)

Procedure.

Abstract This subroutine will search through directories IMAGEFILES, IMAGEFILES1, IMAGEFILES2, IMAGEFILES3, IMAGEFILES4, IMAGEFILES5, IMAGEFILES6, IMAGEFILES7, IMAGEFILES8, IMAGEFILES9 for the image file <PICNM> and returns the full path name to the file <PICFILE> and the logical variable <EX> indicating if it exists or not.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Name of picture image file to search for PICFILE O Path and file name link to the picture image file if files exists EX O Logical variable indicating if file exists or not

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP BLEMISHES LIMBER LITHOS LITHOSP RESIDUALS SPHEREMAPSA SUBROUTINE EXTRACT_DATA_PIC SUBROUTINE FIND_PICFILE SUBROUTINE LIMB_HEIGHTS SUBROUTINE PICINPT

2.40 COMMON/FIND_SLOPES.f File Reference

Functions/Subroutines

- subroutine find_slopes (PMX, NTMP, NPIX, QSZ, NSZ, DN1, SP, CP, PUSE, APR, LAMBDA, PHI, SK, TUSE, TMPL, RESIDUAL, SS)

  Procedure.

2.40.1 Function/Subroutine Documentation

2.40.1.1 subroutine find_slopes (integer *PMX,* integer *NTMP,* integer *NPIX,* integer *QSZ,* integer *NSZ,* real∗4, dimension(5000,5000) *DN1,* double precision, dimension(3,pmx) *SP,* double precision, dimension(3,pmx) *CP,* logical, dimension(pmx) *PUSE,* double precision, dimension(3) *APR,* double precision, dimension(pmx) *LAMBDA,* double precision, dimension(pmx) *PHI,* double precision, dimension(pmx) *SK,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* double precision *RESIDUAL,* double precision *SS*)

Procedure.

Parameters

pmx	Abstract This subroutine sets up the estmation that determines the slope and relative albedo for each pixel in a maplet by minimzing the brightness residuals at each point. The actual heavy lifting for each pixel is done by the subroutine FIND_TMPL. This subroutine determines the residuals and the weights for the estimation. Note that lower resolution images are weighted less as are those with lower incidence angles and very dark pixels. This weighting is not reflected in the "RMS brightness residuals" that are displayed when LITHOS calls this subroutine, so that value may increase slightly with iteration. There is no reason why these two subroutines could not be combined, but for the moment we are taking an "If it ain't broke ..." attitude.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX NTMP NPIX QSZ NSZ DN1 SP CP PUSE APR LAMBDA PHI SK TUSE TMPL RESIDUAL SS

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called ILLUM

SPC_subroutines_called FIND_TMPL

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.41 COMMON/FIND_TMPL.f File Reference

Functions/Subroutines

- subroutine find_tmpl (PMX, NPIX, RSD, SP, CP, APR, LAMBDA, WGT, T1, T2, T3, USE, PUSE, TFND, RS↩ DSQ, WSM, S2S)

  Procedure.

2.41.1 Function/Subroutine Documentation

2.41.1.1 **subroutine find_tmpl (integer *PMX,* integer *NPIX,* double precision, dimension(pmx) *RSD,* double precision, dimension(3,pmx) *SP,* double precision, dimension(3,pmx) *CP,* double precision, dimension(3) *APR,* double precision, dimension(pmx) *LAMBDA,* double precision, dimension(pmx) *WGT,* real∗4 *T1,* real∗4 *T2,* real∗4 *T3,* logical, dimension(pmx) *USE,* logical, dimension(pmx) *PUSE,* logical *TFND,* double precision *RSDSQ,* double precision *WSM,* double precision *S2S*)**

Procedure.

Parameters

pmx	Abstract This subroutine performs the estimation of slope and relative albedo for each pont in the maplet, using residuals and weights from the FIND_SLOPES subroutine. With T1 and T2 the slopes ant 1+T3 the relative albedo, we are minimizing: W(k)(O - P - dT1P/T1 - dT2P/T2 - dT3P/T3)^2 k where the sum is over images, O is the observed brightness at a pixel and P is the predicted brightness. The weights W(k) and residuals O-P are carried in from the FIND_SLOPES subroutine. The partials with respect to T1 and T2 are determined numerically. There is no reason why these two subroutines could not be combined, but for the moment we are taking an "If it ain't broke ..." attitude.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PMX NPIX RSD SP CP APR LAMBDA WGT T1 T2 T3 USE PUSE TFND RSDSQ WSM S2S

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I Setup file for the SP!!toolkit

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called INVERTN

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS SUBROUTINE FIND_SLOPES

2.42 COMMON/FLIP.f File Reference

Functions/Subroutines

- subroutine flip (N, LFLAG, CH1, CH2)

  Procedure.

2.42.1 Function/Subroutine Documentation

2.42.1.1 subroutine flip (integer∗4 *N,* logical *LFLAG,* character∗(∗) *CH1,* character∗(∗) *CH2*)

Procedure.

Parameters

n	Abstract The subroutine will change the endian of the N byte variable CH1 and return the value in CH2 if the logical variable is set to true, else the CH1 value is returned in CH2. The calling routine analyzes the hardware architecture and set the LFLAG based on the analysis.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

N I Number of bytes LFLAG I Logical flag set to TRUE if flip is to be done CH1 O Input variable to be operated on CH2 O Returned value of the operation

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LITHOSP POLE SHIFT DENSIFY DENSIFYA SPHEREMAPS↩ A SPHEREMAPSB

2.43 COMMON/GET_HEIGHTS.f File Reference

Functions/Subroutines

- subroutine get_heights (NTMP, QSZ, UX, UY, UZ, V0, SCALE, HUSE, HT, AL)

  Procedure.

2.43.1 Function/Subroutine Documentation

2.43.1.1 subroutine get_heights (integer∗4 *NTMP,* integer∗4 *QSZ,* real∗8, dimension(3) *UX,* real∗8, dimension(3) *UY,* real∗8, dimension(3) *UZ,* real∗8, dimension(3) *V0,* real∗8 *SCALE,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *AL*)

Procedure.

Parameters

ntmp	Abstract

```
This subroutine populates a maplet with heights and albedos from a
shape model.  The maplet coordinate system shown below defines a
piece of surface by heights relative to a grid on a plane.

                            Uz
                             ^
                ---> i       |
                  ___________|__________
            \    \           |          \
             \    \          |           \
              j    \         ------------\---------> Uy
                    \         \           \
                     \         \           \
                      \_________\___________\
                                 \
                                  \
                                   Ux

Grid points are labled (i,j) where i and j run from -qsz to qsz.  Each
point has an associated height and albedo.  The SCALE is the distance
between adjacent grid points.  It is measured in kilometers.  The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the maplet center (0,0) the vector to a maplet
point i,j is

      V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)

The SHAPE or GLOBAL TOPOGRAPHY model is in the implicitly connected
quadrilateral (ICQ) format.  Body-fixed surface vectors are labled as
v(I,J,F) where I and J have values from 0 to q, and f from 1 to 6.  The
parameter q is always a power of 2, and is usually taken to be 512,
yielding a 1.57 million vector model.  If one or both of I or J is 0 or q
then the point I,J,F corresponds to a point with another F value as
determined by the subroutines CORNERS and MATCHUP.  The labeling scheme
can be visualized as 6 grids on the faces of a cube, with common points
on the edges and corners.  Four adjacent surface points form a cell of
the model as shown below.


            . I,J (0)
                              . I+1,J (1)
                   x I+w1,J+w2
            .
          I,J+1 (2)
                            .  I+1,J+1 (3)

The vectors to each of these points are projected into the maplet frame
Ux, Uy, Uz.  Each point maps into a maplet point X, Y, H(X,Y) where X,Y
are not necessarily integers and so do not corresponding to maplet grid
points.  If the corner points are labeled k=0,1,2,3 as above, the values
of some quantity B(k) associated with those points can be interpolated
to the point I+w1,J+w2 through bilinear interpolation.  With the
definitions

             b0=(0)          b1=B(1)-B(0)
             b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)

the interpolated value is

             B = b0 + b1w1 + b2w2 + b3w1w2

The subroutine determines values w1,w2 so that the X(k) and yYk)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

             i = x0 + x1w1 + x2w2 + x3w1w2
             j = y0 + y1w1 + y2w2 + y3w1w2

then the height and albedo at this point is taken to be

             h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
             a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```



```
There is the possibility that two parts of the shape map into the same
i,j position on the maplet.  For example, we might be looking clear
through the shape model to the other side.  In order to avoid this
mistake, we always choose the values corresponding the the largest height
at the point i,j.
```

Disclaimer

Required_Reading

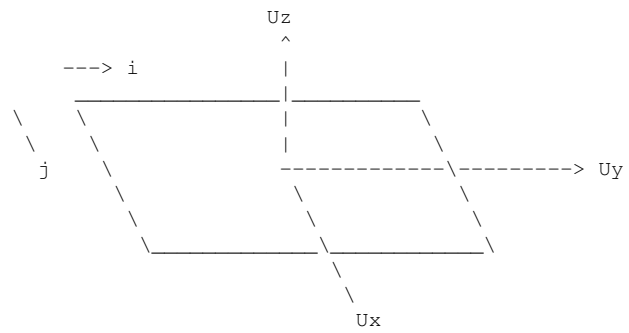
R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract

```
This subroutine populates a maplet with heights and albedos from a
shape model.  The maplet coordinate system shown below defines a
piece of surface by heights relative to a grid on a plane.
```




```
Grid points are labled (i,j) where i and j run from -qsz to qsz.  Each
point has an associated height and albedo.  The SCALE is the distance
between adjacent grid points.  It is measured in kilometers.  The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the maplet center (0,0) the vector to a maplet
point i,j is

      V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)

The SHAPE or GLOBAL TOPOGRAPHY model is in the implicitly connected
quadrilateral (ICQ) format.  Body-fixed surface vectors are labled as
v(I,J,F) where I and J have values from 0 to q, and f from 1 to 6.  The
parameter q is always a power of 2, and is usually taken to be 512,
yielding a 1.57 million vector model.  If one or both of I or J is 0 or q
then the point I,J,F corresponds to a point with another F value as
determined by the subroutines CORNERS and MATCHUP.  The labeling scheme
can be visualized as 6 grids on the faces of a cube, with common points
on the edges and corners.  Four adjacent surface points form a cell of
the model as shown below.


            . I,J (0)
                              . I+1,J (1)
                   x I+w1,J+w2
            .
          I,J+1 (2)
                            .  I+1,J+1 (3)

The vectors to each of these points are projected into the maplet frame
Ux, Uy, Uz.  Each point maps into a maplet point X, Y, H(X,Y) where X,Y
are not necessarily integers and so do not corresponding to maplet grid
points.  If the corner points are labeled k=0,1,2,3 as above, the values
of some quantity B(k) associated with those points can be interpolated
to the point I+w1,J+w2 through bilinear interpolation.  With the
definitions

             b0=(0)          b1=B(1)-B(0)
             b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)

the interpolated value is

             B = b0 + b1w1 + b2w2 + b3w1w2

The subroutine determines values w1,w2 so that the X(k) and yYk)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

             i = x0 + x1w1 + x2w2 + x3w1w2
             j = y0 + y1w1 + y2w2 + y3w1w2

then the height and albedo at this point is taken to be

             h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
             a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```



```
There is the possibility that two parts of the shape map into the same
i,j position on the maplet.  For example, we might be looking clear
through the shape model to the other side.  In order to avoid this
mistake, we always choose the values corresponding the the largest height
at the point i,j.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ UX UY UZ V0 SCALE HUSE HT AL

File_I/O

Filename I/O Description

SHAPE.TXT or SHAPEA.TXT I Shape model that contains heights. Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_MODEL

SPICELIB_functions_called VSUB

SPICELIB_subroutines_called VDOT

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER BIGMAP BIGMAPL SUBROUTINE CREATE↩ _LMFILE SUBROUTINE LIMG_HEIGHTS SUBROUTINE VISIBLE

2.44 COMMON/GET_HEIGHTS_LO.f File Reference

Functions/Subroutines

- subroutine get_heights_lo (NTMP, QSZ, UX, UY, UZ, V0, SCALE, HUSE, HT)

  Procedure.

2.44.1 Function/Subroutine Documentation

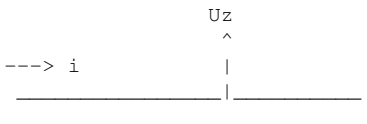
2.44.1.1 subroutine get_heights_lo (integer∗4 *NTMP,* integer∗4 *QSZ,* real∗8, dimension(3) *UX,* real∗8, dimension(3) *UY,* real∗8, dimension(3) *UZ,* real∗8, dimension(3) *V0,* real∗8 *SCALE,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

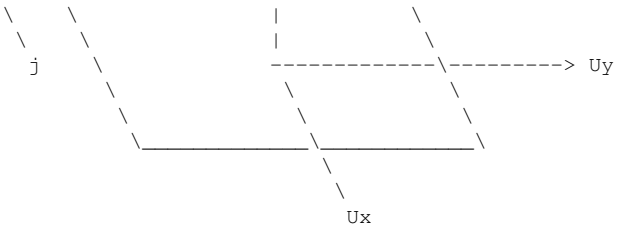
Procedure.

Parameters

ntmp	Abstract This subroutine is identical to GET_HEIGHTS except for the call to GET_MODEL↩ _LO(Q,VEC) instead of GET_MODEL(INFILE,Q,VEC,A0)

The subroutine populates a maplet with heights and albedos from a shape model. The maplet coordinate system shown below defines a piece of surface by heights relative to a grid on a plane.





Grid points are labled (i,j) where i and j run from -qsz to qsz. Each point has an associated height and albedo. The SCALE is the distance between adjacent grid points. It is measured in kilometers. The height HT relative to the plane is also measured in units of the scale, so if V0 is the body-fixed vector to the maplet center (0,0) the vector to a maplet point i,j is

```
V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)
```

The SHAPE or GLOBAL TOPOGRAPHY model is in the implicitly connected quadrilateral (ICQ) format. Body-fixed surface vectors are labled as v(I,J,F) where I and J have values from 0 to q, and f from 1 to 6. The parameter q is always a power of 2, and is usually taken to be 512, yielding a 1.57 million vector model. If one or both of I or J is 0 or q then the point I,J,F corresponds to a point with another F value as determined by the subroutines CORNERS and MATCHUP. The labeling scheme can be visualized as 6 grids on the faces of a cube, with common points on the edges and corners. Four adjacent surface points form a cell of the model as shown below.

```
            . I,J (0)
                              . I+1,J (1)
                  x I+w1,J+w2
            .
          I,J+1 (2)
                           .  I+1,J+1 (3)
```

The vectors to each of these points are projected into the maplet frame Ux, Uy, Uz. Each point maps into a maplet point X, Y, H(X,Y) where X,Y are not necessarily integers and so do not corresponding to maplet grid points. If the corner points are labeled k=0,1,2,3 as above, the values of some quantity B(k) associated with those points can be interpolated to the point I+w1,J+w2 through bilinear interpolation. With the definitions

```
b0=(0)          b1=B(1)-B(0)
b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)
```

the interpolated value is

```
B = b0 + b1w1 + b2w2 + b3w1w2
```

The subroutine determines values w1,w2 so that the X(k) and yYk) interpolate to integer values lying within the maplet (between -qsz and qsz):

```
i = x0 + x1w1 + x2w2 + x3w1w2
j = y0 + y1w1 + y2w2 + y3w1w2
```

then the height and albedo at this point is taken to be

```
h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```

There is the possibility that two parts of the shape map into the same i,j position on the maplet. For example, we might be looking clear through the shape model to the other side. In order to avoid this mistake, we always choose the values corresponding the the largest height at the point i,j.

The interpolation is allowed to spill outside the cell slightly with a small parameter EPS. It is unclear whether this is really necessary.

Disclaimer

Required_Reading

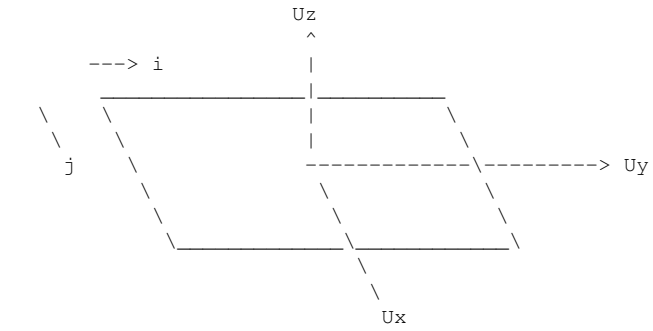
R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract This subroutine is identical to GET_HEIGHTS except for the call to GET_MODEL↩ _LO(Q,VEC) instead of GET_MODEL(INFILE,Q,VEC,A0)

The subroutine populates a maplet with heights and albedos from a shape model. The maplet coordinate system shown below defines a piece of surface by heights relative to a grid on a plane.




```
Grid points are labled (i,j) where i and j run from -qsz to qsz.  Each
point has an associated height and albedo.  The SCALE is the distance
between adjacent grid points.  It is measured in kilometers.  The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the maplet center (0,0) the vector to a maplet
point i,j is

  V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)

The SHAPE or GLOBAL TOPOGRAPHY model is in the implicitly connected
quadrilateral (ICQ) format.  Body-fixed surface vectors are labled as
v(I,J,F) where I and J have values from 0 to q, and f from 1 to 6.  The
parameter q is always a power of 2, and is usually taken to be 512,
yielding a 1.57 million vector model.  If one or both of I or J is 0 or q
then the point I,J,F corresponds to a point with another F value as
determined by the subroutines CORNERS and MATCHUP.  The labeling scheme
can be visualized as 6 grids on the faces of a cube, with common points
on the edges and corners.  Four adjacent surface points form a cell of
the model as shown below.

            . I,J (0)
                              . I+1,J (1)
                  x I+w1,J+w2
            .
          I,J+1 (2)
                            .  I+1,J+1 (3)

The vectors to each of these points are projected into the maplet frame
Ux, Uy, Uz.  Each point maps into a maplet point X, Y, H(X,Y) where X,Y
are not necessarily integers and so do not corresponding to maplet grid
points.  If the corner points are labeled k=0,1,2,3 as above, the values
of some quantity B(k) associated with those points can be interpolated
to the point I+w1,J+w2 through bilinear interpolation.  With the
definitions

             b0=(0)          b1=B(1)-B(0)
             b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)

the interpolated value is

             B = b0 + b1w1 + b2w2 + b3w1w2
```

```
The subroutine determines values w1,w2 so that the X(k) and yYk)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

             i = x0 + x1w1 + x2w2 + x3w1w2
             j = y0 + y1w1 + y2w2 + y3w1w2

then the height and albedo at this point is taken to be

             h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
             a(i,j) = a0 + a1w1 + a2w2 + a3w1w2

There is the possibility that two parts of the shape map into the same
i,j position on the maplet.  For example, we might be looking clear
through the shape model to the other side.  In order to avoid this
mistake, we always choose the values corresponding the the largest height
at the point i,j.

The interpolation is allowed to spill outside the cell slightly with a
small parameter EPS.  It is unclear whether this is really necessary.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ UX UY UZ V0 SCALE HUSE HT AL

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_MODEL_LO

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs LITHOSP SUBROUTINE LIMB_HEIGHTS_LO

History 2013_10_14: Capitalization, compiler warnings fixed and header added.

2.45 COMMON/GET_MAP.f File Reference

Functions/Subroutines

- subroutine get_map (NAME, NTMP, QSZ, V, UX, UY, UZ, S0, S1, AL, HT, HUSE, USE, FCT)

Procedure.

2.45.1 Function/Subroutine Documentation

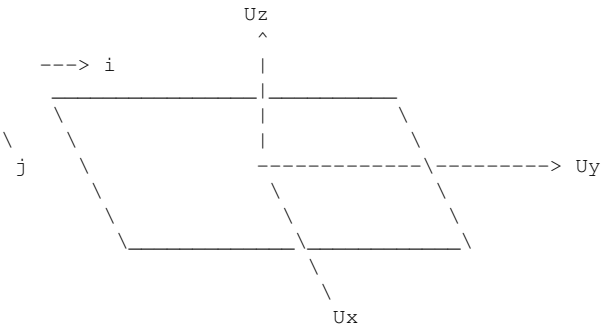
2.45.1.1 subroutine get_map (character∗6 *NAME,* integer *NTMP,* integer *QSZ,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *S0,* double precision *S1,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *AL,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* logical *USE,* double precision *FCT*)

Procedure.

Parameters

ntmp	Abstract

This subroutine populates a target map with heights and albedos from
another sampled map. The target map coordinate system shown below defines !! a piece of s



Grid points are labled (i,j) where i and j run from -qsz to qsz. Each
point has an associated height and albedo. The SCALE is the distance
between adjacent grid points. It is measured in kilometers. The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the map center (0,0) the vector to a map
point i,j is

```
V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)
```

The sampled map has the same format but with, in general, a different
scale, qsz and orientation. Four adjacent surface points form a cell of
the sampled as shown below.

```
. I,J (0)
                    . I+1,J (1)
     x I+w1,J+w2
.
I,J+1 (2)
                  .  I+1,J+1 (3)
```

The vectors to each of these points are projected into the target map
frame Ux, Uy, Uz. Each point maps into a map point X, Y, H(X,Y) where
X,Y are not necessarily integers and so do not correspond to target map
grid points. If the corner points are labeled k=0,1,2,3 as above, the
values of some quantity B(k) associated with those points can be
interpolated to the point I+w1,J+w2 through bilinear interpolation.
With the definitions

```
b0=(0)          b1=B(1)-B(0)
b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)
```

the interpolated value is

```
B = b0 + b1w1 + b2w2 + b3w1w2
```

The subroutine determines values w1,w2 so that the X(k) and Y(k)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

```
i = x0 + x1w1 + x2w2 + x3w1w2
j = y0 + y1w1 + y2w2 + y3w1w2
```

then the height and albedo at this point is taken to be

```
h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```

There is the possibility that mulitple cells of the sampled map projrct
into the same i,j position on the target map, so the average of all such
values is used.

The interpolation is allowed to spill outside the cell slightly with a
small parameter EPS. It is unclear whether this is really necessary.

Disclaimer

Required_Reading

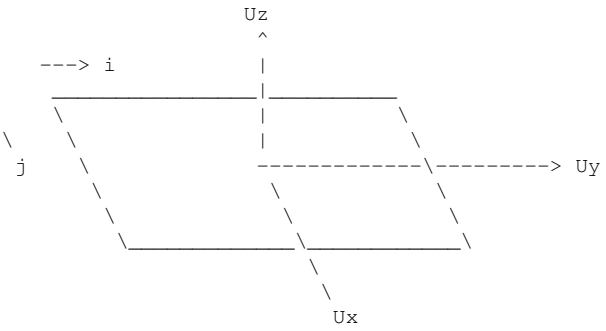
R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract

This subroutine populates a target map with heights and albedos from
another sampled map. The target map coordinate system shown below defines !! a piece of s



Grid points are labled (i,j) where i and j run from -qsz to qsz. Each
point has an associated height and albedo. The SCALE is the distance
between adjacent grid points. It is measured in kilometers. The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the map center (0,0) the vector to a map
point i,j is

```
V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)
```

The sampled map has the same format but with, in general, a different
scale, qsz and orientation. Four adjacent surface points form a cell of
the sampled as shown below.

```
. I,J (0)
                    . I+1,J (1)
     x I+w1,J+w2
.
I,J+1 (2)
                  .  I+1,J+1 (3)
```

The vectors to each of these points are projected into the target map
frame Ux, Uy, Uz. Each point maps into a map point X, Y, H(X,Y) where
X,Y are not necessarily integers and so do not correspond to target map
grid points. If the corner points are labeled k=0,1,2,3 as above, the
values of some quantity B(k) associated with those points can be
interpolated to the point I+w1,J+w2 through bilinear interpolation.
With the definitions

```
b0=(0)          b1=B(1)-B(0)
b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)
```

the interpolated value is

```
B = b0 + b1w1 + b2w2 + b3w1w2
```

The subroutine determines values w1,w2 so that the X(k) and Y(k)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

```
i = x0 + x1w1 + x2w2 + x3w1w2
j = y0 + y1w1 + y2w2 + y3w1w2
```

then the height and albedo at this point is taken to be

```
h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```

There is the possibility that mulitple cells of the sampled map projrct
into the same i,j position on the target map, so the average of all such
values is used.

The interpolation is allowed to spill outside the cell slightly with a
small parameter EPS. It is unclear whether this is really necessary.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NAME I Base name of the maplet file. NTMP QSZ V UX UY UZ S0 S1 AL HT HUSE USE FCT

File_I/O

Filename I/O Description

MAPFILES/<NAME>.MAP I Maplet file to read data from.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called READ_MAP

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs LITHOS REGISTER SUBROUTINE ATTACH SUBROUTINE LIMB_HEIGHTS

History 2014_01_24: Parameter BTMP set to 2501 to be consistent with NTMP

2.46 COMMON/GET_MAP_LO.f File Reference

Functions/Subroutines

- subroutine get_map_lo (NAME, NTMP, QSZ, V, UX, UY, UZ, S0, S1, AL, HT, HUSE, USE, FCT)

  Procedure.

2.46.1 Function/Subroutine Documentation

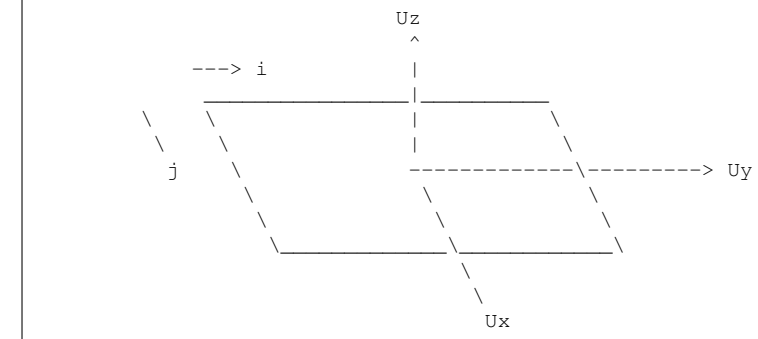
2.46.1.1 subroutine get_map_lo (character∗6 *NAME,* integer *NTMP,* integer *QSZ,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *S0,* double precision *S1,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *AL,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* logical *USE,* double precision *FCT*)

Procedure.

Parameters

ntmp	Abstract

```
This subroutine populates a target map with heights and albedos from
another sampled map. The target map coordinate system shown below defines !!    a piece of s
```




```
Grid points are labled (i,j) where i and j run from -qsz to qsz.  Each
point has an associated height and albedo.  The SCALE is the distance
between adjacent grid points.  It is measured in kilometers.  The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the map center (0,0) the vector to a map
point i,j is

      V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)

The sampled map has the same format but with, in general, a different
scale, qsz and orientation.  Four adjacent surface points form a cell of
the sampled as shown below.


            . I,J (0)
                                . I+1,J (1)
                 x I+w1,J+w2
            .
          I,J+1 (2)
                              .  I+1,J+1 (3)

The vectors to each of these points are projected into the target map
frame Ux, Uy, Uz.  Each point maps into a map point X, Y, H(X,Y) where
X,Y are not necessarily integers and so do not correspond to target map
grid points.  If the corner points are labeled k=0,1,2,3 as above, the
values of some quantity B(k) associated with those points can be
interpolated to the point I+w1,J+w2 through bilinear interpolation.
With the definitions

             b0=(0)          b1=B(1)-B(0)
             b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)

the interpolated value is

             B = b0 + b1w1 + b2w2 + b3w1w2

The subroutine determines values w1,w2 so that the X(k) and Y(k)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

             i = x0 + x1w1 + x2w2 + x3w1w2
             j = y0 + y1w1 + y2w2 + y3w1w2

then the height and albedo at this point is taken to be

             h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
             a(i,j) = a0 + a1w1 + a2w2 + a3w1w2

There is the possibility that mulitple cells of the sampled map projrct
into the same i,j position on the target map, so the average of all such
values is used.

The interpolation is allowed to spill outside the cell slightly with a
small parameter EPS.  It is unclear whether this is really necessary.
This subroutine uses a LOCKOUT file to prevent reading/writing to any
open MAPFILE.
```

Disclaimer

Required_Reading

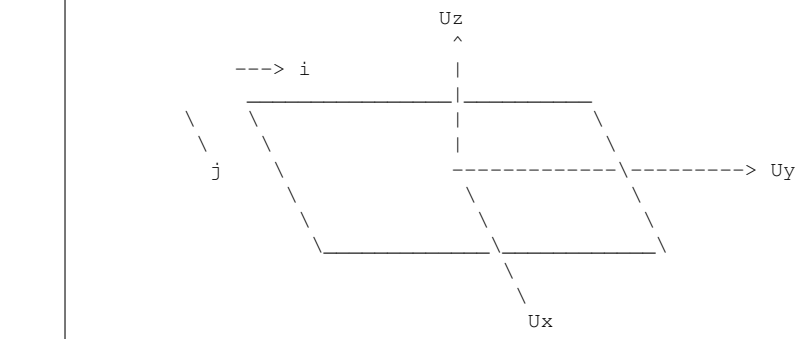
R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract

This subroutine populates a target map with heights and albedos from
another sampled map. The target map coordinate system shown below defines !! a piece of s



Grid points are labled (i,j) where i and j run from -qsz to qsz. Each
point has an associated height and albedo. The SCALE is the distance
between adjacent grid points. It is measured in kilometers. The height
HT relative to the plane is also measured in units of the scale, so if V0
is the body-fixed vector to the map center (0,0) the vector to a map
point i,j is

```
V(i,j) = V0 + SCALE X (iUy + jUx + HT(i,j)Uz)
```

The sampled map has the same format but with, in general, a different
scale, qsz and orientation. Four adjacent surface points form a cell of
the sampled as shown below.

```
. I,J (0)
                    . I+1,J (1)
     x I+w1,J+w2
.
I,J+1 (2)
                  .  I+1,J+1 (3)
```

The vectors to each of these points are projected into the target map
frame Ux, Uy, Uz. Each point maps into a map point X, Y, H(X,Y) where
X,Y are not necessarily integers and so do not correspond to target map
grid points. If the corner points are labeled k=0,1,2,3 as above, the
values of some quantity B(k) associated with those points can be
interpolated to the point I+w1,J+w2 through bilinear interpolation.
With the definitions

```
b0=(0)          b1=B(1)-B(0)
b2=B(2)-B(0)    b3=B(0)-B(1)-B(2)+B(3)
```

the interpolated value is

```
B = b0 + b1w1 + b2w2 + b3w1w2
```

The subroutine determines values w1,w2 so that the X(k) and Y(k)
interpolate to integer values lying within the maplet (between -qsz
and qsz):

```
i = x0 + x1w1 + x2w2 + x3w1w2
j = y0 + y1w1 + y2w2 + y3w1w2
```

then the height and albedo at this point is taken to be

```
h(i,j) = h0 + h1w1 + h2w2 + h3w1w2
a(i,j) = a0 + a1w1 + a2w2 + a3w1w2
```

There is the possibility that mulitple cells of the sampled map projrct
into the same i,j position on the target map, so the average of all such
values is used.

The interpolation is allowed to spill outside the cell slightly with a
small parameter EPS. It is unclear whether this is really necessary.
~~This subroutine uses a LOCKOUT file to prevent reading/writing to any~~
open MAPFILE.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NAME I Base name of the maplet file. NTMP QSZ V UX UY UZ S0 S1 AL HT HUSE USE FCT

File_I/O

Filename I/O Description

./TESTFILES/<NAME>.MAP O Lockout file for <NAME>.MAP ./MAPFILES/<NAME>.MAP I Maplet file to read data from.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs LITHOSP

History 2013_10_14: Capitalization, compiler warnings fixed and header added. 2014_01_27: Parameter BTMP set to 2501 to be consistent with NTMP

2.47 COMMON/GET_MODEL.f File Reference

Functions/Subroutines

- subroutine get_model (INFILE, Q, VEC, A0)

  Procedure.

2.47.1 Function/Subroutine Documentation

2.47.1.1 subroutine get_model (character∗72 *INFILE,* integer∗4 *Q,* real∗8, dimension(3,0:512,0:512,6) *VEC,* real∗8, dimension(0:512,0:512,6) *A0*)

Procedure.

Parameters

vec	Abstract This routine will open a shape model file in ICQ format and read in in the vector values and albedo data (if available).

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

a0	Abstract This routine will open a shape model file in ICQ format and read in in the vector values and albedo data (if available).

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

INFILE I Shape model (ICQ) filename Q I Number of shape model vectors VE!! I Shape model vectors array A0 I Albedo

File_I/O

Filename I/O Description

INFILE (SHAPE MODEL) I ICQ formated shape model to be read in

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LIMBER LITHOSP DENSIFY DENSIFYA SHAPE_INFO SUBRO↩ UTINE GET_HEIGHTS SUBROUTINE GET_MODEL SUBROUTINE MAXLEN SUBROUTINE MM2VN SUBRO↩ UTINE U2VN

2.48 COMMON/GET_MODEL_LO.f File Reference

Functions/Subroutines

- subroutine get_model_lo (Q, VEC)

  Procedure.

2.48.1 Function/Subroutine Documentation

2.48.1.1 subroutine get_model_lo (integer∗4 *Q,* real∗8, dimension(3,0:512,0:512,6) *VEC*)

Procedure.

Parameters

vec	Abstract This routine will open a shape model file in ICQ format and read in in the vector values and albedo data (if available).

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

Q I Number of shape model vectors VE!! O Shape model vectors array

File_I/O

Filename I/O Description

./SHAPEFILES/TSHP<USR>.TXT

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE GET_HEIGHTS_LO

History 2013_10_14: Capitalization, compiler warnings fixed and header added.

2.49 COMMON/GET_REF.f File Reference

Functions/Subroutines

- subroutine get_ref (INFILE, QSZ, UX, UY, UZ, V0, SCALE, HUSE, HT)

  Procedure.

2.49.1 Function/Subroutine Documentation

2.49.1.1 subroutine get_ref (character∗72 *INFILE,* integer∗4 *QSZ,* real∗8, dimension(3) *UX,* real∗8, dimension(3) *UY,* real∗8, dimension(3) *UZ,* real∗8, dimension(3) *V0,* real∗8 *SCALE,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

Procedure.

Abstract

This subroutine opens the reference shape modelin ICQ format and reads in the vector values. This is a rarely used subroutine that encourages maplets to not move very far from a reference shape. It is invoked by thw two lines:

SHAPERF= 'SHAPEFILES/SHAPER.TXT' SIGMARF= 0.05 (sigma in km)

in INIT_LITHOS.TXT. These are usually commented out. When it is used, it creates a predicted maplet near the actual one and the subroutine IPL2VLM is encouraged to keep the actual maplet close.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

INFILE I Name of the shape text file. QSZ UX UY UZ V0 SCALE HUSE HT

File_I/O

Filename I/O Description

<INFILE> I Shape text file from as specified in INIT_LITHOS.TXT

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs GEOMETRY LITHOSP SUBROUTINE IPL2VLM

2.50 COMMON/GET_UL.f File Reference

Functions/Subroutines

- subroutine get_ul (CHR, TP, TL)
  Procedure.

2.50.1 Function/Subroutine Documentation

2.50.1.1 subroutine get_ul (character∗1 *CHR,* integer∗4 *TP,* integer∗4 *TL*)

Procedure.

Abstract Using the input character value (CHR), this routine generates the indices or pointers into the FONT.IN!!data file and returns these values (TP, TL).

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

CHR I character value to used for generating the indices into the FONT.IN!!data file TP O Index integer value of the column value TL O Index integer value of the row value

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP SUBROUTINE ZOOM

2.51 COMMON/GETdIU.f File Reference

Functions/Subroutines

- subroutine getdiu (Z1, Z2, Z3, DIU)

  Procedure.

2.51.1 Function/Subroutine Documentation

2.51.1.1 subroutine getdiu (real∗8 *Z1,* real∗8 *Z2,* real∗8 *Z3,* real∗8, dimension(3,3,3) *DIU*)

Procedure.

Parameters

z1	Abstract The GETIU routine calculates the 3-1-3 rotation matrix IU (i,j) from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA⋆RPD(), Z2=pi/2-DEC⋆RPD(), Z3=(W0+W1⋆T)⋆RPD(). This subroutine returns the partial derivatives of IU(i,j) with respect to Zk as IU/Zk = DIU(i,j,k)

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z2	Abstract The GETIU routine calculates the 3-1-3 rotation matrix IU (i,j) from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA*RPD(), Z2=pi/2-DEC*RPD(), Z3=(W0+W1*T)*RPD(). This subroutine returns the partial derivatives of IU(i,j) with respect to Zk as IU/Zk = DIU(i,j,k)

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z3	Abstract The GETIU routine calculates the 3-1-3 rotation matrix IU (i,j) from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA*RPD(), Z2=pi/2-DEC*RPD(), Z3=(W0+W1*T)*RPD(). This subroutine returns the partial derivatives of IU(i,j) with respect to Zk as IU/Zk = DIU(i,j,k)

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

diu	Abstract The GETIU routine calculates the 3-1-3 rotation matrix IU (i,j) from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA*RPD(), Z2=pi/2-DEC*RPD(), Z3=(W0+W1*T)*RPD(). This subroutine returns the partial derivatives of IU(i,j) with respect to Zk as IU/Zk = DIU(i,j,k)

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

Z1 I Z2 I Z3 I DIU O

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs POLE

2.52 COMMON/GETIU.f File Reference

Functions/Subroutines

- subroutine getiu (Z1, Z2, Z3, IU)

  Procedure.

2.52.1 Function/Subroutine Documentation

2.52.1.1 subroutine getiu (real∗8 *Z1,* real∗8 *Z2,* real∗8 *Z3,* real∗8, dimension(3,3) *IU*)

Procedure.

Parameters

z1	Abstract This routine calculates the 3-1-3 rotation matrix IU from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA∗RPD(), Z2=pi/2-DEC∗RPD(), Z3=(W0+W1∗T)∗RPD().

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z2	Abstract This routine calculates the 3-1-3 rotation matrix IU from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA∗RPD(), Z2=pi/2-DEC∗RPD(), Z3=(W0+W1∗T)∗RPD().

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z3	Abstract This routine calculates the 3-1-3 rotation matrix IU from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA∗RPD(), Z2=pi/2-DEC∗RPD(), Z3=(W0+W1∗T)∗RPD().

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

iu	Abstract This routine calculates the 3-1-3 rotation matrix IU from inertial (J2000) space to body-fixed space from the input Euler angles Z1=pi/2+RA∗RPD(), Z2=pi/2-DEC∗RPD(), Z3=(W0+W1∗T)∗RPD().

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

Z1 I 1st Euler angle, rotation about z Z2 I 2nd Euler angle, rotation about x Z3 I 3rd Euler angle, rotation about z IU O Inertial to body-fixed matrix.

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs POLE SHAPE_INFO IPL2SCOBJPTG

2.53 COMMON/GETMM.f File Reference

Functions/Subroutines

- subroutine getmm (PICNM, MMFL, Z1, Z2, Z3, Z)

  Procedure.

2.53.1 Function/Subroutine Documentation

2.53.1.1 subroutine getmm (character∗12 *PICNM,* double precision *MMFL,* double precision *Z1,* double precision *Z2,* double precision *Z3,* double precision, dimension(2) *Z*)

Procedure.

Parameters

mmfl	Abstract

This subroutine determines the focal plane position corresponding to a camera relative direction (z1,z2,z3). In its simplest form, this is just the gnomoni!!projection Z(1)=MMFL∗z1/z3, Z(2)=MMFL∗z2/z3 where MMFL is the focal length in mm. It also allows for a number of distortions whose parameters are input from the INIT_LITHOS.TXT file followint the keyword DISTORT. The record has the form:

DISTORT= 'image template' 'distortion name' number of parameters

where the 12 character image template describes the form of the image affected by the distortion. For example a wide angle filter B image might be described by W##########B (the # are ignored), so an image named W0193746985B would have the distortion applied. The line of INIT_LITHOS.TXT following the DISTORT record contains the parameters of the distortion. It also has, as the first number of the line, a fractional change in the focal length. This is to handle the (perhaps naive) expectation that all color filters have the same distortion model and differ in effective focal length due to differences in the thicknesses of the filter substrates.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I MMFL I Z1 O Z2 O Z3 O Z O

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I Used to find distortion model

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS REGISTER RESIDUALS POLE SHIFT SUB↩ ROUTINE IPL2RDT SUBROUTINE IPL2SCOBJPTG SUBROUTINE IMP2VLM SUBROUTINE PXMM SUBROU↩ TINE V2IMGPL

2.54 COMMON/HGT2SLP.f File Reference

Functions/Subroutines

- subroutine hgt2slp (NTMP, QSZ, HUSE, HT, TUSE, TMPL)

  Procedure.

2.54.1 Function/Subroutine Documentation

2.54.1.1 subroutine hgt2slp (integer∗4 *NTMP,* integer∗4 *QSZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL*)

Procedure.

Parameters

ntmp	Abstract The first two components of the tmpl array, tmpl(1,i,j) and tmpl(2,i,j) tmpl(2,i,j) represent the negative of the maplet slopes at i,j: tmpl(i,j,1) = -dht(i,j)/dx ~ (ht(i,j-1)-ht(i,j+1)/2 tmpl(i,j,2) = -dht(i,j)/dy ~ (ht(i-1,j)-ht(i+1,j)/2 Note that the discrete map coordinates are arrayed in i,j like _____ i (y) | | | | | | j (x) whereas the continuous coordinates x,y are arranged to make their normal cross product point out of the paper in the direction of positive height. If one of the heights adjacent to ht(i,j) is missing, then the subroutine steps through a heirarchy of approximations, each involving three points, fits a quadrati!!to the points and extracts the slope at (i,j). If none of those work, due to missing data, the subroutine returns the slope from a linear fit between (i,j) and a neighboring point. Finally, if that doesn't work, it sets tmpl(i,j,k)=0 and returns a flag tuse(i,j)=.false.

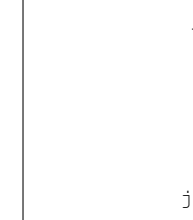
Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract The first two components of the tmpl array, tmpl(1,i,j) and tmpl(2,i,j) tmpl(2,i,j) represent the negative of the maplet slopes at i,j: tmpl(i,j,1) = -dht(i,j)/dx ~ (ht(i,j-1)-ht(i,j+1)/2 tmpl(i,j,2) = -dht(i,j)/dy ~ (ht(i-1,j)-ht(i+1,j)/2 Note that the discrete map coordinates are arrayed in i,j like  whereas the continuous coordinates x,y are arranged to make their normal cross product point out of the paper in the direction of positive height. If one of the heights adjacent to ht(i,j) is missing, then the subroutine steps through a heirarchy of approximations, each involving three points, fits a quadrati!!to the points and extracts the slope at (i,j). If none of those work, due to missing data, the subroutine returns the slope from a linear fit between (i,j) and a neighboring point. Finally, if that doesn't work, it sets tmpl(i,j,k)=0 and returns a flag tuse(i,j)=.false.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ HUSE HT TUSE TMPL

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER SUBROUTINE COMPARE SUBROUTINE E↩ XTRACT_DATA_PIC SUBROUTINE SHOW_SLOPES History 2014_02_02 Additional missing data interpolations added.

2.55 COMMON/ILLUM.f File Reference

Functions/Subroutines

- real ∗8 function illum (NDOTS, NDOTB, ALPHA)

  Procedure.

2.55.1 Function/Subroutine Documentation

2.55.1.1 real∗8 function illum (real∗8 *NDOTS,* real∗8 *NDOTB,* real∗8 *ALPHA*)

Procedure.

Returns

Abstract

```
This function returns the brightness of a pixel of unit albedo as
a function of incident cosine, emission cosine ans phase.  Several
phoometri!!functions are included.  Which one is used is specified in
the INIT_LITHOS.TXT file, with the record for the preferred function
being:  REFLECT='MCEWEN'.  A VESTA illumination function is kept only
by way of illustration.  It was obtained by photometry experts and, as
```

```
such, has nothing to do with real illuminaation and is pretty much
worthless.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

ndotb	Abstract `This function returns the brightness of a pixel of unit albedo as a function of incident cosine, emission cosine ans phase. Several phoometri!!functions are included. Which one is used is specified in the INIT_LITHOS.TXT file, with the record for the preferred function being: REFLECT='MCEWEN'. A VESTA illumination function is kept only by way of illustration. It was obtained by photometry experts and, as such, has nothing to do with real illuminaation and is pretty much worthless.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

ndots	Abstract `This function returns the brightness of a pixel of unit albedo as a function of incident cosine, emission cosine ans phase. Several phoometri!!functions are included. Which one is used is specified in the INIT_LITHOS.TXT file, with the record for the preferred function being: REFLECT='MCEWEN'. A VESTA illumination function is kept only by way of illustration. It was obtained by photometry experts and, as such, has nothing to do with real illuminaation and is pretty much worthless.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

alpha	Abstract `This function returns the brightness of a pixel of unit albedo as a function of incident cosine, emission cosine ans phase. Several phoometri!!functions are included. Which one is used is specified in the INIT_LITHOS.TXT file, with the record for the preferred function being: REFLECT='MCEWEN'. A VESTA illumination function is kept only by way of illustration. It was obtained by photometry experts and, as such, has nothing to do with real illuminaation and is pretty much worthless.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NDOTS I NDOTB I ALPHA I ILLUM O

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I Read REFLECTANCE from file

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER SUBROUTINE FIND_ALBEDO SUBROUTINE FIND_LAMBDA_PIC SUBROUTINE FIND_TMPL SUBROUTINE PREDICT_DATA_PIC

2.56 COMMON/IMGPL2VN.f File Reference

Functions/Subroutines

- subroutine imgpl2vn (PICNM, IMGPL, USE, V, N)

  Procedure.

2.56.1 Function/Subroutine Documentation

2.56.1.1 subroutine imgpl2vn (character∗12 *PICNM,* double precision, dimension(2) *IMGPL,* logical *USE,* double precision, dimension(3) *V,* double precision, dimension(3) *N*)

Procedure.

Abstract

This subroutine determines the expected body-fixed vector V to a surface point on the body appearing at a given image-space (pixel,line) location in an image. After extracting appropriate camera parameters from the PICNM.S↩ UM file, it uses PXMM to transform to the gnomoni!!projection on the focal plane and MM2VN to determine V and the surface normal N.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Base name of the image summary file. IMGPL USE V N

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I Image summary file

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called MM2VN PXMM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER BIGMAP BIGMAPL SUBROUTINE CREATE↩ _LMFILE SUBROUTINE POINT2MAPS

2.57 COMMON/INSERTINLMK_LIM.f File Reference

Functions/Subroutines

- subroutine insertinlmk_lim (LMKNM, PICNM, IPL, SIG)

  Procedure.

2.57.1 Function/Subroutine Documentation

2.57.1.1 subroutine insertinlmk_lim (character∗6 *LMKNM,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* double precision *SIG*)

Procedure.

Parameters

ipl	Abstract This routine will insert or update information for image <PICNM> into the LIMB FITS section of the landmark file <LMKNM>.LMK

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Base name of landmark file to insert limb image PICNM I Image name to insert in landmark file IPL I SIG I

File_I/O

Filename I/O Description

<PICNM>.SUM O Image summary to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LIHOSP SUBROUTINE LIMB_HEIGHTS

2.58 COMMON/INSERTINLMK_LIM_LO.f File Reference

Functions/Subroutines

- subroutine insertinlmk_lim_lo (LMKNM, PICNM, IPL, SIG)

  Procedure.

2.58.1 Function/Subroutine Documentation

2.58.1.1 subroutine insertinlmk_lim_lo (character∗6 *LMKNM,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* double precision *SIG*)

Procedure.

Parameters

ipl	Abstract This routine will insert or update information for image <PICNM> into the LIMB FITS section of the landmark file <LMKNM>.LMK

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Base name of landmark file to insert limb image PICNM I Image name to insert in landmark file IPL I SIG I

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockoutfile for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK O Image summary to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE LIMB_HEIGHTS_LO

History 2013_10_15: Capitalization, compiler warnings fixed and header added.

2.59 COMMON/INSERTINLMK_LMK.f File Reference

Functions/Subroutines

- subroutine insertinlmk_lmk (LMKNM1, LMKNM2, Z1, Z2, Z3)

  Procedure.

2.59.1 Function/Subroutine Documentation

2.59.1.1 subroutine insertinlmk_lmk (character∗6 *LMKNM1,* character∗6 *LMKNM2,* double precision *Z1,* double precision *Z2,* double precision *Z3*)

Procedure.

Parameters

z1	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z2	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z3	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM1 I Landmark file to be modified LMKNM2 I Landmark information to be inserted or updated Z1 I Landmark Ux information to be inserted or updated Z2 I Landmark Uy information to be inserted or updated Z3 I Landmark Uz information to be inserted or updated

File_I/O

Filename I/O Description

<LMKNM1>.LMK O Landmark file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP SUBROUTINE ATTACH REPLICATE_LMFILE

2.60 COMMON/INSERTINLMK_LMK_LO.f File Reference

Functions/Subroutines

- subroutine insertinlmk_lmk_lo (LMKNM1, LMKNM2, Z1, Z2, Z3)

  Procedure.

2.60.1 Function/Subroutine Documentation

2.60.1.1 subroutine insertinlmk_lmk_lo (character∗6 *LMKNM1,* character∗6 *LMKNM2,* double precision *Z1,* double precision *Z2,* double precision *Z3*)

Procedure.

Parameters

z1	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z2	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z3	Abstract This routine will insert or update the information for the landmark <LMKNM2> in the MAP OVERLAPS section of the landmark file <LMKNM1>.LMK.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM1 I Landmark file to be modified LMKNM2 I Landmark information to be inserted or updated Z1 I Landmark Ux information to be inserted or updated Z2 I Landmark Uy information to be inserted or updated Z3 I Landmark Uz information to be inserted or updated

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM1>.LMK O Lockout file for <LMKNM1>.LMK ./LMRFILE/<LMKNM1>.LMK O Landmark file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE ATTACH_LO

History 2013_10_14: Capitalization, compiler warnings fixed and header added.

2.61 COMMON/INSERTINLMK_PIC.f File Reference

Functions/Subroutines

- subroutine insertinlmk_pic (LMKNM, PICNM, IPL, PUSE)

  Procedure.

2.61.1 Function/Subroutine Documentation

2.61.1.1 subroutine insertinlmk_pic (character∗6 *LMKNM,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* logical *PUSE*)

Procedure.

Parameters

ipl	Abstract This routine will insert or update information for image <PICNM> into the PICTU↩ RES section of the landmark file <LMKNM>.LMK

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Base name of landmark file to be modified PICNM I Image name inserted or update in landmark file IPL I Image pixel and line number of landmark center PUSE I

File_I/O

Filename I/O Description

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP SHIFT

2.62 COMMON/INSERTINLMK_PIC_LO.f File Reference

Functions/Subroutines

- subroutine insertinlmk_pic_lo (LMKNM, PICNM, IPL, PUSE)

  Procedure.

2.62.1 Function/Subroutine Documentation

2.62.1.1 subroutine insertinlmk_pic_lo (character∗6 *LMKNM,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* logical *PUSE*)

Procedure.

Parameters

ipl	Abstract This routine will insert or update information for image <PICNM> into the PICTU↩ RES section of the landmark file <LMKNM>.LMK‘

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Base name of landmark file to be modified PICNM I Image name inserted or update in landmark file IPL I Image pixel and line number of landmark center PUSE I

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockfile for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK O Landmark file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None Called_by_SPC_Programs AUTOREGISTERP

History 2013_10_11: Capitalization, compiler warnings fixed and header added.

2.63 COMMON/INSERTINNOM_FRAME.f File Reference

Functions/Subroutines

- subroutine insertinnom_frame (PICNM, FRAME)

  Procedure.

2.63.1 Function/Subroutine Documentation

2.63.1.1 subroutine insertinnom_frame (character∗12 *PICNM,* character∗9 *FRAME*)

Procedure.

Abstract This routine updates the frame type for the second entry in image nominal <PICNM>.NOM file

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Base name of the nominal file to be modified FRAME I New frame variable to be updated

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be updated

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DYNAMICS

2.64 COMMON/INSERTINNOM_PSIG.f File Reference

Functions/Subroutines

- subroutine insertinnom_psig (PICNM, PSIG)

  Procedure.

2.64.1 Function/Subroutine Documentation

2.64.1.1 subroutine insertinnom_psig (character∗12 *PICNM,* double precision, dimension(3) *PSIG*)

Procedure.

Parameters

psig	Abstract This subroutine updates the pointing sigma values in the nominal file PICNM.NOM.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the nominal file to be modified PSIG I New position sigma value

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version None

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DYNAMICS

2.65 COMMON/INSERTINNOM_PTG.f File Reference

Functions/Subroutines

- subroutine insertinnom_ptg (PICNM, CX, CY, CZ)

  Procedure.

2.65.1 Function/Subroutine Documentation

2.65.1.1 subroutine insertinnom_ptg (character∗12 *PICNM,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ*)

Procedure.

Parameters

cx	Abstract The routine updates the camera pointing vectors in the PICNM.NOM nominal file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the nominal file to be modified CX I Camera X pointing vector CY I Camera Y pointing vector CZ I Camera Z pointing vector

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP REGISTER SUBROUTINE IPL2RDT

2.66 COMMON/INSERTINNOM_SOV.f File Reference

Functions/Subroutines

- subroutine insertinnom_sov (PICNM, V0)

  Procedure.

2.66.1 Function/Subroutine Documentation

2.66.1.1 subroutine insertinnom_sov (character∗12 *PICNM,* double precision, dimension(3) *V0*)

Procedure.

Parameters

v0	Abstract This routine updates the S/!!to object distance vector.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the nominal file to be modified V0 I S/!!to object distance vector

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be updated

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP REGISTER SUBROUTINE IPL2RDT

2.67 COMMON/INSERTINNOM_TRJ.f File Reference

Functions/Subroutines

- subroutine insertinnom_trj (PICNM0, PICNM1, W10, SIG)

  Procedure.

2.67.1 Function/Subroutine Documentation

2.67.1.1 subroutine insertinnom_trj (character∗12 *PICNM0,* character∗12 *PICNM1,* double precision, dimension(3) *W10,* double precision *SIG*)

Procedure.

Parameters

w10	Abstract This routine will insert image name <PICNM1>, the trajectory vector and sigma value for that image into the nominal file. PICNM1 is one of 4 surrounding images of image <PICNM0>.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

sig	Abstract This routine will insert image name <PICNM1>, the trajectory vector and sigma value for that image into the nominal file. PICNM1 is one of 4 surrounding images of image <PICNM0>.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICMN0 I Basename of the nominal file to be edited. PICMN1 I Image name of the surrounding image to <PICN↩ M0> W10 I Trajectory vector for image <PICNM1> SIG I Sigma values associated with trajectory vector.

File_I/O

Filename I/O Description

<PICNM0>.NOM O Nominal file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DYNAMICS

2.68 COMMON/INSERTINNOM_VSIG.f File Reference

Functions/Subroutines

- subroutine insertinnom_vsig (PICNM, VSIG)

  Procedure.

2.68.1 Function/Subroutine Documentation

2.68.1.1 subroutine insertinnom_vsig (character∗12 *PICNM,* double precision, dimension(3) *VSIG*)

Procedure.

Parameters

vsig	Abstract This routine will update the S/!!to object vector sigma values.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of nominal file to be edited VSIG I Sigma associated with the S/!!to object position vector

File_I/O

Filename I/O Description

<PICNM>.NOM O Nominal file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DYNAMICS

2.69 COMMON/INSERTINPIC_LIM.f File Reference

Functions/Subroutines

- subroutine insertinpic_lim (PICNM, LMKNM, IPL, SIG)

  Procedure.

2.69.1 Function/Subroutine Documentation

2.69.1.1 subroutine insertinpic_lim (character∗12 *PICNM,* character∗6 *LMKNM,* double precision, dimension(2) *IPL,* double precision *SIG*)

Procedure.

Parameters

ipl	Abstract This routine will search through an image summary file (PICNM>.SUM and either insert the landmark, center pixel, pixel sigmas in the LIMB FIT section of the file or update the landmark information if the entry already exists in the file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the image summary file to be modified LMKNM I Landmark that is visiable in image IPL I Center of landmark pixel position SIG I Sigmas associated with pixel position

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS LITHOSP SUBROUTINE LIMB_HEIGHTS REPLICATE_LMFILE

2.70 COMMON/INSERTINPIC_LIM_LO.f File Reference

Functions/Subroutines

- subroutine insertinpic_lim_lo (PICNM, LMKNM, IPL, SIG)

  Procedure.

2.70.1 Function/Subroutine Documentation

2.70.1.1 subroutine insertinpic_lim_lo (character∗12 *PICNM,* character∗6 *LMKNM,* double precision, dimension(2) *IPL,* double precision *SIG*)

Procedure.

Parameters

ipl	Abstract This routine will search through an image summary file (PICNM>.SUM and either insert the landmark, center pixel, pixel sigmas in the LIMB FIT section of the file or update the landmark information if the entry already exists in the file.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the image summary file to be modified LMKNM I Landmark that is visiable in image IPL I Center of landmark pixel position SIG I Sigmas associated with pixel position

File_I/O

Filename I/O Description

./TESTFILES/<PICNM>.SUM O Lockout file for <PICNM>.SUM ./SUMFILES/<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_11: Capitalization, compiler warnings fixed and header added.

2.71 COMMON/INSERTINPIC_LMK.f File Reference

Functions/Subroutines

- subroutine insertinpic_lmk (PICNM, LMKNM, IPL, CUSE)

  Procedure.

2.71.1 Function/Subroutine Documentation

2.71.1.1 subroutine insertinpic_lmk (character∗12 *PICNM,* character∗6 *LMKNM,* double precision, dimension(2) *IPL,* logical *CUSE*)

Procedure.

Parameters

ipl	Abstract This routine will search through an image summary file (PICNM>.SUM and either insert the landmark, center pixel in the LANDMARKS section of the file or update the landmark information if the entry already exists in the file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of summary file to be modified LMKNM I Name of landmark to be updated IPL I Pixel of landmark center CUSE I Use or not logi!!variable

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP SHIFT SUBROUTINE REP↩ LICATE_LMFILE

2.72 COMMON/INSERTINPIC_LMK_LO.f File Reference

Functions/Subroutines

- subroutine insertinpic_lmk_lo (PICNM, LMKNM, IPL)

  Procedure.

2.72.1 Function/Subroutine Documentation

2.72.1.1 subroutine insertinpic_lmk_lo (character∗12 *PICNM,* character∗6 *LMKNM,* double precision, dimension(2) *IPL*)

Procedure.

Parameters

ipl	Abstract This routine will search through an image summary file (PICNM>.SUM and either insert the landmark, center pixel in the LANDMARKS section of the file or update the landmark information if the entry already exists in the file.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of summary file to be modified LMKNM I Name of landmark to be updated IPL I Pixel of landmark center CUSE I Use or not logi!!variable

File_I/O

Filename I/O Description

./TESTFILES/<PICNM>.SUM O Lockout file for <PICNM>.SUM ./SUMFILES/<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_11: Capitalization, compiler warnings fixed and header added.

2.73 COMMON/INSERTINPIC_POLE.f File Reference

Functions/Subroutines

- subroutine insertinpic_pole (PICNM, Z1, Z2, Z3)

  Procedure.

2.73.1 Function/Subroutine Documentation

2.73.1.1 subroutine insertinpic_pole (character∗12 *PICNM,* double precision *Z1,* double precision *Z2,* double precision *Z3*)

Procedure.

Parameters

z1	Abstract This routine will replace the DISTORTION data line in the summary file with the RA, Dec, and Twist Pole information.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z2	Abstract This routine will replace the DISTORTION data line in the summary file with the RA, Dec, and Twist Pole information.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

z3	Abstract This routine will replace the DISTORTION data line in the summary file with the RA, Dec, and Twist Pole information.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of summary file to be modified Z1 I Pole right ascension Z2 I Pole declination Z3 I Pole twist

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP REGISTER SUBROUTINE IPL2RDT

2.74 COMMON/INSERTINPIC_PTG.f File Reference

Functions/Subroutines

- subroutine insertinpic_ptg (PICNM, CX, CY, CZ)

  Procedure.

2.74.1 Function/Subroutine Documentation

2.74.1.1 subroutine insertinpic_ptg (character∗12 *PICNM,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ*)

Procedure.

Parameters

cx	Abstract This routine will update the camera pointing vector entries in the PICNM.SUM summary file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the summary file to modified CX I X value of the camera pointing vector CY I Y value of the camera pointing vector CZ I Z value of the camera pointing vector

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LITHOS REGISTER SUBROUTINE IPL2RDT

2.75 COMMON/INSERTINPIC_SIG.f File Reference

Functions/Subroutines

- subroutine insertinpic_sig (PICNM, SIGV, SIGP)

  Procedure.

2.75.1 Function/Subroutine Documentation

2.75.1.1 subroutine insertinpic_sig (character∗12 *PICNM,* double precision, dimension(3) *SIGV,* double precision, dimension(3) *SIGP*)

Procedure.

Parameters

sigv	Abstract This routine will replace the following sigma line entries in PICNM.SUM summary file: 1. S/!!to object vector (SIGMA_VS0) 2. Camera pointing vector (SIGMA-PTG)

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the summary file to be modified SIGV I Sigma values for the S/!!to object position vector SIGP I Sigma values for the Camera pointing vector

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.76 COMMON/INSERTINPIC_SOV.f File Reference

Functions/Subroutines

- subroutine insertinpic_sov (PICNM, V0)

  Procedure.

2.76.1 Function/Subroutine Documentation

2.76.1.1 subroutine insertinpic_sov (character∗12 *PICNM,* double precision, dimension(3) *V0*)

Procedure.

Parameters

v0	Abstract The routine will replace the S/!!to object vector values in summary file <PICNM>.SUM with V0.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICMN I Basename of the summary file to be updated V0 I S/!!to object vector data

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LITHOS REGISTER SUBROUTINE IPL2RDT

2.77 COMMON/INSERTINPIC_SZ.f File Reference

Functions/Subroutines

- subroutine insertinpic_sz (PICNM, SZ)

  Procedure.

2.77.1 Function/Subroutine Documentation

2.77.1.1 subroutine insertinpic_sz (character∗12 *PICNM,* double precision, dimension(3) *SZ*)

Procedure.

Parameters

sz	Abstract This routine update the sun vector values in the summary file <PICNM>.SUM

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008) Declarations Variable_I/O

Variable I/O Description

PICNM I Basename of the summary file to be modified SZ I

File_I/O

Filename I/O Description

<PICNM>.SUM I Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP REGISTER SUBROUTINE IPL2RDT

2.78 COMMON/INSERTINPIC_VFLG.f File Reference

Functions/Subroutines

- subroutine insertinpic_vflg (PICNM, VUSE)

  Procedure.

2.78.1 Function/Subroutine Documentation

2.78.1.1 subroutine insertinpic_vflg (character∗12 *PICNM,* logical *VUSE*)

Procedure.

Abstract

Some images do not participate in the determination of landmark positions. These are images marked with a 'T' as the 16th character of the first record. To date, only Mariner 10 images of Mercury are used in this way, !! since the spacecraft state is not well known. However, it might make sense to treat low resolution images in this way since we would still like to keep them for trajectory reconstruction. This routine will update this Vlm use flag in the <PICNM>.SUM summary file,

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Basename of summary file to be modified VUSE I Use or no-use logical variable

File_I/O

Filename I/O Description

<PICNM>.SUM O Summary file to be modified

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS REGISTER

2.79 COMMON/INTERPOLATE.f File Reference

Functions/Subroutines

- subroutine interpolate (Q, DQ, VEC)

  Procedure.

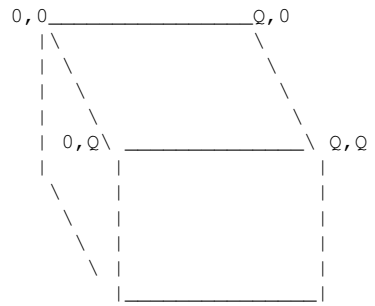
2.79.1 Function/Subroutine Documentation

2.79.1.1 subroutine interpolate (integer *Q,* integer *DQ,* double precision, dimension(3,0:512,0:512,6) *VEC*)

Procedure.

Abstract

In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. The grid labels run from 0 to Q, with Q a power of 2.




```
This subroutine is called by the DENSIFY programs, which multiply an
initial Q by a factor K=2^n, so that Q -> KQ and points in the grid are
now spaced by DQ=K.  A cell of the original model on one of the new grids
has a spacing of one unit, but has vectors defined every DQ units (o).
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

Q DQ VEC

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called UCRSS VADD VEQU VSCL VSUB

Called_by_SPC_Programs DENSIFY DENSIFYA

History 2013_10_18: Capitalization, compiler warnings fixed and header added.

2.80 COMMON/INVERTN.f File Reference

Functions/Subroutines

- subroutine invertn (N, M, MINV)

  Procedure.

2.80.1 Function/Subroutine Documentation

2.80.1.1 subroutine invertn (integer∗4 *N,* real∗8, dimension(6,6) *M,* real∗8, dimension(6,6) *MINV*)

Procedure.

Parameters

n	Abstract This subroutine inverts a symmetri!!NxN matrix up to rank 6. It first uses a Cholesky decomposition to write M = UtDU where U is upper triangular with unit diagonal elements and D is diagonal. It then inverts U and (trivially) D to form the inverse. The matrix is given a maximum rank of 6 because that is all we need, although there is no reason it cannot be greater.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

N I Rank of the matrix to be inverted (N<=6) M I Matrix to be inverted MINV O Inverted matrix

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS REGISTER RESIDUALS POLE SUBROUT↩ INE FIND_TMPL SUBROUTINE IPL2RDT SUBROUTINE IPL2SCOBJPTG SUBROUTINE PATCH_COORDS

Notes: 6/11/13: Modification to ensure matrix rank, N, doesn not excede 6

2.81 COMMON/IPL2RDT.f File Reference

Functions/Subroutines

- subroutine ipl2rdt (PICNM, NITER, VERBOSE, RA, DEC, TWIST, FOUND)

  Procedure.

2.81.1 Function/Subroutine Documentation

2.81.1.1 subroutine ipl2rdt (character∗12 *PICNM,* integer *NITER,* logical *VERBOSE,* double precision *RA,* double precision *DEC,* double precision *TWIST,* logical *FOUND*)

Procedure.

Abstract

This subroutine determines the transformation from inertial space to body-fixed space for an image by using landmarks fixed on the body to solve for spacecraft position V0 and camera orientation Ci and determining the transformation to these coordinates from those in inertial space.

The estimation for V0 and Ci is taken from the subroutine IPL2SCOBJPTG, but it contatns none of the external constraints. There are inherent errors due to aliasing between pointing and transverse s/!!position errors that would be minimized with the use of wide-angle imagers like NAVCAM.

The inertial space values W0, Qi are obtained from the file IFRAME.TXT that will be created by the make_sumfiles program.

The matrix M transforms inertial space vectors Qi to body-fixed vectors Ci with MQi = Ci or:

```
   M(Q1,Q2,Q3)=(C1,C2,C3)

or its transpose:

  |Q1|       |C1|
  |Q2| Mt =  |C2|    written as  |MA| |Mt| = |MD|
  |Q3|       |C3|

so

  |Mt| = |inv MA| |MD| = |MAt| |MD|

and M is the transpose of this.

This software is still subject to change since it is part of the ongoing
effort to prepare for the wobble contingency.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Name of image whose SUMFILE is processed LFLAG I If 'y' value limb apparations of landmarks are used

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I IFRAME.TXT I LMKFILES/<LMKNM>.LMK I NOMINALS/<PICNM>.NOM I SUMFILES/<↩ PICNM>.SUM I

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called LMCOUNT SLEN

SPC_subroutines_called GETIU GETMM INVERTN MMPX

SPICELIB_functions_called CLIGHT HALFPI LDPOOL RPD SPD VDOT VNORM

SPICELIB_subroutines_called MXV MTXV PXFORM UCRSS UTC2ET VHAT VMINUS VSUB

Called_by_SPC_Programs GEOMETRY SUBROUTINE SOLVE_RDT

History 2014_02_02 RETURN if < 5 landmarks.

2.82 COMMON/IPL2SCOBJPTG.f File Reference

Functions/Subroutines

- subroutine ipl2scobjptg (PICNM, LFLAG)

  Procedure.

2.82.1 Function/Subroutine Documentation

2.82.1.1 subroutine ipl2scobjptg (character∗12 *PICNM,* character∗1 *LFLAG*)

Procedure.

Abstract

This subroutine updates the camera pointing and spacecraft - object vector in a SUMFILE from landmark image space locations both on the body and the limb and from several external constraints. It also produces formal uncertainties in these updates.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Name of image whose SUMFILE is processed LFLAG I If 'y' value limb apparations of landmarks are used

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I LA_SCOBJ.TXT I LMKFILES/<LMKNM>.LMK I NOMINALS/<PICNM>.NOM I POLE.TXT I SUMFILES/<NAME>.SUM I SUMFILES/<PICNM>.SUM I

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called LMCOUNT SLEN

SPC_subroutines_called GETIU GETMM INVERTN MMPX

SPICELIB_functions_called CLIGHT HALFPI LDPOOL RPD SPD VDOT VNORM

SPICELIB_subroutines_called MXV MTXV PXFORM UCRSS UTC2ET VHAT VMINUS VSUB

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS

History 2014_02_01 External LIDAR constraint modified. 2014_02_01 Input flag PFLAG renamed to LFLAG. 2014_02_02 RETURN if no landmarks. 2014_04_20 WOBBLE=.TRUE. constrains VDOT(C,V0) to inertial value. 2014_04_20 Existance test for INIT_LITHOS.TXT removed. 2014_04_30 Wobble now computes sigmas from N↩ OMINAL file.

2.83 COMMON/IPL2SCOBJPTG_LO.f File Reference

Functions/Subroutines

- subroutine ipl2scobjptg_lo (PICNM, LFLAG)

  Procedure.

2.83.1 Function/Subroutine Documentation

2.83.1.1 subroutine ipl2scobjptg_lo (character∗12 *PICNM,* character∗1 *LFLAG*)

Procedure.

AbstractF

This subroutine updates the camera pointing and spacecraft - object vector in a SUMFILE from landmark image space locations both on the body and the limb and from several external constraints. It also produces formal uncertainties in these updates.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I LFLAG I

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I SP!!toolkit parameter file. ./SUMFILES/<PICNM>.SUM I&O Image <PICNM> summary file. ./NOMINALS/<PICNM>.NOM I Image <PICNM> nominal pointing file. ./TESTFILES/<LMKNM>.LMK I Lockout file for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK I Landmark summary file

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called LMCOUNT_LO SLEN

SPC_subroutines_called DELAY GETMM INVERTN MMPX

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called UCRSS VHAT VMINUS

Called_by_SPC_Programs AUTOREGISTERP

History 09_16_2013 LMCOUNT_LO replaces LMCOUNT 10_04_2013 Capitalization, header added, compiler warnings checked and fixed. 02_01_2014 Removed APR_POS, APR_PTG and INIT_LITHOS test. 04_20_↩ 2014 WOBBLE=.TRUE. constrains VDOT(C,V0) to inertial value. 2014_04_30 Wobble now computes sigmas from NOMINAL file.

2.84 COMMON/IPL2VLM.f File Reference

Functions/Subroutines

- subroutine ipl2vlm (LMKNM)

  Procedure.

2.84.1 Function/Subroutine Documentation

2.84.1.1 subroutine ipl2vlm (character∗6 *LMKNM*)

Procedure.

Abstract

This subroutine updates the body-fixed location for landmark LMKNM from pixel/line locations of the landmark in an number of images, including limb apparitions, and constraints due to topography sharing with adjacent landmarks and restriction to the surface of a pre-existing shape model. It also produces formal uncertainties in these positions.

A version of this subroutine is used in the parallel version of LITHOS. As such, it is more efficient to have the L↩ MKNM.LMK file opened for as short a time as possible. The records containing the three constraints in the file LMKNM.LMK are read and saved and the file closed immediately, before any further processing.

Some images do not participate in the determination of landmark positions. These are images marked with a 'T' as the 16th character of the first record. To date, only Mariner 10 images of Mercury are used in this way, !! since the spacecraft state is not well known. However, it might make sense to treat low resolution images in this way since we would still like to keep them for trajectory reconstruction.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark name to be processed.

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I SP!!toolkit parameter file LMKFILES/<LMKNM>.LMK I Landmark file for LMKNM LMKFILE↩S/<NAME>.LMK I Landmark file for NAME MAPFILES/<LMKNM>.MAP I Maplet file for LNKNM SUMFILES/<↩PICNM>.SUM I Summary file for PICNM

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called PXCOUNT SLEN

SPC_subroutines_called FLIP GET_REF GETMM INVERTN MMPX READ_MAP U2VRF

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called VHAT VSUB

Called_by_SPC_Programs GEOMETRY LITHOS

2.85 COMMON/IPL2VLM_LO.f File Reference

Functions/Subroutines

- subroutine ipl2vlm_lo (LMKNM)

  Procedure.

2.85.1 Function/Subroutine Documentation

2.85.1.1 subroutine ipl2vlm_lo (character∗6 *LMKNM*)

Procedure.

Abstract

This subroutine updates the body-fixed location for landmark LMKNM from pixel/line locations of the landmark in an number of images, including limb apparitions, and constraints due to topography sharing with adjacent landmarks and restriction to the surface of a pre-existing shape model. It also produces formal uncertainties in these positions.

This subroutine is used in the parallel version of LITHOS. As such, it is more efficient to have the LMKNM.LMK file opened for as short a time as possible. The records containing the three constraints in the file LMKNM.LMK are read and saved and the file closed immediately, before any further processing.

Some images do not participate in the determination of landmark positions. These are images marked with a 'T' as the 16th character of the first record. To date, only Mariner 10 images of Mercury are used in this way, !! since the spacecraft state is not well known. However, it might make sense to treat low resolution images in this way since we would still like to keep them for trajectory reconstruction.

This subroutine is identical to IPL2VLM except that lockout files are used in order to avoid inadvertant reading or writing to a file that is opened by another process.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark name to be processed.

File_I/O

Filename I/O Description

INIT_LITHOS.TXT I SP!!toolkit parameter file TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>.LMK LMKFILES/<LMKNM>.LMK I Landmark file for LMKNM TESTFILES/<NAME>.LMK O Lockout file for <NA↩ ME>.LMK LMKFILES/<NAME>.LMK I Landmark file for NAME TESTFILES/<LMKNM>.MAP O Lockout file for <LMKNM>.MAP MAPFILES/<LMKNM>.MAP I Maplet file for LNKNM TESTFILES/<PICNM>.SUM O Lockout file <PICNM>.SUM SUMFILES/<PICNM>.SUM I Summary file for PICNM

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called PXCOUNT_LO SLEN

SPC_subroutines_called DELAY FLIP GET_REF GETMM INVERTN MMPX READ_MAP U2VRF

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called VHAT VSUB

Called_by_SPC_Programs LITHOSP

History 2013_10_09: Capitalization, compiler warnings fixed and header added.

2.86 COMMON/LIMB_HEIGHTS.f File Reference

Functions/Subroutines

- subroutine limb_heights (LMKNM, NTMP, HT, HUSE, SAVE)

  Procedure.

2.86.1 Function/Subroutine Documentation

2.86.1.1 subroutine limb_heights (character∗6 *LMKNM,* integer *NTMP,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* logical *SAVE*)

Procedure.

Parameters

ntmp	Abstract This subroutine searches for limb apparitions of maplets, points of a maplet where the topography is illuminated and is tangent to the line of sight in an image. Two results are returned depending on the value of the logical variable SAVE. If SAVE=.FALSE. any limb points found in a maplet are used as constraints in the subroutine SLP2HGT that determines topography from slopes in LITHOS. If SAVE=.TRUE., the deviations of the limbpoints from the corresponding surface points are used to estimate the pixel/line position of the maplet center in the image and to constrain the landmark vector solution the spacecraft state solution in the direction normal to the limb. If SAVE=.TRUE. the user has the option, almost always used, of clearing the existing limb data from the landmark file and the corresponding .SUM files. There are three user input search parameters. ZEXP is an expansion parameter that is always set to 1 but is kept for backward compatibility with old scripts. ZRES restricts the image RESOLUTION (in km/px) relative to the maplet SCALE: RESOLUTION < ZRES*SCALE. ZHGT is the maximum allowed difference between predicted limb point height on a maplet and the measured height in units of maplet SCALE. After a number of quick checks to reject images that cannot have a lit maplet on the limb, there are three initial tests to veto the limb search for a maplet pixel. The first is the scale veto discussed above and set by ZRES. The second is a global veto that searches the shape model along the line of sight to the maplet point to see if any part of that model obstructs the potential limb point. For those maplet pixels and images that have survived all vetoes, the subroutine now searches for the height of the limb above or below the predicted maplet position by moving from 15 image pixels below the prediction to 15 pixels above and searching for a maximum in the brightness gradient B(L)-B(L+1). The brightness used is actually an average over a +/- 1 pixel swath perpendicular to the line of sight and the limb normal. If the maximum is further from the predicted height than ZHGT*SCALE the limb point is rejected. After a gauntlet of tests to reject lower quality limb fits, the actual limb height is obtained from a quadrati!!fit to the maximum slope and its nearest neighbors. One of the tests for the limb is that data off the limb (space) be dark enough. In order to handle the possibility that there could be blooming that brightens pixels near the body, a parameter BLOOM set in INIT_LITHOS.TXT sets the dark limit higher by a fraction of the nearby on-body value. A possible future change to this subroutine is the following: As the software cycles through the images it determines heights at maplet pixels from each image. If several images determine a height at the same pixel, only the final image's height is used. This is generally a rare event, so such a change is not a priority.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM NTMP HT HUSE SAVE

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I SP!!toolkit parameter file ./LIMBLIST.TXT or ./PICTLIST.TXT I List of images. ./LMKFILES/<LMKNM>.LMK I Landmark file for LMKNM ./SUMFILES/<PICFILE>.SUM I Summary file for PICFILE

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called BRT1 LMCOUNT MAXLEN SLEN

SPC_subroutines_called DEBLEMISH DELETEINLMK_LIM DELETEINPIC_LIM DENOISE FIND_PICFILE GET_HEIGHTS GET_MAP INSERTINLMK_LIM INSERTINPIC_LIM PICINPT READ_HEADER V2IMGPL

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called UCRSS VADD VHAT

Called_by_SPC_Programs LITHOS

History 2013_10_18: Capitalization, compiler warnings fixed and header added. 2014_02_12: Constant P(i,j) replaced with EPS. 2014_05_12: Limb cleared before SAVE in case clear='n' and NSM<10

2.87 COMMON/LIMB_HEIGHTS_LO.f File Reference

Functions/Subroutines

- subroutine limb_heights_lo (LMKNM, NTMP, HT, HUSE, SAVE)

  Procedure.

2.87.1 Function/Subroutine Documentation

2.87.1.1 subroutine limb_heights_lo (character∗6 *LMKNM,* integer *NTMP,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* logical *SAVE*)

Procedure.

Parameters

ntmp	Abstract This subroutine searches for limb apparitions of maplets, points of a maplet where the topography is illuminated and is tangent to the line of sight in an image. Two results are returned depending on the value of the logical variable SAVE. If SAVE=.FALSE. any limb points found in a maplet are used as constraints in the subroutine SLP2HGT that determines topography from slopes in LITHOS. If SAVE=.TRUE., the deviations of the limbpoints from the corresponding surface points are used to estimate the pixel/line position of the maplet center in the image and to constrain the landmark vector solution the spacecraft state solution in the direction normal to the limb. If SAVE=.TRUE. the user has the option, almost always used, of clearing the existing limb data from the landmark file and the corresponding .SUM files. There are three user input search parameters. ZEXP is an expansion parameter that is always set to 1 but is kept for backward compatibility with old scripts. ZRES restricts the image RESOLUTION (in km/px) relative to the maplet SCALE: RESOLUTION < ZRES*SCALE. ZHGT is the maximum allowed difference between predicted limb point height on a maplet and the measured height in units of maplet SCALE. After a number of quick checks to reject images that cannot have a lit maplet on the limb, there are three initial tests to veto the limb search for a maplet pixel. The first is the scale veto discussed above and set by ZRES. The second is a global veto that searches the shape model along the line of sight to the maplet point to see if any part of that model obstructs the potential limb point. For those maplet pixels and images that have survived all vetoes, the subroutine now searches for the height of the limb above or below the predicted maplet position by moving from 15 image pixels below the prediction to 15 pixels above and searching for a maximum in the brightness gradient B(L)-B(L+1). The brightness used is actually an average over a +/- 1 pixel swath perpendicular to the line of sight and the limb normal. If the maximum is further from the predicted height than ZHGT*SCALE the limb point is rejected. After a gauntlet of tests to reject lower quality limb fits, the actual limb height is obtained from a quadrati!!fit to the maximum slope and its nearest neighbors. One of the tests for the limb is that data off the limb (space) be dark enough. In order to handle the possibility that there could be blooming that brightens pixels near the body, a parameter BLOOM set in INIT_LITHOS.TXT sets the dark limit higher by a fraction of the nearby on-body value. A possible future change to this subroutine is the following: As the software cycles through the images it determines heights at maplet pixels from each image. If several images determine a height at the same pixel, only the final image's height is used. This is generally a rare event, so such a change is not a priority. This subroutine is identical to LIMB_HEIGHTS except that lockout files are used in order to avoid inadvertant reading or writing to a file that is opened by another process.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM NTMP HT HUSE SAVE

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I SP!!toolkit parameter file ./LMKFILES/<LMKNM>.LMK I Landmark file for LMKNM ./TEST↩ FILES/<LMKNM>.LMK I Lockfile file for LMKNM.LMK ./TESTFILES/<LMKNM>.MAP I Lockfile file for LMKNM.↩

MAP ./LIMBLIST.TXT or ./PICTLIST.TXT I List of images ./SUMFILES/<PICNM>.SUM I Summary file for PICNM ./TESTFILES/<PICNM>.SUM I Summary file for PICNM.SUM ./BLEMISHES/<PICNM>.BLM I Blemish file for PICNM ./TESTFILES/<PICNM>.BLM I Summary file for PICNM.BLM

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called BRT1 LMCOUNT_LO MAXLEN_LO SLEN

SPC_subroutines_called DEBLEMISH DELAY DELETEINLMK_LIM_LO DELETEINPIC_LIM_LO DENOISE FIND_PICFILE GET_HEIGHTS_LO GET_MAP_LO INSERTINLMK_LIM_LO INSERTINPIC_LIM_LO PICINPT_LO READ_HEADER V2IMGPL

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called UCRSS VADD VHAT

Called_by_SPC_Programs LITHOSP

History 2013_10_18: Capitalization, compiler warnings fixed and header added. 2014_02_12: Constant P(i,j) replaced with EPS. 2014_02_12: Peak height limit made consistent with LIMB_HEIGHTS subroutine. 2014_02_26: Bug fix BRT1(Z0,ZZ,KK,DN0,ESZ) replaces BRT1(Z0,ZZ,KK,DN0) 2014_05_12: Limb cleared before SAVE in case clear='n' and NSM<10

2.88 COMMON/LMCOUNT.f File Reference

Functions/Subroutines

- integer function lmcount (PICNM, K)

  Procedure.

2.88.1 Function/Subroutine Documentation

2.88.1.1 integer function lmcount (character∗12 *PICNM,* integer *K*)

Procedure.

Returns

Abstract

```
This function will count the landmarks and limbmarks in the
given image <PICNM> summary file.

  If K=0 the routine counts landmarks in the image
  If K=1 the routine counts landmarks + limbmarks in the image
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Image to count the landmarks. K I K=0 count landmarks, K=1 count landmarks/limbmarks

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I Image summary file

Restrictions PICTNM summary must exits.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS RESIDUALS POLE SUBROUTINE IMP2SC↩ OBJPTG SUBROUTINE LIMB_HEIGHTS

Notes: 6/11/13: Added check for summary file prior to opening.

2.89 COMMON/LMCOUNT_LO.f File Reference

Functions/Subroutines

- integer function lmcount_lo (PICNM)

  Procedure.

2.89.1 Function/Subroutine Documentation

2.89.1.1 integer function lmcount_lo (character∗12 *PICNM*)

Procedure.

Returns

Required_Reading

```
R.W. Gaskell, et.al, "Characterizing and navigating small bodies
      with imaging data", Meteoritics & Planetary Science 43,
      Nr 6, 1049-1061 (2008)
```

Declarations Variable_I/O

Variable I/O Description

PICNM I Image to count the landmarks.

File_I/O

Filename I/O Description

./SUMFILES/<PICNM>.SUM I Image summary file for <PICNM> ./TESTFILES/<PICNM>.SUM O Lockout file for <PICNM>.SUM

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

History 2013_10_09: Capitalization, compiler warnings fixed and header added.

2.90 COMMON/LOC2PIX.f File Reference

Functions/Subroutines

- subroutine loc2pix (PICNM, V, UX, UY, UZ, SCALE, DPIC_DLOC)

  Procedure.

2.90.1 Function/Subroutine Documentation

2.90.1.1 subroutine loc2pix (character∗12 *PICNM,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *SCALE,* double precision, dimension(2,3) *DPIC_DLOC*)

Procedure.

Parameters

v	Abstract This subroutine computes the change in the image-space location (p,l) of a point (x,y,z) on a maplet as the point is moved in the maplet frame: dp/dx dp/dy dp/dz dl/dx dl/dy dl/dz The derivative are obtained numerically by steppint +/- 5 map pixels from the center in each of the three defining directions of the maplet system and differencing the predicted image space locations.

Disclaimer

The partial derivatives are computed numerically, with a step size of 5 map pixels. This value could be optomized in the future.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Image name for the summary file V O UX O UY O UZ O SCALE O DPIC_DLO!! O

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I Summary file for PICNM

Restrictions Summary file for PICNM must exist.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called V2IMGPL

SPICELIB_functions_called None

SPICELIB_subroutines_called VADD VSCL VSUB

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER RESIDUALS SUBROUTINE EXTRACT_DAT↩A_PIC FUNCTION PICRES

Notes: 6/11/13: Added checks for the existence of PICTFILE

2.91 COMMON/LOC2PIX_LO.f File Reference

Functions/Subroutines

- subroutine loc2pix_lo (PICNM, V, UX, UY, UZ, SCALE, DPIC_DLOC)

  Procedure.

2.91.1 Function/Subroutine Documentation

2.91.1.1 subroutine loc2pix_lo (character∗12 *PICNM,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *SCALE,* double precision, dimension(2,3) *DPIC_DLOC*)

Procedure.

Parameters

v	Abstract This subroutine computes the change in the image-space location (p,l) of a point (x,y,z) on a maplet as the point is moved in the maplet frame: dp/dx dp/dy dp/dz dl/dx dl/dy dl/dz The derivative are obtained numerically by steppint +/- 5 map pixels from the center in each of the three defining directions of the maplet system and differencing the predicted image space locations. This subroutine employs a lockout file to guarantee that the file PICNM.SUM cannot be opened simultaneously with another process that might be attampting to write to it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Image name for the summary file V O UX O UY O UZ O SCALE O DPIC_DLO!! O

File_I/O

Filename I/O Description

./TESTFILES/<PICNM>.SUM O Lockout file for <PICNM>.SUM ./SUMFILES//<PICNM>.SUM I PICNM image summary file

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY V2IMGPL

SPICELIB_functions_called None

SPICELIB_subroutines_called VADD VSCL VSUB

Called_by_SPC_Programs LITHOSP SUBROUTINE EXTRACT_DATA_PIC_LO

History 2013_10_09: Capitalization, compiler warnings fixed and header added.

2.92 COMMON/MAKE_TMPLIST_LO.f File Reference

Functions/Subroutines

- subroutine make_tmplist_lo (LMKNM, USR)

  Procedure.

2.92.1 Function/Subroutine Documentation

2.92.1.1 subroutine make_tmplist_lo (character∗6 *LMKNM,* character∗2 *USR*)

Procedure.

Parameters

usr	Abstract This subroutine creates a list of image files found to have a given maplet occuring on their limbs by reading the LIMB_FITS records of THE LMKNM.LMK file. The purpose is to restrict the limb search time for maplet iteration in parallel mode. The complete search can be carried out by including the record NEWLIM=.TRUE. in the INIT_LITHOS.TXT file. This subroutine employs a lockout file to guarantee that the file LMKNM.LMK cannot be opened simultaneously with another process that might be attampting to write to it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Landmark name USR I

File_I/O

Filename I/O Description

./LMKFILES/<LMKNM>.LMK I ./TESTFILES/<LMKNM>.LMK O Lockout file for /<LMKNM>.LMK ./TMPLIS↩T/<USR>.TXT O

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_18: Capitalization, compiler warnings fixed and header added.

2.93 COMMON/MATCHUP.f File Reference

Functions/Subroutines

- subroutine matchup (SZ, I1, J1, F1, I2, J2, F2)

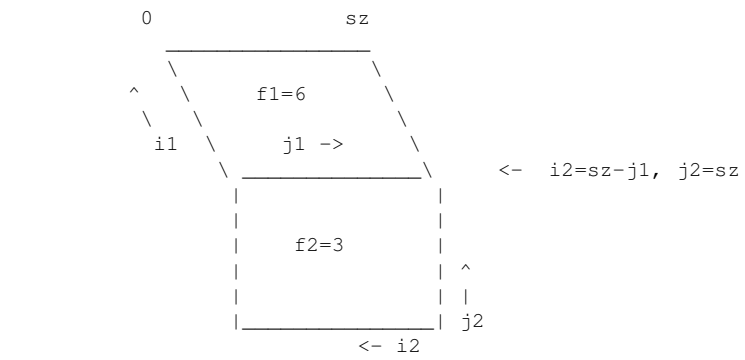
  Procedure.

2.93.1 Function/Subroutine Documentation

2.93.1.1 subroutine matchup (integer∗4 *SZ,* integer∗4 *I1,* integer∗4 *J1,* integer∗4 *F1,* integer∗4 *I2,* integer∗4 *J2,* integer∗4 *F2*)

Procedure.

Parameters

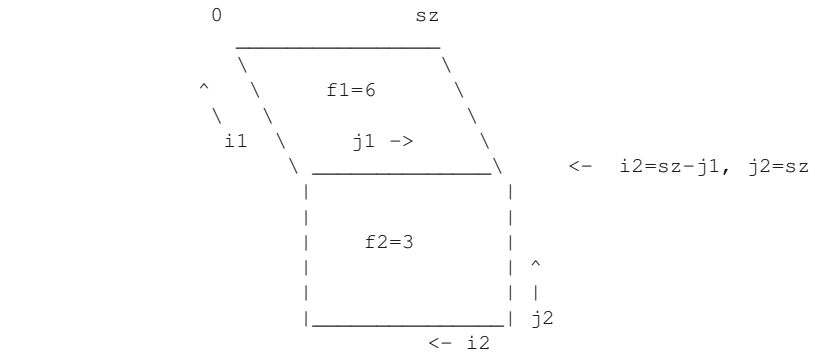
sz	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

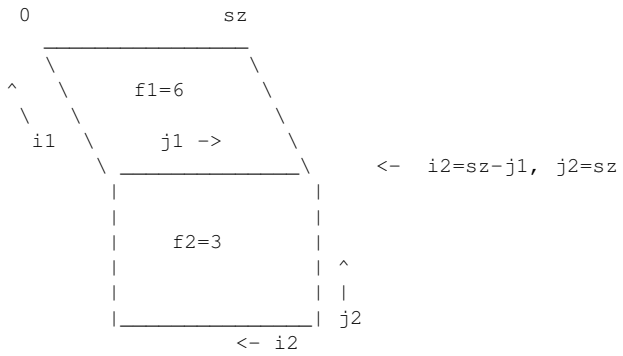
i1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

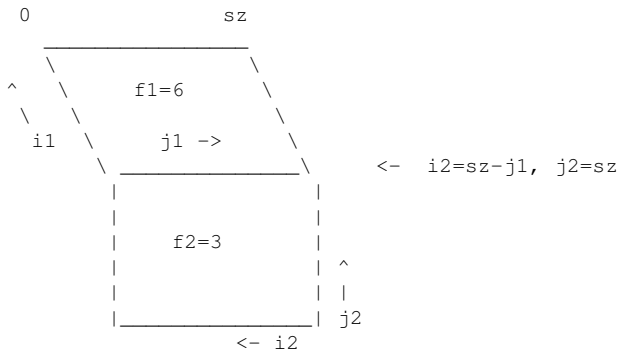
j1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

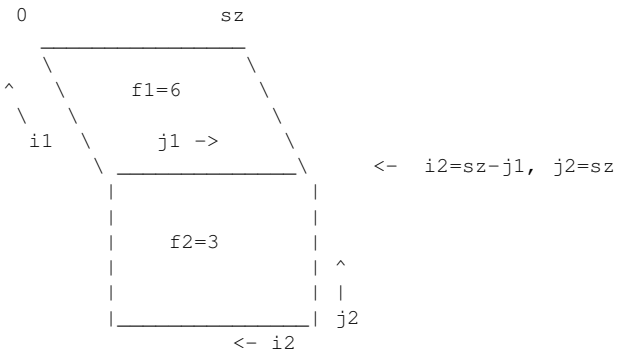
f1	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

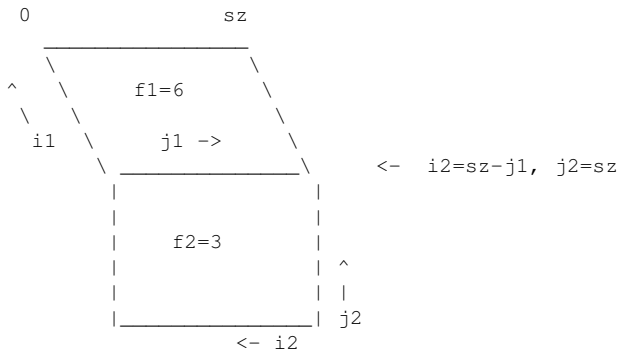
i2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

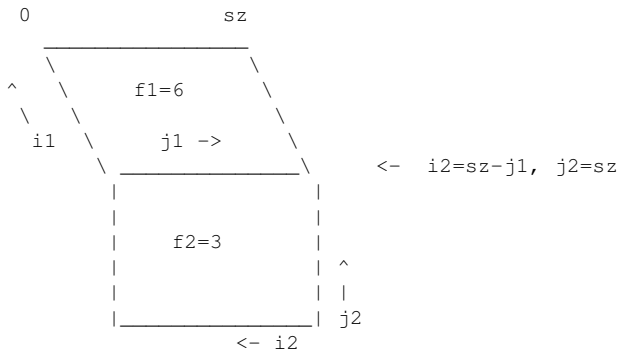
j2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

f2	Abstract In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. Each edge of the cube contains common points from two faces. The purpose of this subroutine is to identify which points on faces sharing a common edge match up with each other. SZ is the size of a grid, with i,j ranging from 0 to SZ. For example, if i=0 on face number 6, MATCHUP(SZ, 0,J1,6, I2,J2,F2) would return I2=SZ-J1, J2=SZ on face F2=3 as shown below:  The subroutine is often used to guarantee that values computed at these common points are the same.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

SZ I1 J1 F1 I2 J2 F2

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DENSIFY DENSIFYA VECS2SHAPE SUBROUTINE RELAXATION

History 2013_10_18: Capitalization, compiler warnings fixed and header added.

2.94 COMMON/MAXLEN.f File Reference

Functions/Subroutines

- double precision function maxlen ()

  Procedure.

2.94.1 Function/Subroutine Documentation

2.94.1.1 double precision function maxlen ( )

Procedure.

Abstract

This function will calculate the maximum length between 10000 randomly generated pairs of the shape model geometri!!points. It is used to estimate the maximum size of the body, both for displays and to set the scale for various visibility tests.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

MAXLEN O Maximum length between two randomly choosen geometri!!points on the shape model.

File_I/O

Filename I/O Description

SHAPE.TXT I ASCII text shape model.

Restrictions SHAPE.TXT must exist.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN

SPC_subroutines_called GET_MODEL

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called VEQU VSUB

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP DENSIFY SUBROUTINE L↩ IMB_HEIGHTS

History 2013_10_07: Capitalization, compiler warnings fixed and header added. 2014_03_03: WRITE statement commented out.

2.95 COMMON/MAXLEN_LO.f File Reference

Functions/Subroutines

- double precision function maxlen_lo ()

  Procedure.

2.95.1 Function/Subroutine Documentation

2.95.1.1 double precision function maxlen_lo ( )

Procedure.

Abstract

This function will calculate the maximum length between 10000 randomly generated pairs of the shape model geometri!!points. It is used to estimate the maximum size of the body, both for displays and to set the scale for various visibility tests.

It includes a (probably unnecessary) lockout procedure that prevents two processes from reading the shape model at the same time.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

MAXLEN O Maximum length between two randomly choosen geometri!!points on the shape model.

File_I/O

Filename I/O Description

./TESTFILES/SHAPE.TXT O Lockout file for SHAPE.TXT ./SHAPEFILES/SHAPE.TXT I ASCII text shape model.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN

SPC_subroutines_called GET_MODEL

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called VEQU VSUB

Called_by_SPC_Programs SUBROUTINE LIMB_HEIGHTS_LO

History 2013_10_07: Capitalization, compiler warnings fixed and header added. 2014_03_03: WRITE statement commented out.

2.96 COMMON/MM2VN.f File Reference

Functions/Subroutines

- subroutine mm2vn (MM, MMFL, V0, CX, CY, CZ, USE, V, N)

  Procedure.

2.96.1 Function/Subroutine Documentation

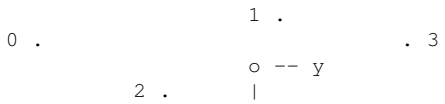
2.96.1.1 subroutine mm2vn (double precision, dimension(2) *MM,* double precision *MMFL,* double precision, dimension(3) *V0,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(3) *V,* double precision, dimension(3) *N*)

Procedure.

Abstract

This subroutine predicts the vector to a landmark center at a given gnomonic projection on the focal plane based on the current shape model. It also estimates the normal to the surface N at the landmark.

The unit vector UZ = W/|W|, W = MM(1)∗CX + MM(2)∗CY + MMFL∗CZ, where Ci are the camera pointing unit vectors, pierces the reference shape model within, or on the boundary of, one of the quadrilateral cells. Projected onto a plane with unit vectors UX and UY perpendicular to UZ, the corners of the cell are at X(I), Y(I), I=0,3.



```
                          x
```

The conditions for UZ to pierce the cell are:

```
X(0)*Y(1)-Y(0)*X(1) < 0         X(1)*Y(3)-Y(1)*X(3) < 0
X(3)*Y(2)-Y(3)*X(2) < 0         X(2)*Y(0)-Y(2)*X(0) < 0
```

the first, for example, being the cross product of the o0 and o1 vectors (in the UZ direction).

Once a cell has been determined, the coordinates of the piercing point o are determined by bilinear interpolation. The corner points have labels (i,j) where 0=(0,0), 1=(1,0), 2=(0,1), 3=(1,1) and the bilinear interpolation in terms of i,j takes the form:

```
F(i,j) = F0 + F1*i + F2*j + F3*i*j
```

where, for example,

```
F0 = F(0)
F1 = F(1)-F(0)
F2 = F(2)-F(0)
F3 = F(0)-F(1)-F(2)+F(3)
```

The piercing point is at (0,0,Zo) where Zo is the magnitude of the vector along UZ from the spacecraft to the surface. The bilinear interpolations giving 0 for the x- and y- coordinates are:

```
0 = X0 + X1*i +X2*j + X3*i*j
0 = Y0 + Y1*i +Y2*j + Y3*i*j
```

These are solved for the i,j corresponding to the piercing point. If Z(0) - Z(3) are the magnidudes to the corner vectors, the magnitude Zo is

```
Zo = Z0 + Z1*i +Z2*j + Z3*i*j
```

and the vector from the body center to the surface is V = UZ∗Zo - V0.

For reasons now lost in antiquity, a small parameter EPS has been included to allow the piercing point to spill slightly outsige the cell. This choice will be examined during testing.

The normal N at the landmark is estimated as the cross product of the diagonals of the cell.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

MM MMFL V0 CX CY CZ USE V N

File_I/O

Filename I/O Description

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_MODEL

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called UCRSS VEQU VHAT VSUB

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER BIGMAP BIGMAPL SUBROUTINE IMGPL2VN SUBROUTINE VISIBLE

Notes: 06/11/13: Added verification of existence of SHAPEFILES/SHAPE.TXT 09/17/13: W1 and W1 constratned to be between ∼0 and ∼1.

2.97 COMMON/MMPX.f File Reference

Functions/Subroutines

- subroutine mmpx (KMAT, CTR, D, MM, PX)

  Procedure.

2.97.1 Function/Subroutine Documentation

2.97.1.1 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

Procedure.

Parameters

kmat	Abstract

This subroutine computes the image-space (pixel/line) location of a point (X,Y) on the focal plane. In all current work, X=MM(1) and Y=MM(2). In some older cases (such as NEAR images of Eros) a distortion and focal length change was introduced, appearing in the .SUM files as the parameters D(1) - D(4). The origin (0,0) of the focal plane "millimeter space" maps into the central image space position CTR. The K-matrix translates X and Y into differential image space locations so that

```
p = CTR(1) + K(1,1)X + K(1,2)Y + K(1,3)XY
l = CTR(2) + K(2,1)X + K(2,2)Y + K(2,3)XY
```

K(1,3) and K(2,3) are nonzero for vidicon images such as Mariner, Viking and Voyager.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

KMAT CTR D MM PX

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS REGISTER RESIDUALS POLE SHIFT BIG↩ MAP BIGMAPL SUBROUTINE IPL2RDT SUBROUTINE IPL2SCOBJPTG SUBROUTINE IPL2VLM SUBROUTINE PXMM SUBROUTINE V2IMGPL

2.98 COMMON/NNEIGHBORS.f File Reference

Functions/Subroutines

- subroutine nneighbors (NTMP, I, J, TUSE, TMPL, ZUSE, ZHT, WT, HUSE, HT)

Procedure.

2.98.1 Function/Subroutine Documentation

2.98.1.1 subroutine nneighbors (integer∗4 *NTMP,* integer∗4 *I,* integer∗4 *J,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZHT,* real∗8 *WT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

Procedure.

Parameters

ntmp	Abstract

```
This subroutine carries out one iteration of equation 5 in tha reference
below.  The central point (o) below and each of ite nearest neighbors (x)
is characterized by a height and a template, where t(1)=-dh/dx and
t(2)=-dh/dy.

                      x
                    h3,t3               ----------> y
                                        |
               x       o       x        |
             h1,t1   h0,t0   h2,t2      |
                                        |
                      x                 v
                    h4,t4               x

The average y-slope between, say, points 0 and 2 is (-t0(2)-2(2))/2.
It is also (h2-h0), since the spacing between points is one unit.  We can
therefore write:
                h0 = h2 + (t0(2)+t2(2))/2.
Similarly,
                h0 = h1 - (t0(2)+t1(2))/2
                h0 = h4 + (t0(1)+t4(1))/2
                h0 = h3 - (t0(1)+t3(1))/2

At this stage of the iteration, we assume that all templates are known
as well as the neighboring heights, so h0 is given by the average of the
four values above.  In addition to this data, there is a constraining
height h!!at some randomly selected points, coming from limbs, overlapping
maplets or differential stereo.  This height is included in the average
with a weight wc.  Thus the new estimate for h0 is:

           h0 = (h1-(t0(2)+t1(2))/2+h2+(t0(2)+t2(2))/2
          +h3-(t0(1)+t3(1))/2+h4+(t0(1)+t4(1))/2+wc*hc)/(4+wc)

This is only one of many iterations involved in the relaxation process.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

i	Abstract

```
This subroutine carries out one iteration of equation 5 in tha reference
below.  The central point (o) below and each of ite nearest neighbors (x)
is characterized by a height and a template, where t(1)=-dh/dx and
t(2)=-dh/dy.

                      x
                    h3,t3               ----------> y
                                       |
               x       o       x       |
             h1,t1   h0,t0   h2,t2      |
                                       |
                      x                v
                    h4,t4              x

The average y-slope between, say, points 0 and 2 is (-t0(2)-2(2))/2.
It is also (h2-h0), since the spacing between points is one unit.  We can
therefore write:
                h0 = h2 + (t0(2)+t2(2))/2.
Similarly,
                h0 = h1 - (t0(2)+t1(2))/2
                h0 = h4 + (t0(1)+t4(1))/2
                h0 = h3 - (t0(1)+t3(1))/2

At this stage of the iteration, we assume that all templates are known
as well as the neighboring heights, so h0 is given by the average of the
four values above.  In addition to this data, there is a constraining
height h!!at some randomly selected points, coming from limbs, overlapping
maplets or differential stereo.  This height is included in the average
with a weight wc.  Thus the new estimate for h0 is:

           h0 = (h1-(t0(2)+t1(2))/2+h2+(t0(2)+t2(2))/2
          +h3-(t0(1)+t3(1))/2+h4+(t0(1)+t4(1))/2+wc*hc)/(4+wc)

This is only one of many iterations involved in the relaxation process.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

j	Abstract

```
This subroutine carries out one iteration of equation 5 in tha reference
below.  The central point (o) below and each of ite nearest neighbors (x)
is characterized by a height and a template, where t(1)=-dh/dx and
t(2)=-dh/dy.

                      x
                    h3,t3               ----------> y
                                       |
               x       o       x       |
             h1,t1   h0,t0   h2,t2      |
                                       |
                      x                v
                    h4,t4              x

The average y-slope between, say, points 0 and 2 is (-t0(2)-2(2))/2.
It is also (h2-h0), since the spacing between points is one unit.  We can
therefore write:
                h0 = h2 + (t0(2)+t2(2))/2.
Similarly,
                h0 = h1 - (t0(2)+t1(2))/2
                h0 = h4 + (t0(1)+t4(1))/2
                h0 = h3 - (t0(1)+t3(1))/2

At this stage of the iteration, we assume that all templates are known
as well as the neighboring heights, so h0 is given by the average of the
four values above.  In addition to this data, there is a constraining
height h!!at some randomly selected points, coming from limbs, overlapping
maplets or differential stereo.  This height is included in the average
with a weight wc.  Thus the new estimate for h0 is:

           h0 = (h1-(t0(2)+t1(2))/2+h2+(t0(2)+t2(2))/2
          +h3-(t0(1)+t3(1))/2+h4+(t0(1)+t4(1))/2+wc*hc)/(4+wc)

This is only one of many iterations involved in the relaxation process.
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP I J TUSE TMPL ZUSE ZHT WT HUSE HT

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS BIGMAP BIGMAPL SUBROUTINE SLP2HGT

2.99 COMMON/NORM2ALT.f File Reference

Functions/Subroutines

- subroutine norm2alt (Q, N, V0, N0, ZR, ZUSE, WT, A)

  Procedure.

2.99.1 Function/Subroutine Documentation

2.99.1.1 subroutine norm2alt (integer∗4 *Q,* double precision, dimension(3,0:512,0:512,6) *N,* double precision, dimension(3,0:512,0:512,6) *V0,* double precision, dimension(3,0:512,0:512,6) *N0,* double precision, dimension(0:512,0:512,6) *ZR,* logical, dimension(0:512,0:512,6) *ZUSE,* double precision *WT,* double precision, dimension(0:512,0:512,6) *A*)

Procedure.

Parameters

q	Abstract This subroutine is a wrapper for assigning random points to be iterated with the subroutine RELAXATION to iteratively determine a the altitude distribution of shape model points relative to a reference shape. The iteration is required to be consistent wth the maplet slopes (normals) and a sparse set of constraining altitudes. The subroutine carries out as many iterationa as there are surface points in the shape model, but must be repeated many times to ensure convergence.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

Q N V0 N0 ZR ZUSE WT A

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called RELAXATION

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs DENSIFY DEASIFYA

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.100 COMMON/ORIENT.f File Reference

Functions/Subroutines

- subroutine orient (UX, UY, UZ)

  Procedure.

2.100.1 Function/Subroutine Documentation

2.100.1.1 subroutine orient (double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

Procedure.

Parameters

ux	Abstract This routine will generate a set of x, y, z 3 dimensional unit vectors from the set of x, y, z three dimensional unit supplied by the calling program. The output values will over write the input values.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

UX I 3 dimensional x vector UY I 3 dimensional y vector UZ I 3 dimensional z vector UX O 3 dimensional x unit vector UY O 3 dimensional y unit vector UZ O 3 dimensional x unit vector

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called VHAT UCRSS

Called_by_SPC_Programs LITHOS BIGMAP BIGMAPL SUBROUTINE CREATE_LMFILE

2.101 COMMON/OVERLAPS.f File Reference

Functions/Subroutines

- subroutine overlaps (LMKNM)

  Procedure.

2.101.1 Function/Subroutine Documentation

2.101.1.1 subroutine overlaps (character∗6 *LMKNM*)

Procedure.

Abstract

This subroutine creates a file OVERLAPS.TXT that LITHOS uses to search for maplets that overlap a given one (LMKNM). It searches a list of maplets in an input list (INFILE). If a maplet has a resolution times lower than LMKNM it is excluded. If the maplet center differs in position by more than twice the sum of its half width and that of LMKNM, it is excluded as well. The remaining maplets are those over which the overlap searches will be made.

The input file is generally LMRKLIST.TXT, but this can be very large. In some circumstances, such as when we are dealing with maplets that are completely inside a BIGMAP, a file USED_MAPS.TXT generated by the BIGMAP program that has all maplets used to construct the BIGMAP can be copied into LMRKLISTO.TXT. This will be read as the input file for this subroutine, and it is guaranteed that only maplets in that file should be in OVERLAPS.TXT.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMKNM I Landmark to process for overlaps.

File_I/O

Filename I/O Description

LMKFILES/<LMKNM>.LMK I Landmark file for <LMKNM> LMRKLISTO.TXT OR LMRKLIST.TXT I File containing list of all landmarks. OVERLAPS.TXT O Landmark overlap text file that gets updated. MAPFILES/<LMKNM>.M↩ AP I Maplet file associated with landmark <LMKNM>

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called READ_HEADER

SPICELIB_functions_called None

SPICELIB_subroutines_called VNORM VSUB

Called_by_SPC_Programs LITHOS

Notes: 6/13/13: Verified files exist prior to opening. 3/24/14: LMRKLISTX.TXT used as list if LMKLX=.TRUE. IN INIT_LITHOS.TXT.

2.102 COMMON/PATCH_COORDS.f File Reference

Functions/Subroutines

- subroutine patch_coords (NTMP, QSZ, HUSE, HT, UX, UY, UZ)

  Procedure.

2.102.1 Function/Subroutine Documentation

2.102.1.1 subroutine patch_coords (integer∗4 *NTMP,* integer∗4 *QSZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗8, dimension(3) *UX,* real∗8, dimension(3) *UY,* real∗8, dimension(3) *UZ*)

Procedure.

Parameters

ntmp	Abstract This subroutine fits a plane to the heights in a maplet in order to refine the maplet's reference plane. It returns new values for the unit vectors UX, UY and UZ and the new heights HT(i,j) relative to the new reference plane.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract This subroutine fits a plane to the heights in a maplet in order to refine the maplet's reference plane. It returns new values for the unit vectors UX, UY and UZ and the new heights HT(i,j) relative to the new reference plane.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ HUSE HT UX UY UZ

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called INVERTN

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called UCRSS VEQU VHAT

Called_by_SPC_Programs LITHOS BIGMAP BIGMAPL SUBROUTINE CREATE_LMFILE

2.103 COMMON/PICINPT.f File Reference

Functions/Subroutines

- subroutine picinpt (PICNM, Z0, KK, DN, ESZ)

  Procedure.

2.103.1 Function/Subroutine Documentation

2.103.1.1 subroutine picinpt (character∗12 *PICNM,* integer, dimension(2) *Z0,* integer *KK,* integer, dimension(-1024:1024,-1024:1024) *DN,* integer *ESZ*)

Procedure.

Parameters

dn	Abstract

This subroutine extracts data from the 8 or 16 bit image PICNM centered on image pixel [Z0(1),Z0(2)] and spaced by KK pixels and lines to fill an integer array i,j = (-ESZ,ESZ) centered on (0,0) and nested in integer array DN(-1024:1024,-1024:1024) and centered at (0,0).

This subroutine is called by LITHOS with ESZ=128, enabling many images to be loaded without swapping. If an image has a much higher resolution than the maplet, a value KK>1 is chosen to sample the image data at lower resolution.

This subroutine is called by REGISTER and AUTOREGISTER with ESZ=1024, enabling use of images up to 2049x2049. If larger images are used, a value of KK>1 is used to sample image data at lower resolution.

The program detects whether PICNM contains 8 or 16 bit data and, if 16 bits, whether the data is LSB or MSB.

If a template file is available to remove shadows and global blemishes, the program adjusts the extracted data accordingly.

Disclaimer

Abondon hope, ye who enter here.

Required_Reading

NONE

Declarations Variable_I/O

Variable I/O Description

PICNM I IMAGE NAME Z0 I CENTRAL PIXEL/LINE IN IMAGE OF DATA EXTRACTED KK I PIXEL/LINE SPAC↩ ING OF EXTRACTED DATA ESZ I HALF-SIZE OF TARGET ARRAY NPX I NUMBER OF PIXELS IN ORIGINAL IMAGE (FROM SUMFILE) NLN I NUMBER OF LINES IN ORIGINAL IMAGE (FROM SUMFILE) T2 I MAXIMUM VALID BRIGHTNESS VALUE (FROM SUMFILE) DN O OUTPUT ARRAY OF EXTRACTED DATA

File_I/O

Filename I/O Description

PICNM.SUM I IMAGE GEOMETRY SUMMARY FILE PICNM.DAT I RAW IMAGE DATA (8 OR 16 BIT) PICNM.↩ RAW I OPTIONAL TEMPLATE FILE

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

Version 0.01

SPC_functions_called

SLEN

SPC_subroutines_called

FIND_PICFILE

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP REGISITER SUBROUTINE LIMB_HEIGHTS

2.104 COMMON/PICINPT_LO.f File Reference

Functions/Subroutines

- subroutine picinpt_lo (PICNM, Z0, KK, DN)

  Procedure.

2.104.1 Function/Subroutine Documentation

2.104.1.1 subroutine picinpt_lo (character∗12 *PICNM,* integer, dimension(2) *Z0,* integer *KK,* integer, dimension(-128:128,-128:128) *DN*)

Procedure.

Abstract

This subroutine extracts data from the 8 or 16 bit image PICNM centered on image pixel [Z0(1),Z0(2)] and spaced by KK pixels and lines to fill an integer array DN(-128:128,-128:128).

This subroutine is called by LITHOSP, enabling many images to be loaded without swapping. If an image has a much higher resolution than the maplet, a value KK>1 is chosen to sample the image data at lower resolution.

The program detects whether PICNM contains 8 or 16 bit data and, if 16 bits, whether the data is LSB or MSB.

If a template file is available to remove shadows and global blemishes, the program adjusts the extracted data accordingly.

This subroutine employs several lockout files to guarantee that the files associated with PICNM cannot be opened simultaneously with another process that might be attampting to write to it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I IMAGE NAME Z0 I CENTRAL PIXEL/LINE IN IMAGE OF DATA EXTRACTED KK I PIXEL/LINE SPAC↩ ING OF EXTRACTED DATA ESZ I HALF-SIZE OF TARGET ARRAY NPX I NUMBER OF PIXELS IN ORIGINAL IMAGE (FROM SUMFILE) NLN I NUMBER OF LINES IN ORIGINAL IMAGE (FROM SUMFILE) T2 I MAXIMUM VALID BRIGHTNESS VALUE (FROM SUMFILE) DN O OUTPUT ARRAY OF EXTRACTED DATA

File_I/O

Filename I/O Description

./TESTFILES/PICNM.SUM O Lockout file for PICNM.SUM ./SUMFILES/PICNM.SUM I IMAGE GEOMETRY S↩ UMMARY FILE ./TESTFILES/PICNM.DAT O Lockout file for PICNM.DAT ./IMAGES/PICNM.DAT I RAW IMAGE DATA (8 OR 16 BIT) ./TESTFILES/PICNM.RAW O Lockout file for PICNM.RAW ./TEMPLATES/PICNM.RAW I OPTIONAL TEMPLATE FILE

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY FIND_PICFILE

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_08: Capitalization, compiler warnings fixed and header added.

2.105 COMMON/PICRES.f File Reference

Functions/Subroutines

- double precision function picres (PICNM, LMKNM)

  Procedure.

2.105.1 Function/Subroutine Documentation

2.105.1.1 double precision function picres (character*12 *PICNM,* character*6 *LMKNM*)

Procedure.

Returns

Abstract

This subroutine conmputes the resolution (km/pixel) of an image at the location of a maplet in the image. A 1x1 maplet pixel square has sides projected on the image of (dp/dx,dl/dx) and (dp/dy,dl/dy). The area of ths cell of the image in square pixels is the cross product of these two vectors:

A = (dp/dx,dl/dx) x (dp/dy,dl/dy) = (dp/dx)(dl/dy) - (dp/dy)(dl/dx)

since the area of the maplet square is scale^2, the image resolution in km/px is scale/sqrt(A).

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Image name to calculate the resolution LMKNM I Landmark name in which the image is use.

File_I/O

Filename I/O Description

LMKFILES/<LMKNM>.LMK I Landmark file for <LMKNM> to provide the SCALE, unit vectors and S/!!to object position.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called LOC2PIC

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOS

Notes: 6/13/13: Add verification of LMRKFILE existence prior to opening Function returns -1 in this case.

2.106 COMMON/PICRES_LO.f File Reference

Functions/Subroutines

- double precision function picres_lo (PICNM, LMKNM)

  Procedure.

2.106.1 Function/Subroutine Documentation

2.106.1.1 double precision function picres_lo (character∗12 *PICNM,* character∗6 *LMKNM*)

Procedure.

Returns

Abstract

This subroutine conmputes the resolution (km/pixel) of an image at the location of a maplet in the image. A 1x1 maplet pixel square has sides projected on the image of (dp/dx,dl/dx) and (dp/dy,dl/dy). The area of ths cell of the image in square pixels is the cross product of these two vectors:

A = (dp/dx,dl/dx) x (dp/dy,dl/dy) = (dp/dx)(dl/dy) - (dp/dy)(dl/dx)

since the area of the maplet square is scale^2, the image resolution in km/px is scale/sqrt(A).

This subroutine employs a lockout file to guarantee that the file LMKNM.LMK cannot be opened simultaneously with another process that might be attampting to write to it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM I Image name to calculate the resolution LMKNM I Landmark name in which the image is use.

File_I/O

Filename I/O Description

./LMKFILES/<LMKNM>.LMK I Landmark file for <LMKNM> to provide the SCALE, unit vectors and S/!!to object position. ./TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>. Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY LOC2PIX

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LITHOSP

History 2013_10_08: Capitalization, compiler warnings fixed and header added.

2.107 COMMON/POINT2MAPS.f File Reference

Functions/Subroutines

- subroutine point2maps ()

  Procedure.

2.107.1 Function/Subroutine Documentation

2.107.1.1 subroutine point2maps ( )

Procedure.

Abstract

This subroutine determines which maplets and maps include a given surface point. A surface point can be specified by its image-space location in an image, by its latitude and longitude or by its coordinates in a map or maplet.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

None

File_I/O

Filename I/O Description

BIGLIST.TXT I File listing the big maps. LMRKLIST.TXT I File listing the landmark maps LMRKLISTO.TXT I File listing nearby landmark maps

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called IMGPL2VN READ_HEADER U2VN

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called LATREC RECLAT VSUB

CALLED_BY_SPC_PROGRAMS LITHOS

History 2014_02_24: Reads LMRKLISTO.TXT if available.

2.108 COMMON/PREDICT_DATA.f File Reference

Functions/Subroutines

- subroutine predict_data (NTMP, QSZ, PMX, NPIX, LAMBDA, PHI, TUSE, TMPL, CP, SP, NSZ, DN1)

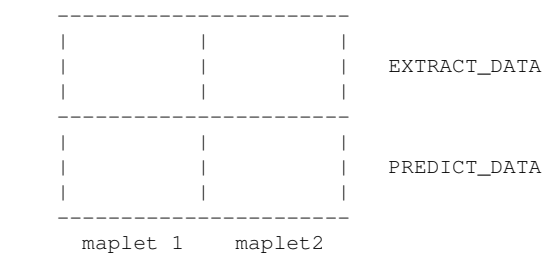
  Procedure.

2.108.1 Function/Subroutine Documentation

2.108.1.1 subroutine predict_data (integer∗4 *NTMP,* integer∗4 *QSZ,* integer∗4 *PMX,* integer∗4 *NPIX,* real∗8, dimension(pmx) *LAMBDA,* real∗8, dimension(pmx) *PHI,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* real∗8, dimension(3,pmx) *CP,* real∗8, dimension(3,pmx) *SP,* integer∗4 *NSZ,* real∗4, dimension(5000,5000) *DN1*)

Procedure.

Parameters

ntmp	Abstract This subroutine is a wrapper that takes the images from subroutine PREDICT_DATA_PI!!and puts the images into an array for comparison with with real orthorectified imaging data supplied by the EXTRACT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

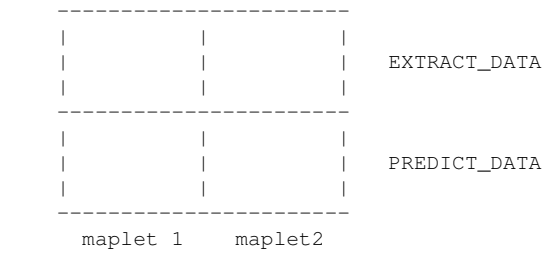
Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract This subroutine is a wrapper that takes the images from subroutine PREDICT_DATA_PI!!and puts the images into an array for comparison with with real orthorectified imaging data supplied by the EXTRACT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

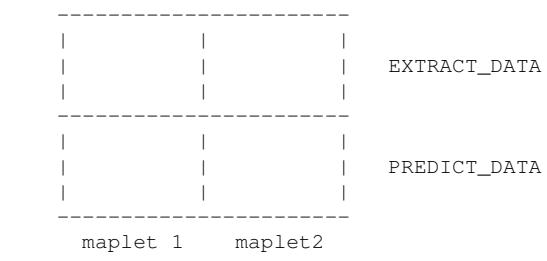
Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

pmx	Abstract This subroutine is a wrapper that takes the images from subroutine PREDICT_DATA_PI!!and puts the images into an array for comparison with with real orthorectified imaging data supplied by the EXTRACT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

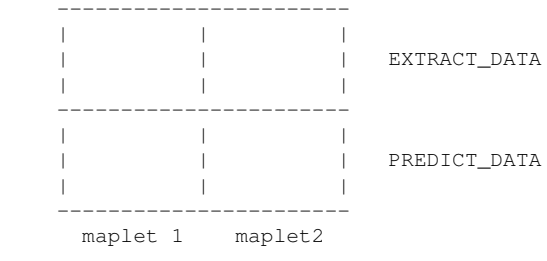
Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

npix	Abstract This subroutine is a wrapper that takes the images from subroutine PREDICT_DATA_PI!!and puts the images into an array for comparison with with real orthorectified imaging data supplied by the EXTRACT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

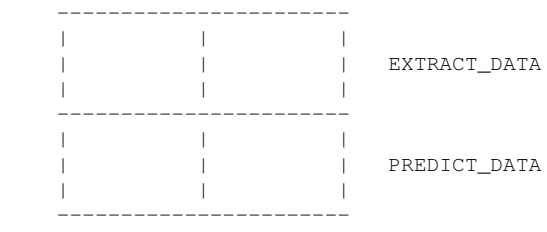
Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

nsz	Abstract This subroutine is a wrapper that takes the images from subroutine PREDICT_DATA_PI!!and puts the images into an array for comparison with with real orthorectified imaging data supplied by the EXTRACT_DATA subroutine. The array is as square as possible. A 200x200 pixel array can have two 100x100 maplets (actually 99x99 with a one picel separation) with the maplet images  The maximum number of maplets in a 400x400 pixel array is 8, 18 in a 600x600, etc.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ PMX NPIX LAMBDA PHI TUSE TMPL CP SP NSZ DN1

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called PREDICT_DATA_PIC

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS AUTOREGISTERP

2.109 COMMON/PREDICT_DATA_PIC.f File Reference

Functions/Subroutines

- subroutine predict_data_pic (NTMP, QSZ, LAMBDA, PHI, TUSE, TMPL, CP, SP, DNX)

  Procedure.

2.109.1 Function/Subroutine Documentation

2.109.1.1 subroutine predict_data_pic (integer∗4 *NTMP,* integer∗4 *QSZ,* real∗8 *LAMBDA,* real∗8 *PHI,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* real∗8, dimension(3) *CP,* real∗8, dimension(3) *SP,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *DNX*)

Procedure.

Parameters

ntmp	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ LAMBDA PHI TUSE TMPL CP SP DNX

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called ILLUM

SPC_subroutines_called None

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOSP REGISTER SUBROUTINE PREDICT_DATA_PIC

2.110 COMMON/PXCOUNT.f File Reference

Functions/Subroutines

- integer function pxcount (LMKNM, K)

  Procedure.

2.110.1 Function/Subroutine Documentation

2.110.1.1 integer function pxcount (character∗6 *LMKNM,* integer *K*)

Procedure.

Returns

Abstract

```
This function will count the number of pictures, overlaps, or limbs
for the landmark <LNKNM> depending on input variable K.

  If K=0, counts pictures containing a landmark
  If K=1, counts overlaps for a landmark
  If K=2, counts limb appearances for a landmark
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Name of the landmark to count images, landmarks, or overlaps K I Control variable to tell software rather to count images, landmarks, overlaps. PXCOUNT O Counts calculated

File_I/O

Filename I/O Description

LMKFILES/<LMKNM>.LMK I Landmark file used to calculate images, landmarks, or overlaps

Restrictions LMKFILES/<LMKNM>.LMK must exist.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS RESIDUALS SUBROUTINE IPL2VLM

Notes: 6/13/13: Added verification of LMRKFILE existence Function returns -1 if this condition exists.

2.111 COMMON/PXCOUNT_LO.f File Reference

Functions/Subroutines

- integer function pxcount_lo (LMKNM, K)

  Procedure.

2.111.1 Function/Subroutine Documentation

2.111.1.1 integer function pxcount_lo (character∗6 *LMKNM,* integer *K*)

Procedure.

Returns

Abstract

```
This function will count the number of pictures, overlaps, or limbs
for the landmark <LNKNM> depending on input variable K.

  If K=0, counts pictures containing a landmark
  If K=1, counts overlaps for a landmark
  If K=2, counts limb appearances for a landmark

The functions opens a lockout file before opening LMKNM.LMK in case
another process might want to write to LMKNM.LMK at the same time.  If
the lockout file already exists, the function will wait before proceeding.
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMKNM I Name of the landmark to count images, landmarks, or overlaps K I Control variable to tell software rather to count images, landmarks, overlaps. PXCOUNT O Counts calculated

File_I/O

Filename I/O Description

./TESTFILES/<LMKNM>.LMK O Lockout file for <LMKNM>.LMK ./LMKFILES/<LMKNM>.LMK I Landmark file used to calculate images, landmarks, or overlaps

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called DELAY

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LITHOSP SUBROUTINE IPL2VLMP

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.112 COMMON/PXMM.f File Reference

Functions/Subroutines

- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)

  Procedure.

2.112.1 Function/Subroutine Documentation

2.112.1.1 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

Procedure.

Parameters

kmat	Abstract

This subroutine is not the inverse of MMPX. It transforms the image-space (pixel/line) locationn of a point directly into the gnomoni!!projection, reversing the transformation GETMM(PICNM,MMFL,Z1,Z2,Z3,Z) followed by MMP↩ X(KMAT,CTR,D,Z,PX0). Due to the complexity of these transformations, the subroutine is iterative, predicting and correcting the solution.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

PICNM MMFL KMAT CTR D PX MM

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GETMM MMPX

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER GEOMETRY LITHOS REGISTER RESIDUALS POLE BIGMAP BI↩ GMAPL SUBROUTINE IMGPL2VN SUBROUTINE VISIBLE

2.113 COMMON/RANN.f File Reference

Functions/Subroutines

- real ∗8 function rann (SEED)
  Procedure.

2.113.1 Function/Subroutine Documentation

2.113.1.1 real∗8 function rann (real∗8 *SEED*)

Procedure.

Abstract This function will generate a random number using the provided seed <SEED> from the calling routine. It is a wrapper that impliments the intrinsi!!RAND routine. SEED should be a large integer or floating point number. It is rounded to an integer ISEED.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

SEED I Value of the seed to use to generate the random number seed ISEED used by SRAND RANN O Random number generated

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called PI

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTERP LIMBER LITHOS LITHOSP BIGMAP BIGMAPL DENSIFY DENS↩ IFYA SPHEREMAPSA SPHEREMAPSB SUBROUTINE MAXLEN SUBROUTINE SLP2HGT SUBROUTINE ST↩ EREO

2.114 COMMON/RAW2PGM.f File Reference

Functions/Subroutines

- subroutine raw2pgm (INFILE, OUTFILE, NPX, NLN)

  Procedure.

2.114.1 Function/Subroutine Documentation

2.114.1.1 subroutine raw2pgm (character∗72 *INFILE,* character∗72 *OUTFILE,* integer∗4 *NPX,* integer∗4 *NLN*)

Procedure.

Parameters

npx	Abstract This subroutine adds a .pgm header to a raw 8 bit gray scale image to produce an image file readable by various applications such as xv, GraphicConverter, ImageMagick, etc. The added header looks like: P5 <- indicates portable gray map (pgm) file #. <- commented out line 1024 1024 <- Number of pixels and number of lines in image (NPX, NLN) 255 <- Maximum data number

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

nln	Abstract This subroutine adds a .pgm header to a raw 8 bit gray scale image to produce an image file readable by various applications such as xv, GraphicConverter, ImageMagick, etc. The added header looks like: P5 <- indicates portable gray map (pgm) file #. <- commented out line 1024 1024 <- Number of pixels and number of lines in image (NPX, NLN) 255 <- Maximum data number

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

INFILE I Input gray scale image file OUTFILE O Output pgm format image file NPX I Number of pixels in image NLN I Number of lines in image

File_I/O

Filename I/O Description

INFILE I Input gray scale file provided calling routine OUTFILE O Gray scale file converted to pgm format.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER BLEMISHES BIGMAP BIGMAPL SUBROUTI↩ NE ATTACH SUBROUTINE COMPARE SUBROUTINE DISPLAY1 SUBROUTINE SHOW_SLOPES SUBROUT↩ INE ZOOM

History 2014_05_04: Algorithm replaced by one from spheremapsB

2.115 COMMON/RAW2PPM.f File Reference

Functions/Subroutines

- subroutine raw2ppm (INFILE, OUTFILE, NPX, NLN)

  Procedure.

2.115.1 Function/Subroutine Documentation

2.115.1.1 subroutine raw2ppm (character∗72 *INFILE,* character∗72 *OUTFILE,* integer∗4 *NPX,* integer∗4 *NLN*)

Procedure.

Parameters

npx	Abstract This subroutine adds a .ppm header to a raw 24 bit color (rgb) image to produce an image file readable by various applications such as xv, GraphicConverter, ImageMagick, etc. The added header looks like: P6 <- indicates portable pixel map (ppm) file #. <- commented out line 2881 1441 <- Number of pixels and number of lines in image (NPX, NLN) 255 <- Maximum data number per channel

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

nln	Abstract
	
	`This subroutine adds a .ppm header to a raw 24 bit color (rgb) image`
	`to produce an image file readable by various applications such as xv,`
	`GraphicConverter, ImageMagick, etc. The added header looks like:`
	
	`P6 <- indicates portable pixel map (ppm) file`
	`#. <- commented out line`
	`2881 1441 <- Number of pixels and number of lines in image (NPX, NLN)`
	`255 <- Maximum data number per channel`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

INFILE I Input gray scale image file OUTFILE O Output ppm format image file NPX I Number of pixels in image NLN I Number of lines in image Var name O Description of output variables

File_I/O

Filename I/O Description

INFILE I Input gray scale file provided calling routine OUTFILE O Gray scale file converted to ppm format.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS REGISTER

History 2014_05_04: Algorithm replaced by one from spheremapsB

2.116 COMMON/READ_HEADER.f File Reference

Functions/Subroutines

- subroutine read_header (LMRKFILE, QSZ, SCALE, V, UX, UY, UZ)

  Procedure.

2.116.1 Function/Subroutine Documentation

2.116.1.1 subroutine read_header (character∗72 *LMRKFILE,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

Procedure.

Parameters

qsz	Abstract This routine will read the pixel size of the map, scale of the map, body-fixed vector from the center of the body to the center of the map and the uncertainty of that vector, and the unit X,Y,Z vector of the center of the map. It allows the software to see whether it is necessary to read the complete mapfile.

The mapfile header has 72 bytes records containing information describing the size, scale, orientation and position of the map:

bytes 1-6 Unused bytes 7-10 Scale in km/pixel (real∗4 msb) bytes 11-12 qsz where map is 2∗qsz+1 x 2∗qsz+1 pixels (unsigned short lsb) bytes 16-27 map center body fixed position vector in km 3 x (real∗4 msb) bytes 28-39 Ux body fixed unit map axis vector 3 x (real∗4 msb) bytes 40-51 Uy body fixed unit map axis vector 3 x (real∗4 msb) bytes 52-63 Uz body fixed unit map normal vector 3 x (real∗4 msb) bytes 64-67 Hscale = maximum abs(height)/30000 (real∗4 msb) ∗ byte 13 255∗ X position uncertainty unit vector component (byte) + byte 14 255∗ Y position uncertainty unit vector component (byte) + byte 15 255∗ Z position uncertainty unit vector component (byte) + bytes 68-71 magnitude of position uncertainty (real∗4 msb) + byte 72 Unused

Disclaimer Nonbe

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O Variable I/O Description

LMRKFILE I Map file to read header from QSZ O Pixel size of map file SCALE O Scale in KM of map V O Vector from body center to landmark center UX O X unit vector of landmark center UY O Y unit vector of landmark center UZ O Z unit vector of landmark center

File_I/O

Filename I/O Description

<LMRKFILE> I Mapfile to read header info

Restrictions MAP file must exist

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called FLIP

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS REGISTER BIGMAP BIGMAPL SPH↩ EREMAPSB SUBROUTINE LIMB_HEIGHTS SUBROUTINE OVERLAPS SUBROUTINE POINTS2MAPS See if architecture is big or little endian

2.117 COMMON/READ_MAP.f File Reference

Functions/Subroutines

- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)

  Procedure.

2.117.1 Function/Subroutine Documentation

2.117.1.1 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

Procedure.

Parameters

ntmp	Abstract This routine will read the header information: Map size = (2∗QSZ+1) x (2∗QSZ+1) pixels Map scale (SCALE in KM) Body fixed vector from body center to landmark center (V) Landmark reference plane orientation unit vectors in body fixed coordinates system UX,U↩ Y,UZ And the height and albedo information for each point on the map from file <LMRKFI↩ LE>.

The mapfile is made up of 72 byte records. The first record contains information describing the size, scale, orientation and position of the map:

bytes 1-6 Unused bytes 7-10 Scale in km/pixel (real∗4 msb) bytes 11-12 qsz where map is 2∗qsz+1 x 2∗qsz+1 pixels (unsigned short lsb) bytes 16-27 map center body fixed position vector in km 3 x (real∗4 msb) bytes 28-39 Ux body fixed unit map axis vector 3 x (real∗4 msb) bytes 40-51 Uy body fixed unit map axis vector 3 x (real∗4 msb) bytes 52-63 Uz body fixed unit map normal vector 3 x (real∗4 msb) bytes 64-67 Hscale = maximum abs(height)/30000 (real∗4 msb) ∗ byte 13 255∗ X position uncertainty unit vector component (byte) + byte 14 255∗ Y position uncertainty unit vector component (byte) + byte 15 255∗ Z position uncertainty unit vector component (byte) + bytes 68-71 magnitude of position uncertainty (real∗4 msb) + byte 72 Unused

- heights are in units of map scale
- these are pretty much unused as far as I can see.

The remaining records are made up of 3 byte chunks:

bytes 1-2 height/hscale (integer∗2 msb) byte 3 relative "albedo" (1-199) (byte)

If there is missing data at any point, both height and albedo are set to zero.

The map array is read row by row from the upper left (i,j = -qsz). Rows are increasing in the Uy direction with spacing = scale Columns are increasing in the Ux direction with spacing = scale Heights are positive in the Uz direction with units = scale

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

LMRKFILE I Name of map file to read NTMP O Array dimension size for height & albedo arrays QSZ O Map size SCALE O Map scale in KM V O Vector from center of body to landmark center UX O X unit vector of landmark center UY O Y unit vector of landmark center UZ O Z unit vector of landmark center HT O Height for each pixel ALB O Albedo for each pixel

File_I/O

Filename I/O Description

<LMRKFILE> I Map file to read

Restrictions Map file must exists

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called FLIP

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP LITHOS LITHOSP BIGMAP DENSIFY DENSIFYA SPHEREMAPSA SPHEREMAPSB SUBROUTINE ATTACH SUBROUTINE CREATE_LMFILE SUBROUTINE GET_MAP SUBROUTINE IPL2VLM SUBROUTINE POINT2MAP SUBROUTINE VISIBLE

2.118 COMMON/RELAXATION.f File Reference

Functions/Subroutines

- subroutine relaxation (Q, I, J, F, N, V0, N0, ZR, ZUSE, WT, A)

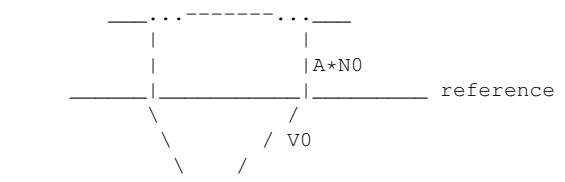
  Procedure.

2.118.1 Function/Subroutine Documentation

2.118.1.1 subroutine relaxation (integer∗4 *Q,* integer∗4 *I,* integer∗4 *J,* integer∗4 *F,* double precision, dimension(3,0:512,0:512,6) *N,* double precision, dimension(3,0:512,0:512,6) *V0,* double precision, dimension(3,0:512,0:512,6) *N0,* double precision, dimension(0:512,0:512,6) *ZR,* logical, dimension(0:512,0:512,6) *ZUSE,* double precision *WT,* double precision, dimension(0:512,0:512,6) *A*)

Procedure.

Parameters

q	Abstract The program densify first constructs a reference surface by interpolating the surface points of a lower resolution shape model. At each point of the reference there is a vector V0 from the model center to that point and a normal N0 to the surface. That normal is extended some distance until it pierces one or more of the ensemble of maplets, and the average A of those distances is taken to represent the piercing point on the new model's surface, so the new surface vector is V = V0 + A*N0. Because,  especially at the early stages, there are mismatches in maplet locations simply due to the formal uncertainties of the estimation process, we have found it better to average the maplet normals N at each point, keeping a small randomly selected set of the A as conditioning heights. If Wa is the average maplet normal between two neighboring points a and o, then (Va - Vo).Wa = 0 = (Va0+Aa*Na0).Wa - (Vo0+Ao*No0).Wa so, as we did in the slope to height determination in subroutines SLP2HGT and NNEIGHBORS, we determine Ao from the average of Ao = (Va0.Wa-Vo0.Wa+Aa*Na0.Wa)/NoO.Wa over the four neighboring points (a,b,d,d), including a random constraining altitude Ax with a weight wx: Ao = [(Va0-Vo0+Aa*Na0).Wa/No0.Wa +(Vb0-Vo0+Ab*Nb0).Wb/No0.Wb +(Vc0-Vo0+Ac*Nc0)/No0.W!!+(Vd0-Vo0+Ab*Nb0).Wd/NoO.Wd + wx*Ax]/(4+wx) This is only one of many iterations involved in the relaxation process.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

Q I J F N V0 N0 ZR ZUSE WT A

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called CORNERS MATCHUP

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs SUBROUTINE NORM2ALT

History 2013_10_18: Capitalization, compiler warnings fixed and header added.

2.119 COMMON/REPLICATE_LMFILE.f File Reference

Functions/Subroutines

- subroutine replicate_lmfile (LMRKOLD, LMRKNEW)

  Procedure.

2.119.1 Function/Subroutine Documentation

2.119.1.1 subroutine replicate_lmfile (character*6 *LMRKOLD,* character*6 *LMRKNEW*)

Procedure.

Abstract This subroutine creates a new landmark file LMKFILES/<LMRKNEW>.LMK and a new landmark map file MAPFILES/<LMRKNEW>.MAP and writes the data from LMKFILES/<LMRKOLD>.LMK & MAPFILES/<LMR↩KOLD>.MAP to them. The subroutine also updates the LMRKLIST.TXT, LMRKLIST1.TXT, LMRKLISTO.TXT with the new landmark name <LMRKNEW>.

This subroutine is used in LITHOS to rename landmarks and to create dublicate landmarks that can be moved to a neighboring position and/or be changes in size, scale or orientation.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMRKOLD I Old landmark name LMRKNEW I New landmark name

File_I/O

Filename I/O Description

LMRKLIST.TXT O Updated with new landmark name LMRKLIST0.TXT O Updated with new landmark name LM↩RKLIST1.TXT O Updated with new landmark name LMKFILES/<LMRKOLD>.LMK I Old landmark file LMKFIL↩ES/<LMRKNEW>.LMK O New landmark file MAPFILES/<LMRKOLD>.MAP I Old landmark map file MAPFILE↩S/<LMRKNEW>.MAP O New landmark map file

Restrictions LMRKLIST.TXT must exist.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called INSERTINLMK_LMK INSERTINPIC_LIM INSERTINPIC_LMK

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

Notes: Added check for existence of LMRKLIST.TXT file

2.120 COMMON/SHIFT_MODEL.f File Reference

Functions/Subroutines

- subroutine shift_model (INFILE, W)

  Procedure.

2.120.1 Function/Subroutine Documentation

2.120.1.1 subroutine shift_model (character∗72 *INFILE,* real∗8, dimension(3) *W*)

Procedure.

Abstract This subroutine will shift the shape model vectors in the shape model file <INFILE> by the values in the 3D vector W. The subroutine is expecting the data to be in the format 3F12.5 or 4F12.5 accordint to whether albedo is included, and produces this format. The program DENSIFY produces files of this form for most objects of interest, but for the Earth and any object with diameter greater than 9000 km it produces a 3F15.5 format and the user should be aware of this.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

INFILE I Name of shape model to be shifted W I 3D vector containing shift values

File_I/O

Filename I/O Description

<INFILE> I Shape model provided by program SHIFT

Restrictions Shape model file must exist.

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs SHIFT

Notes: 6/14/2013: Added checks to verify existence of shape model file

2.121 COMMON/SHOW_SLOPES.f File Reference

Functions/Subroutines

- subroutine show_slopes (NTMP, QSZ, HUSE, HT)

  Procedure.

2.121.1 Function/Subroutine Documentation

2.121.1.1 subroutine show_slopes (integer∗4 *NTMP,* integer∗4 *QSZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

Procedure.

Parameters

ntmp	Abstract `This subroutine creates an image file slope.pgm that displays the two components of slope of a BIGMAP or maplet. The slopes are computed by the HGT2SLP. The image is similar to two views of the surface, one illuminated from the south and the other from the east.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract `This subroutine creates an image file slope.pgm that displays the two components of slope of a BIGMAP or maplet. The slopes are computed by the HGT2SLP. The image is similar to two views of the surface, one illuminated from the south and the other from the east.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ HUSE HT

File_I/O

Filename I/O Description

INFILE O slope.gray - gray scale file OUTFILE I slope.pgm - pgm format file converted from slope.gray file

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called HGT2SLP RAW2PGM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS BIGMAP

2.122 COMMON/SLEN.f File Reference

Functions/Subroutines

- integer ∗4 function slen (STRING)

  Procedure.

2.122.1 Function/Subroutine Documentation

2.122.1.1 integer∗4 function slen (character∗(∗) *STRING*)

Procedure.

Returns

Abstract This function calculates the length of a character string not including trailing white spaces and returns that value.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

STRING I Character string to calculate the length of. SLEN O Length of the character string <STRING>

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs AUTOREGISTER AUTOREGISTERP BLEMISHES LIMBER LITHOS LITHOSP POLE REGISTER SHIFT BIGMAPS SPHEREMAPSA SPHEREMAPSB SUBROUTINE CREATE_LMFILE SUBROUTINE DEBLEMISH SUBROUTINE DELETEINNOM_TRJ SUBROUTINE DELETEINPIC_LIM SUBROUTINE DELETEINPIC_LMK SUBROUTINE DELETE_LMFILE SUBROUTINE DENOISE SUBROUTINE EXTRACT_DATA_PIC SUBROUTINE FIND_PICFILE SUBROUTINE IMGPL2VN SUBROUTINE INSERTINNOM_FRAME SUBROUTINE INSERTINNOM_PSIG SUBROUTINE INSERTINNOM_PTG SUBROUTINE INSERTINNOM_SOV SUBROUTINE INSERTINNOM_TRJ SUBROUTINE INSERTINNOM_VSIG SUBROUTINE INSERTINPIC_LIM SUBROUTINE INSERTINPIC_LMK SUBROUTINE INSERTINPIC_POLE SUBROUTINE INSERTINPIC_PTG SUBROUTINE INSERTINPIC_SIG SUBROUTINE INSERTINPIC_SOV SUBROUTINE INSERTINPIC_SZ SUBROUTINE INSERTINPIC_VFLG SUBROUTINE IPL2RDT SUBROUTINE IPL2SCOBJPTG SUBROUTINE IPL2VLM SUBROUTINE LIMB_HEIGHTS SUBROUTINE LMCOUNT SUBROUTINE LOC2PIX SUBROUTINE PICINPT SUBROUTINE SOLVE_RDT SUBROUTINE STEREO SUBROUTINE TUCK SUBROUTINE VISIBLE

2.123 COMMON/SLP2HGT.f File Reference

Functions/Subroutines

- subroutine slp2hgt (NTMP, QSZ, TUSE, TMPL, ZUSE, ZHT, WT, HUSE, HT)

  Procedure.

2.123.1 Function/Subroutine Documentation

2.123.1.1 subroutine slp2hgt (integer∗4 *NTMP,* integer∗4 *QSZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *TUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZHT,* real∗8 *WT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT*)

Procedure.

Parameters

ntmp	Abstract `This subroutine is a wrapper for assigning random points to be iterated with the subroutine NNEIGHBORS to iteratively determine a maplet height distribution consistent wth the measured slopes and a sparse set of constraining heights. The subroutine carries out as many iterationa as there are pixels in the map or maplet, but must be repeated many times to ensure convergence.`

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Parameters

qsz	Abstract This subroutine is a wrapper for assigning random points to be iterated with the subroutine NNEIGHBORS to iteratively determine a maplet height distribution consistent wth the measured slopes and a sparse set of constraining heights. The subroutine carries out as many iterationa as there are pixels in the map or maplet, but must be repeated many times to ensure convergence.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP QSZ TUSE TMPL ZUSE ZHT WT HUSE HT

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN

SPC_subroutines_called NNEIGHBORS

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS BIGMAP BIGMAPL

2.124 COMMON/SOLVE_RDT.f File Reference

Functions/Subroutines

- subroutine solve_rdt

  Procedure.

2.124.1 Function/Subroutine Documentation

2.124.1.1 subroutine solve_rdt ( )

Procedure.

Abstract

This subroutine is a wrapper for calling the IPL2RDT subroutine in the context of the GEOMETRY program. Its use there is still non-functional but is a place holder in case we need to deal with wobble. The precise procedures are still under study.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

None

File_I/O

Filename I/O Description

POLELIST.TXT I 1st choice for list of images PICTLISTS.TXT I 2nd choice for list of images PICTLISTR.TXT I 3rd choice for list of images PICTLIST.TXT I Last choice for list of images

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called IPL2RDT

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs GEOMETRY POLE

Notes: 6/14/2013: Added check for the existence for PICTLIST.TXT

2.125 COMMON/STEREO.f File Reference

Functions/Subroutines

- subroutine stereo (QSZ, SCALE, NPIX, PICID, PUSE, V, UX, UY, UZ, HUSE, HT, Z0, KK, ESZ, DNK, ZUSE, ZHT, MAN)

  Procedure.

2.125.1 Function/Subroutine Documentation

2.125.1.1 **subroutine stereo (integer *QSZ,* double precision *SCALE,* integer *NPIX,* character∗12, dimension(pmx) *PICID,* logical, dimension(pmx) *PUSE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* integer, dimension(2,pmx) *Z0,* integer, dimension(pmx) *KK,* integer *ESZ,* integer, dimension(-128:128,-128:128,pmx) *DNK,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZHT,* logical *MAN*)**

Procedure.

Abstract

This subroutine determines local heights in a maplets at a sparse set of points to act as conditioning heights in the slope to height integration in subroutines SLP2HGT and NNEIGHBORS. Because the projection of an image !! onto a maplet depends upon the height at the maplet point (i,j), a pixel from an obliquely viewed image will move on trajectory di/dh, dj/dh as the !! height is varied. Heights are varied until a maximum correlation is achieved between the set of images at a given point. The correlation can be between raw brightnesses or between a gradient value that is independent of local solar elevation (see subroutine EXTRACT_GRAD). The matchup among image data can be done automatically (preferred) or manually.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

QSZ SCALE NPIX PICID PUSE V UX UY UZ HUSE HT Z0 KK ESZ DNK ZUSE ZHT MAN

File_I/O

Filename I/O Description

SUMFILES/<PICID>.SUM I Image summary files

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN SLEN

SPC_subroutines_called CLEAR_DATA (Inside program LITHOS) DISPLAY1 EXTRACT_DATA_PIC V2IMGPL

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

History 2014_02_22: DISPLAY replaced with DISPLAY1.

2.126 COMMON/STEREO_LO.f File Reference

Functions/Subroutines

- subroutine stereo_lo (QSZ, SCALE, NPIX, PICID, PUSE, V, UX, UY, UZ, HUSE, HT, Z0, KK, DNK, ZUSE, ZHT)

  Procedure.

2.126.1 Function/Subroutine Documentation

2.126.1.1 subroutine stereo_lo (integer *QSZ,* double precision *SCALE,* integer *NPIX,* character∗12, dimension(pmx) *PICID,* logical, dimension(pmx) *PUSE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* integer, dimension(2,pmx) *Z0,* integer, dimension(pmx) *KK,* integer, dimension(-128:128,-128:128,pmx) *DNK,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ZHT*)

Procedure.

Abstract

This subroutine determines local heights in a maplets at a sparse set of points to act as conditioning heights in the slope to height integration in subroutines SLP2HGT and NNEIGHBORS. Because the projection of an image !! onto a maplet depends upon the height at the maplet point (i,j), a pixel from an obliquely viewed image will move on trajectory di/dh, dj/dh as the !! height is varied. Heights are varied until a maximum correlation is achieved between the set of images at a given point. The correlation can be between raw brightnesses or between a gradient value that is independent of local solar elevation (see subroutine EXTRACT_GRAD). The matchup among image data is done automatically (preferred). There is no manual option.

This subroutine employs a lockout file to guarantee that the file PICNM.SUM cannot be opened simultaneously with another process that might be attampting to write to it.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

QSZ SCALE NPIX PICID PUSE V UX UY UZ HUSE HT Z0 KK ESZ DNK ZUSE ZHT MAN

File_I/O

Filename I/O Description

./TESTFILES/<PICID>.SUM O Lockout file for <PICID>.SUM ./SUMFILES/<PICID>.SUM I Image summary files

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN SLEN

SPC_subroutines_called DELAY EXTRACT_DATA_PIC_LO

SPICELIB_functions_called RPD VDOT

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOSP

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.127 COMMON/TUCK.f File Reference

Functions/Subroutines

- subroutine tuck (PICNM)

  Procedure.

2.127.1 Function/Subroutine Documentation

2.127.1.1 subroutine tuck (character∗12 *PICNM*)

Procedure.

Abstract

Thus subroutine "tucks away" an image so that its .SUM file is neither used nor processed by the software. It removes all landmark data from the .SUM file and places a ! indicator in the first column of all PICTLIST files. If the image has already been tucked, then application of this subroutine will reverse (untuck) the image without, however, restoring the landmark data.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

PICNM I Image name

File_I/O

Filename I/O Description

PICTLIST.TXT O Image listing file PICTLISTR.TXT O Image listing file PICTLISTRX.TXT O Image listing file SU↩ MFILES/<PICID>.SUM I Image summary file for <PICID>

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELETEINLMK_LIM DELETEINLMK_PIC

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS REGISTER

2.128 COMMON/U2VN.f File Reference

Functions/Subroutines

- subroutine u2vn (UZ, V, N)

  Procedure.

2.128.1 Function/Subroutine Documentation

2.128.1.1 subroutine u2vn (double precision, dimension(3) *UZ,* double precision, dimension(3) *V,* real∗8, dimension(3) *N*)

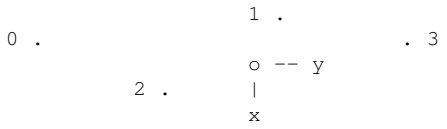
Procedure.

Parameters

n	Abstract

This subroutine predicts the vector to a landmark center at a given latitude and longitude (specified by a unit vector UZ) based on the current shape model. It also estimates the normal to the surface N at the landmark.

The unit vector UZ pierces the reference shape model within, or on the boundary of, one of the quadrilateral cells. Projected onto a plane with unit vectors UX and UY perpendicular to UZ, the corners of the cell are at X(I), Y(I), I=0,3.



The conditions for UZ to pierce the cell are:

```
X(0)*Y(1)-Y(0)*X(1) < 0        X(1)*Y(3)-Y(1)*X(3) < 0
X(3)*Y(2)-Y(3)*X(2) < 0        X(2)*Y(0)-Y(2)*X(0) < 0
```

the first, for example, being the cross product of the o0 and o1 vectors (in the UZ direction).

Once a cell has been determined, the coordinates of the piercing point o are determined by bilinear interpolation. The corner points have labels (i,j) where 0=(0,0), 1=(1,0), 2=(0,1), 3=(1,1) and the bilinear interpolation in terms of i,j takes the form:

```
F(i,j) = F0 + F1*i + F2*j + F3*i*j
```

where, for example,

```
F0 = F(0)
F1 = F(1)-F(0)
F2 = F(2)-F(0)
F3 = F(0)-F(1)-F(2)+F(3)
```

The piercing point is at (0,0,Zo) where Zo is the magnitude of the vector along UZ. The bilinear interpolations giving 0 for the x- and y- coordinates are:

```
0 = X0 + X1*i +X2*j + X3*i*j
0 = Y0 + Y1*i +Y2*j + Y3*i*j
```

These are solved for the i,j corresponding to the piercing point. If Z(0) - Z(3) are the magnidudes of the corner vectors, the magnitude Zo is

```
Zo = Z0 + Z1*i +Z2*j + Z3*i*j
```

For reasons now lost in antiquity, a small parameter EPS has been included to allow the piercing point to spill slightly outsige the cell. This choice will be examined during testing.

The normal at the landmark is estimated as the cross product of the diagonals of the cell.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

UZ V N

File_I/O

Filename I/O Description

None

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_MODEL

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called UCRSS VEQU VHAT VSCL VSUB

Called_by_SPC_Programs LITHOS REGISTER RESIDUALS BIGMAP BIGMAPL SUBROUTINE CREATE_LMF↩ ILE SUBROUTINE POINTS2MAPS

2.129 COMMON/U2VRF.f File Reference

Functions/Subroutines

- subroutine u2vrf (UZ, V)

  Procedure.

2.129.1 Function/Subroutine Documentation

2.129.1.1 subroutine u2vrf (double precision, dimension(3) *UZ,* double precision, dimension(3) *V*)

Procedure.

Abstract

This subroutine predicts the vector to a landmark center at a given latitude and longitude (specified by a unit vector UZ) based on a nominal reference shape. It is occasionally used to constrain landmark solutions when the s/!!states are not well known.

The unit vector UZ pierces the reference shape model within, or on the boundary of, one of the quadrilateral cells. Projected onto a plane with unit vectors UX and UY perpendicular to UZ, the corners of the cell are at X(I), Y(I), I=0,3.

```
                    1 .
 0 .                          . 3
```

```
                    o -- y
          2 .       |
                    x
```

The conditions for UZ to pierce the cell are:

```
X(0)*Y(1)-Y(0)*X(1) < 0        X(1)*Y(3)-Y(1)*X(3) < 0
X(3)*Y(2)-Y(3)*X(2) < 0        X(2)*Y(0)-Y(2)*X(0) < 0
```

the first, for example, being the cross product of the o0 and o1 vectors (in the UZ direction).

Once a cell has been determined, the coordinates of the piercing point o are determined by bilinear interpolation. The corner points have labels (i,j) where 0=(0,0), 1=(1,0), 2=(0,1), 3=(1,1) and the bilinear interpolation in terms of i,j takes the form:

```
F(i,j) = F0 + F1*i + F2*j + F3*i*j
```

where, for example,

```
F0 = F(0)
F1 = F(1)-F(0)
F2 = F(2)-F(0)
F3 = F(0)-F(1)-F(2)+F(3)
```

The piercing point is at (0,0,Zo) where Zo is the magnitude of the vector along UZ. The bilinear interpolations giving 0 for the x- and y- coordinates are:

```
0 = X0 + X1*i +X2*j + X3*i*j
0 = Y0 + Y1*i +Y2*j + Y3*i*j
```

These are solved for the i,j corresponding to the piercing point. If Z(0) - Z(3) are the magnidudes of the corner vectors, the magnitude Zo is

```
Zo = Z0 + Z1*i +Z2*j + Z3*i*j
```

For reasons now lost in antiquity, a small parameter EPS has been included to allow the piercing point to spill slightly outsige the cell. This choice will be examined during testing.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

Var name I Description of input variables Var name O Description of output variables

File_I/O

Filename I/O Description

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called None

SPICELIB_functions_called UCRSS VEQU VHAT VSCL VSUB

SPICELIB_subroutines_called VDOT VNORM

Called_by_SPC_Programs LITHOS GEOMETRY SUBROUTINE IPL2VLM

2.130 COMMON/V2IMGPL.f File Reference

Functions/Subroutines

- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D, CX, CY, CZ, USE, IMGPL)

  Procedure.

2.130.1 Function/Subroutine Documentation

2.130.1.1 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(2) *IMGPL*)

Procedure.

Abstract

V2INGPL determines pixel/line location of a landmark in image PICNM.

V0 is body fixed vector from s/!!to body center. V is body-fixed landmark vector. U=V0+V is s/!!- landmark vector. VDOT(U,Ci) = Zi (i=x,y,z) are projections of U on camera axes. GETMM projects these onto the camera focal plane, taking into account optical distortions. MMPX translates from focal plane space to image pixel/line space.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

V V0 PICNM NPX NLN MMFL CTR KMAT D CX CY CZ USE IMGPL

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GETMM MMPX

SPICELIB_functions_called VDOT

SPICELIB_subroutines_called VADD VHAT

Called_by_SPC_Programs AUTOREGISTER LITHOS REGISTER RESIDUALS SHIFT REGRES BIGMAP SUB↩ ROUTINE CREATE_LMFILE SUBROUTINE EXTRACT_DATA_PIC SUBROUTINE LIMB_HEIGHTS SUBROUT↩ INE LOC2PIX SUBROUTINE STEREO SUBROUTINE VISIBLE

2.131 COMMON/VERSION.F File Reference

Functions/Subroutines

- character ∗80 function version ()

  Procedure.

2.131.1 Function/Subroutine Documentation

2.131.1.1 character∗80 function version ( )

Procedure.

Returns

Abstract

```
SIMPLY RETURNS THE VERSION NUMBER OF THE BUILD
```

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

None

File_I/O

Filename I/O Description

None

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

E. E. Palmer (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_UL RAW2PGM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

Notes: 2014_04_22 Created 2014_05_07 Changed to character∗80 to be consistent.

2.132 COMMON/VISIBLE.f File Reference

Functions/Subroutines

- subroutine visible (NTMP, LMKNM, PICNM, DZ)

  Procedure.

2.132.1 Function/Subroutine Documentation

2.132.1.1 subroutine visible (integer *NTMP,* character∗6 *LMKNM,* character∗12 *PICNM,* double precision *DZ*)

Procedure.

Parameters

ntmp	Abstract

This subroutine determines the maximum distance between the central and corner points of a maplet and the closest corresponding surface points to the camera along a line of sight, determined from the shape model. If this distance DZ is greater than some threshold, it indicates that all or part of the maplet is blocked from view by a closer portion of the body.

If a file NVIZLIST.TXT containing images with the potential for such occlusions exists, the subroutine will return a value DZ=0 when the image is not on the list.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP LMKNM PICNM DZ

File_I/O

Filename I/O Description

NVIZLIST.TXT I MAPFILES/<LMKNM>.MAP I Map file for landmark <LMKNM> LMKFILES/<LMKNM>.LMK I Landmark file for landmark <LMKNM> SUMFILES/<PICNM>.SUM I Summary file for image <PICNM>

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called READ_MAP GET_HEIGHTS MM2VN V2IMGPL PXMM

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs LITHOS AUTOREGISTER

2.133 COMMON/VISIBLE_LO.f File Reference

Functions/Subroutines

- subroutine visible_lo (NTMP, LMKNM, PICNM, DZ)

  Procedure.

2.133.1 Function/Subroutine Documentation

2.133.1.1 subroutine visible_lo (integer *NTMP,* character∗6 *LMKNM,* character∗12 *PICNM,* double precision *DZ*)

Procedure.

Parameters

ntmp	Abstract

This subroutine determines the maximum distance between the central and corner points of a maplet and the closest corresponding surface points to the camera along a line of sight, determined from the shape model. If this distance DZ is greater than some threshold, it indicates that all or part of the maplet is blocked from view by a closer portion of the body.

If a file NVIZLIST.TXT containing images with the potential for such occlusions exists, the subroutine will return a value DZ=0 when the image is not on the list.

This form of the VISIBLE subroutine has a lockout file that prevents two processes from reading or writing to a landmark file at the same time.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations Variable_I/O

Variable I/O Description

NTMP LMKNM PICNM DZ

File_I/O

Filename I/O Description

./NVIZLIST.TXT I ./TESTFILES.<LMKNM>.LMK O Lockout file indicator landmark ./LMKFILES/<LMKNM>.LMK I Landmark file for landmark <LMKNM> ./SUMFILES/<PICNM>.SUM I Summary file for image <PICNM>

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called DELAY READ_MAP GET_HEIGHTS MM2VN V2IMGPL PXMM

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called VSUB

Called_by_SPC_Programs AUTOREGISTERP

History 2013_10_07: Capitalization, compiler warnings fixed and header added.

2.134 COMMON/WHOLEPIC.f File Reference

Functions/Subroutines

- subroutine wholepic (PICNM, dn)

2.134.1 Function/Subroutine Documentation

2.134.1.1 subroutine wholepic (character∗12 *PICNM,* integer, dimension(4096,4096) *dn*)

2.135 COMMON/WRITE_MAP.f File Reference

Functions/Subroutines

- subroutine write_map (LMRKFILE, NTMP, QSZ, SCALE, V, VSIG, UX, UY, UZ, HT, HUSE, TMPL)

  Procedure.

2.135.1 Function/Subroutine Documentation

2.135.1.1 subroutine write_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *VSIG,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp,3) *TMPL*)

Procedure.

Abstract This subroutine will create a mapfile <LMRKFILE>.

The mapfile is made up of 72 byte records. The first record contains information describing the size, scale, orientation and position of the map:

bytes 1-6 Unused bytes 7-10 Scale in km/pixel (real∗4 msb) bytes 11-12 qsz where map is 2∗qsz+1 x 2∗qsz+1 pixels (unsigned short lsb) bytes 16-27 map center body fixed position vector in km 3 x (real∗4 msb) bytes 28-39 Ux body fixed unit map axis vector 3 x (real∗4 msb) bytes 40-51 Uy body fixed unit map axis vector 3 x (real∗4 msb) bytes 52-63 Uz body fixed unit map normal vector 3 x (real∗4 msb) bytes 64-67 Hscale = maximum abs(height)/30000 (real∗4 msb) ∗ byte 13 255∗ X position uncertainty unit vector component (byte) + byte 14 255∗ Y position uncertainty unit vector component (byte) + byte 15 255∗ Z position uncertainty unit vector component (byte) + bytes 68-71 magnitude of position uncertainty (real∗4 msb) + byte 72 Unused

- heights are in units of map scale
- these are pretty much unused as far as I can see.

The remaining records are made up of 3 byte chunks:

bytes 1-2 height/hscale (integer∗2 msb) byte 3 relative "albedo" (1-199) (byte)

If there is missing data at any point, both height and albedo are set to zero.

The map array is filled row by row from the upper left (i,j = -qsz). Rows are increasing in the Uy direction with spacing = scale Columns are increasing in the Ux direction with spacing = scale Heights are positive in the Uz direction with units = scale

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

LMRKFILE I Basename of the maplet file to write to NTMP I QSZ I Map size/2 SCALE I Scale of Map in km V I VSIG I Uncertainties in V vector values UX I X unit vector UY I Y unit vector UZ I Z unit vector HT I Height HUSE I Use flag TMPL I Map values

File_I/O

Filename I/O Description

<LMRKFILE> O Map file that will be created or updated.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called FLIP

SPICELIB_functions_called VNORM

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

2.136 COMMON/ZOOM.f File Reference

Functions/Subroutines

- subroutine zoom (DN0, QSZ0, NSZ0, N)

  Procedure.

2.136.1 Function/Subroutine Documentation

2.136.1.1 subroutine zoom (real*4, dimension(5000,5000) *DN0,* integer *QSZ0,* integer *NSZ0,* integer *N*)

Procedure.

Abstract

If there are many images represented in a landmark display, it is not easy to identify one of interest by number. This subroutine recasts the display with ten images on a line and the number of the last image in the line to the right. The cells are 99x99 pixels with a one pixel separation (ie qsz=49). It the maplet has qsz different from 49, the display includes only the central 99x99 pixels of the maplet.

Disclaimer

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

Variable_I/O

Variable I/O Description

DN0 QSZ0 NSZ0 N

File_I/O

Filename I/O Description

LMRK_DISPLAY1.gray I Gray scale file. LMRK_DISPLAY0.pgm I pgm file converted from LMRK_DISPLAY1.gray

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called GET_UL RAW2PGM

SPICELIB_functions_called None

SPICELIB_subroutines_called None

Called_by_SPC_Programs LITHOS

Notes: 2013_01_12 font now enters as include file

2.137 COVERAGE/coverage.f File Reference

Functions/Subroutines

- program __coverage.f__
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_data (INFILE, MX, IMAX, JMAX, V, UX, UY, UZ, VK, ALB, KMSCALE)
- subroutine flip (n, lflag, ch1, ch2)
- integer function kount (LMKNM, K)

2.137.1 Function/Subroutine Documentation

2.137.1.1 program __coverage.f__ ( )

2.137.1.2 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.137.1.3 subroutine get_data (character∗72 *INFILE,* integer *MX,* integer *IMAX,* integer *JMAX,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision, dimension(3,mx,mx) *VK,* double precision, dimension(mx,mx) *ALB,* double precision *KMSCALE*)

2.137.1.4 integer function kount (character∗6 *LMKNM,* integer *K*)

2.137.1.5 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.138 COVERAGE/coverage_p.f File Reference

Functions/Subroutines

- program __coverage_p.f__

2.138.1 Function/Subroutine Documentation

2.138.1.1 program __coverage_p.f__ ( )

2.139 COVERAGE/map_coverage.f File Reference

Functions/Subroutines

- program __map_coverage.f__

  Procedure map_coverage.

2.139.1 Function/Subroutine Documentation

2.139.1.1 program __map_coverage.f__ ( )

Procedure map_coverage.

Abstract Required_Reading Declarations

2.140 COVERAGE/map_coverage_p.f File Reference

Functions/Subroutines

- program __map_coverage_p.f__

2.140.1 Function/Subroutine Documentation

2.140.1.1 program __map_coverage_p.f__ ( )

2.141 extra/BACKGROUND.f File Reference

Functions/Subroutines

- program __background.f__
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec)
- subroutine picinpt (PICNM, dn)
- subroutine find_picfile (PICNM, PICFILE, EX)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, PX, MM)
- subroutine mmpx (KMAT, CTR, MM, PX)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)

2.141.1 Function/Subroutine Documentation

2.141.1.1 program __background.f__ ( )

2.141.1.2 subroutine find_picfile (character∗12 *PICNM,* character∗72 *PICFILE,* logical *EX*)

2.141.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.141.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.141.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.141.1.6 subroutine picinpt (character∗12 *PICNM,* integer, dimension(4096,4096) *dn*)

2.141.1.7 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.141.1.8 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.141.1.9 integer∗4 function slen (character∗(∗) *STRING*)

2.142 extra/inertial.f File Reference

Functions/Subroutines

- program __inertial.f__
- integer ∗4 function slen (STRING)
- integer function lmcount (PICNM, K)

2.142.1 Function/Subroutine Documentation

2.142.1.1 program __inertial.f__ ( )

2.142.1.2 integer function lmcount (character∗12 *PICNM,* integer *K*)

2.142.1.3 integer∗4 function slen (character∗(∗) *STRING*)

2.143 extra/LIMBEX.f File Reference

Functions/Subroutines

- program __limbex.f__
- integer ∗4 function slen (STRING)
- subroutine get_model (infile, q, vec)
- subroutine extend (sz, i1, j1, f1, i2, j2, f2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine mmpx (KMAT, CTR, D, MM, PX)

2.143.1 Function/Subroutine Documentation

2.143.1.1 program __limbex.f__ ( )

2.143.1.2 subroutine extend (integer∗4 *sz,* integer∗4 *i1,* integer∗4 *j1,* integer∗4 *f1,* integer∗4 *i2,* integer∗4 *j2,* integer∗4 *f2*)

2.143.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.143.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.143.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.143.1.6 integer∗4 function slen (character∗(∗) *STRING*)

2.143.1.7 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.144 extra/mosaicM.f File Reference

Functions/Subroutines

- program __mosaicm.f__
- subroutine v2imgpl (V, V0, NPX, NLN, MMFL, CTR, KMAT, D, CX, CY, CZ, USE, IMGPL)
- integer ∗4 function slen (STRING)
- subroutine extract_data (QSZ, SCALE, PICNM, IPL, V, VC, UX, UY, UZ, HUSE, HT, dpic_dloc, didh, djdh, dn, dnx)
- subroutine mm2v (MM, MMFL, V0, CX, CY, CZ, USE, V)
- subroutine picinpt (PICNM, dn)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine u2v (UZ, V)
- real ∗8 function brt0 (z, npx, nln, dn)
- subroutine find_picfile (PICNM, PICFILE, EX)
- subroutine get_model (infile, q, vec)
- subroutine loc2pix (PICNM, V, UX, UY, UZ, SCALE, DPIC_DLOC)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine pxmm (KMAT, CTR, D, PX, MM)
- subroutine get_map (NAME, QSZ, V, UX, UY, UZ, S0, S1, AL, HT, HUSE)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)
- subroutine imgpl2v (PICNM, IMGPL, USE, V)

2.144.1 Function/Subroutine Documentation

2.144.1.1 program __mosaicm.f__ ( )

2.144.1.2 real∗8 function brt0 (real∗8, dimension(2) *z,* integer∗4 *npx,* integer∗4 *nln,* integer∗4, dimension(2048,2048) *dn*)

2.144.1.3 subroutine extract_data (integer *QSZ,* double precision *SCALE,* character∗12 *PICNM,* double precision, dimension(2) *IPL,* double precision, dimension(3) *V,* double precision, dimension(3) *VC,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *HUSE,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* double precision, dimension(2,3) *dpic_dloc,* double precision *didh,* double precision *djdh,* integer, dimension(2048,2048) *dn,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *dnx*)

2.144.1.4 subroutine find_picfile (character∗12 *PICNM,* character∗72 *PICFILE,* logical *EX*)

2.144.1.5 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.144.1.6 subroutine get_map (character∗6 *NAME,* integer *QSZ,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *S0,* double precision *S1,* real∗4, dimension(-btmp:btmp,-btmp:btmp) *AL,* real∗4, dimension(-btmp:btmp,-btmp:btmp) *HT,* logical, dimension(-btmp:btmp,-btmp:btmp) *HUSE*)

2.144.1.7 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.144.1.8 subroutine imgpl2v (character∗12 *PICNM,* double precision, dimension(2) *IMGPL,* logical *USE,* double precision, dimension(3) *V*)

2.144.1.9 subroutine loc2pix (character∗12 *PICNM,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* double precision *SCALE,* double precision, dimension(2,3) *DPIC_DLOC*)

2.144.1.10 subroutine mm2v (double precision, dimension(2) *MM,* double precision *MMFL,* double precision, dimension(3) *V0,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(3) *V*)

2.144.1.11 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.144.1.12 subroutine picinpt (character∗12 *PICNM,* integer, dimension(2048,2048) *dn*)

2.144.1.13 subroutine pxmm (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.144.1.14 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.144.1.15 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.144.1.16 integer∗4 function slen (character∗(∗) *STRING*)

2.144.1.17 subroutine u2v (double precision, dimension(3) *UZ,* double precision, dimension(3) *V*)

2.144.1.18 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(2) *IMGPL*)

2.145 extra/NOBODY.f File Reference

Functions/Subroutines

- program __nobody.f__
- integer ∗4 function slen (STRING)
- subroutine get_model (infile, q, vec)
- subroutine extend (sz, i1, j1, f1, i2, j2, f2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine mmpx (KMAT, CTR, D, MM, PX)

2.145.1 Function/Subroutine Documentation

2.145.1.1 program __nobody.f__ ( )

2.145.1.2 subroutine extend (integer∗4 *sz,* integer∗4 *i1,* integer∗4 *j1,* integer∗4 *f1,* integer∗4 *i2,* integer∗4 *j2,* integer∗4 *f2*)

2.145.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.145.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.145.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.145.1.6 integer∗4 function slen (character∗(∗) *STRING*)

2.145.1.7 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.146 extra/TERMEX.f File Reference

Functions/Subroutines

- program __termex.f__
- integer ∗4 function slen (STRING)
- subroutine get_model (infile, q, vec)
- subroutine extend (sz, i1, j1, f1, i2, j2, f2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine mmpx (KMAT, CTR, D, MM, PX)

2.146.1 Function/Subroutine Documentation

2.146.1.1 program __termex.f__ ( )

2.146.1.2 subroutine extend (integer∗4 *sz,* integer∗4 *i1,* integer∗4 *j1,* integer∗4 *f1,* integer∗4 *i2,* integer∗4 *j2,* integer∗4 *f2*)

2.146.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.146.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.146.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.146.1.6 integer∗4 function slen (character∗(∗) *STRING*)

2.146.1.7 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.147 FONT.INC File Reference

2.148 IMAGER/Display.f File Reference

Functions/Subroutines

- program __display.f__

2.148.1 Function/Subroutine Documentation

2.148.1.1 program __display.f__ ( )

2.149 IMAGER/Imager_Grid.f File Reference

Functions/Subroutines

- program __imager_grid.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)

2.149.1 Function/Subroutine Documentation

2.149.1.1 program __imager_grid.f__ ( )

2.149.1.2 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.149.1.3 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.149.1.4 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.149.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.149.1.6 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.149.1.7 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.149.1.8 integer∗4 function slen (character∗(∗) *STRING*)

2.150 IMAGER/Imager_GridL.f File Reference

Functions/Subroutines

- program __imager_gridl.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)

2.150.1 Function/Subroutine Documentation

2.150.1.1 program __imager_gridl.f__ ( )

2.150.1.2 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.150.1.3 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.150.1.4 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.150.1.5 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.150.1.6 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.150.1.7 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.150.1.8 integer∗4 function slen (character∗(∗) *STRING*)

2.151 IMAGER/Imager_MG.f File Reference

Functions/Subroutines

- program __imager_mg.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)

- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine read_header (LMRKFILE, QSZ, SCALE, V, UX, UY, UZ)

2.151.1 Function/Subroutine Documentation

2.151.1.1 program __imager_mg.f__ ( )

2.151.1.2 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.151.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.151.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.151.1.5 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.151.1.6 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.151.1.7 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.151.1.8 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.151.1.9 subroutine read_header (character∗72 *LMRKFILE,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.151.1.10 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.151.1.11 integer∗4 function slen (character∗(∗) *STRING*)

2.151.1.12 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.152 IMAGER/Imager_MGF.f File Reference

Functions/Subroutines

- program __imager_mgf.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine read_header (LMRKFILE, QSZ, SCALE, V, UX, UY, UZ)

2.152.1 Function/Subroutine Documentation

2.152.1.1 program __imager_mgf.f__ ( )

2.152.1.2 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.152.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.152.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.152.1.5 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.152.1.6 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.152.1.7 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.152.1.8 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.152.1.9 subroutine read_header (character∗72 *LMRKFILE,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.152.1.10 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.152.1.11 integer∗4 function slen (character∗(∗) *STRING*)

2.152.1.12 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.153 IMAGER/Imager_MGFL.f File Reference

Functions/Subroutines

- program __imager_mgfl.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine read_header (LMRKFILE, QSZ, SCALE, V, UX, UY, UZ)

2.153.1 Function/Subroutine Documentation

2.153.1.1 program __imager_mgfl.f__ ( )

2.153.1.2 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.153.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.153.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.153.1.5 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.153.1.6 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.153.1.7 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.153.1.8 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.153.1.9 subroutine read_header (character∗72 *LMRKFILE,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.153.1.10 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.153.1.11 integer∗4 function slen (character∗(∗) *STRING*)

2.153.1.12 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.154 IMAGER/Imager_MGL.f File Reference

Functions/Subroutines

- program __imager_mgl.f__
- real ∗8 function illum (ndots, ndotb, alpha)
- integer ∗4 function slen (STRING)
- subroutine raw2pgm (infile, outfile, npx, nln)
- subroutine get_model (infile, q, vec, a)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine pxmm (PICNM, MMFL, KMAT, CTR, D, PX, MM)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)
- subroutine v2imgpl (V, V0, PICNM, NPX, NLN, MMFL, CTR, KMAT, D,
- subroutine read_header (LMRKFILE, QSZ, SCALE, V, UX, UY, UZ)

2.154.1 Function/Subroutine Documentation

2.154.1.1 program __imager_mgl.f__ ( )

2.154.1.2 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.154.1.3 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* real∗8, dimension(0:512,0:512,6) *a*)

2.154.1.4 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.154.1.5 real∗8 function illum (real∗8 *ndots,* real∗8 *ndotb,* real∗8 *alpha*)

2.154.1.6 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.154.1.7 subroutine pxmm (character∗12 *PICNM,* double precision *MMFL,* double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.154.1.8 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.154.1.9 subroutine read_header (character∗72 *LMRKFILE,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.154.1.10 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.154.1.11 integer∗4 function slen (character∗(∗) *STRING*)

2.154.1.12 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D*)

2.155 IMAGER/make_font.f File Reference

Functions/Subroutines

- program __make_font.f__

2.155.1 Function/Subroutine Documentation

2.155.1.1 program __make_font.f__ ( )

2.156 IMAGER/MOSAIC.f File Reference

Functions/Subroutines

- program __mosaic.f__

2.156.1 Function/Subroutine Documentation

2.156.1.1 program __mosaic.f__ ( )

2.157 IMAGER/SHOWMAP.f File Reference

Functions/Subroutines

- program __showmap.f__

2.157.1 Function/Subroutine Documentation

2.157.1.1 program __showmap.f__ ( )

2.158 IMAGER/SHOWMAPC.f File Reference

Functions/Subroutines

- program __showmapc.f__

2.158.1 Function/Subroutine Documentation

2.158.1.1 program __showmapc.f__ ( )

2.159 IMAGER/view_map_rgb.f File Reference

Functions/Subroutines

- program __view_map_rgb.f__
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat, mx1, mx2,
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds1, ds2)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)
- subroutine get_data (INFILE, MX, IMAX, JMAX, V, UX, UY, UZ,
- subroutine flip (n, lflag, ch1, ch2)
- subroutine get_ul (chr, tp, tl)
- subroutine raw2ppm (infile, outfile, npx, nln)

2.159.1 Function/Subroutine Documentation

2.159.1.1 program __view_map_rgb.f__ ( )

2.159.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.159.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.159.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.159.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1↩ :mx1,-mx1:mx1) *arr,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.159.1.6 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.159.1.7 subroutine get_data (character∗72 *INFILE,* integer *MX,* integer *IMAX,* integer *JMAX,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.159.1.8 subroutine get_ul (character∗1 *chr,* integer∗4 *tp,* integer∗4 *tl*)

2.159.1.9 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.159.1.10 subroutine raw2ppm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.159.1.11 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.159.1.12 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.159.1.13 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use, cmat,* integer∗4 *mx1,* integer∗4 *mx2*)

2.160 IMAGER/view_map_stereo.f File Reference

Functions/Subroutines

- program __view_map_stereo.f__
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat, mx1, mx2,
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds1, ds2)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)
- subroutine get_data (INFILE, MX, IMAX, JMAX, V, UX, UY, UZ,
- subroutine flip (n, lflag, ch1, ch2)
- subroutine get_ul (chr, tp, tl)
- subroutine raw2pgm (infile, outfile, npx, nln)

2.160.1 Function/Subroutine Documentation

2.160.1.1 program __view_map_stereo.f__ ( )

2.160.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.160.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.160.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.160.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1↩ :mx1,-mx1:mx1) *arr,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.160.1.6 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.160.1.7 subroutine get_data (character∗72 *INFILE,* integer *MX,* integer *IMAX,* integer *JMAX,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.160.1.8 subroutine get_ul (character∗1 *chr,* integer∗4 *tp,* integer∗4 *tl*)

2.160.1.9 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.160.1.10 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.160.1.11 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.160.1.12 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.160.1.13 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use, cmat,* integer∗4 *mx1,* integer∗4 *mx2*)

2.161 IMAGER/view_maps.f File Reference

Functions/Subroutines

- program __view_maps.f__
- subroutine raw2pgm (infile, outfile, npx, nln)
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat,
- subroutine dzbuffer0 (mx1, max0, mx, my, arr, ds0, ds1, ds2)
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds0, ds1, ds2)
- subroutine invertn (N, M, MINV)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)
- subroutine get_data (INFILE, MX, IMAX, JMAX, V, UX, UY, UZ,
- subroutine flip (n, lflag, ch1, ch2)
- integer function kount (LMKNM, K)
- subroutine get_model (infile, q, vec)

2.161.1 Function/Subroutine Documentation

2.161.1.1 program __view_maps.f__ ( )

2.161.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.161.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.161.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.161.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1:mx1,-mx1:mx1) *arr,* real∗8, dimension(-mx1:mx1,-mx1:mx1) *ds0,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.161.1.6 subroutine dzbuffer0 (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1:mx1,-mx1:mx1) *arr,* real∗8, dimension(-mx1:mx1,-mx1:mx1) *ds0,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.161.1.7 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.161.1.8 subroutine get_data (character∗72 *INFILE,* integer *MX,* integer *IMAX,* integer *JMAX,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.161.1.9 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.161.1.10 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.161.1.11 subroutine invertn (integer∗4 *N,* real∗8, dimension(6,6) *M,* real∗8, dimension(6,6) *MINV*)

2.161.1.12 integer function kount (character∗6 *LMKNM,* integer *K*)

2.161.1.13 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.161.1.14 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.161.1.15 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.161.1.16 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use, cmat*)

2.162 IMAGER/view_shape.f File Reference

Functions/Subroutines

- program __view_shape.f__
- subroutine raw2pgm (infile, outfile, npx, nln)
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat, mx1, mx2,
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds1, ds2)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)

2.162.1 Function/Subroutine Documentation

2.162.1.1 program __view_shape.f__ ( )

2.162.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.162.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.162.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.162.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1↩ :mx1,-mx1:mx1) *arr,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.162.1.6 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.162.1.7 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.162.1.8 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.162.1.9 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.162.1.10 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use, cmat,* integer∗4 *mx1,* integer∗4 *mx2*)

2.163 IMAGER/view_shapeA.f File Reference

Functions/Subroutines

- program __view_shapea.f__
- subroutine raw2pgm (infile, outfile, npx, nln)
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat, mx1, mx2,
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds1, ds2)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)

2.163.1 Function/Subroutine Documentation

2.163.1.1 program __view_shapea.f__ ( )

2.163.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.163.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.163.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.163.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1↩ :mx1,-mx1:mx1) *arr,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.163.1.6 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.163.1.7 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.163.1.8 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.163.1.9 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.163.1.10 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use, cmat,* integer∗4 *mx1,* integer∗4 *mx2*)

2.164 IMAGER/view_shapeC.f File Reference

Functions/Subroutines

- program __view_shapec.f__
- subroutine raw2ppm (infile, outfile, npx, nln)
- subroutine raw2pgm (infile, outfile, npx, nln)
- real ∗8 function illum (ndotb, ndots, beta)
- subroutine zbuffer (imax, jmax, vk, use, cmat, mx1, mx2,
- subroutine dzbuffer (mx1, max0, mx, my, arr, ds1, ds2)
- subroutine unit (v, mag)
- subroutine cross (v1, v2, v)
- subroutine ucross (v1, v2, v, mag)
- real ∗8 function dot (v1, v2)
- subroutine bilinear (div, c0, c1, c2, c3)

2.164.1 Function/Subroutine Documentation

2.164.1.1 program __view_shapec.f__ ( )

2.164.1.2 subroutine bilinear (integer∗4 *div,* real∗8 *c0,* real∗8 *c1,* real∗8 *c2,* real∗8 *c3*)

2.164.1.3 subroutine cross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v*)

2.164.1.4 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.164.1.5 subroutine dzbuffer (integer∗4 *mx1,* real∗8 *max0,* integer∗4 *mx,* integer∗4 *my,* real∗8, dimension(-mx1↩ :mx1,-mx1:mx1) *arr,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds1,* real∗4, dimension(-mx1:mx1,-mx1:mx1) *ds2*)

2.164.1.6 real∗8 function illum (real∗8 *ndotb,* real∗8 *ndots,* real∗8 *beta*)

2.164.1.7 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.164.1.8 subroutine raw2ppm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.164.1.9 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.164.1.10 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.164.1.11 subroutine zbuffer (integer∗4 *imax,* integer∗4 *jmax,* real∗8, dimension(3,mx2,mx2) *vk,* logical, dimension(mx2,mx2) *use,* *cmat,* integer∗4 *mx1,* integer∗4 *mx2*)

2.165 LITHOSPHERE/AUTOREGISTER.f File Reference

Functions/Subroutines

- program __autoregister.f__

  Procedure AUTOREGISTER.
- subroutine correlate (LMX, NPIX, QSZ, DN1, LMKID, X0, Y0, Z0)

  Variable_I/O.
- subroutine correlate_lmk (DN1, QSZ, I0, J0, S, X0, Y0, Z0, FOUND)
- subroutine clear_data (QSZ, N, DN)
- subroutine display (DN, QSZ, N)

2.165.1 Function/Subroutine Documentation

2.165.1.1 program __autoregister.f__ ( )

Procedure AUTOREGISTER.

Abstract

This procedure aligns existing maplets with imaging data from a single image and uses subroutine IPL2SCOBJPTG to update the camera pointing and spacecraft-object vector in the corresponding SUMFILE. The program uses the file LMRKLISTX.TXT to pre- screen the maplets, so if maplets have been added or deleted recently, the procedure MAKE_LMRKLISTX should be run.

The program first asks for a picture name. It produces a list of existing landmarks, if any, and asks if more should be added. An initial filter for added images is set in INIT_LITHOS.TXT. RESLM is the maximum ratio of the image resolution to the maplet scale, while SIZLM is the maximum ratio of the linear maplet size to the image size. NUMLM sets the minimum number of maplets found in the image before the other two filters come into play. The user is asket for two more inputs:

The fractional width=0.5 (what I usually use) allows images that overlap any part of a window that is half the size of the maplet window. If 0.0 is chosen, the image must contain the landmark center.

If reject invisibles is chosen, the program uses the current shape to determine whether there is topography blocking the camera's view of the landmark center. Unless the object is bizarre such as Eros, choose 'n'.

IMPORTANT: When maplets are added in AUTOREGISTER they immediately populate the SUMFILE. The remaining processing must be carried out to eliminate unwanted ones, unlike LITHOS where added images do not stay added until the landmark file is updated.

The remaining processing steps are shown in the main menu below:

```
...     MAIN MENU     ...

0. Exit

REMOVE MAPLETS
a. Auto remove
n. Auto remove new only
m. Manual remove
p. Check peripheral visibility
o. Remove low-correlation lmks

IMAGE/MAPLET ALIGNMENT
1. Auto align
2. Manual align

LANDMARK ADJUSTMENTS
3. Repredict px/ln
4. Change flags
l. Change repredict limit
```

The first entry, 0, exits the processing of the current image and asks for a new one. Entering 'q' for the image name quits the process altogether.

The next block of options remove landmarks according to a variety of filters. If any of the first four are chosen a table is produced with the column headers:

```
#       = landmark number
LMKNM   = landmark name
EMISS   = emission angle
COV     = maplet-image overlap
RES     = image resolution/maplet scale
INV     = 1000 x invisible fraction of maplet
#PI!!   = number of images in maplet
```

Option a and n remove landmarks according to a filter: INVLIM: maximum fraction (in thousandths) of invisible points in the maplet according to the current topography. For example, an obliquely viewed maplet may have part of a crater bottom that can't be seen. An INVLIM of 27 represents 2.7%. SLIM: maximum emission angle. CLIM:

coverage limit. Minimum fraction of maplet covered by illuminated image data. RSMN: Minimum allowed ratio of image resolution (km/px) to maplet scale. RSMX: Maximum allowed ratio of image resolution (km/px) to maplet scale. The a option filters all landmarks and n filters only newly added ones. A display provided for the a and n options shows the number of images with resolutions from 0 to 3 times the maplet scale (column labels 00 - 30) and emission angles from 00 to 90 (rows). Option m allows the user to remove landmarks manually. Option p removes images in which part of the maplet is obscured by another part of the body. Option o eliminates images whose correlation with the illuminated maplet is less than a specified value. Option 1 must be run first to establish those correlation values.

The next menu block aligns extracted imaging data with the corresponding illuminated maplets. The EXTRACT_↩ DATA subroutines populate the landmark displays with image data projected onto the current maplet surface. We assume the maplet surfaces are correctly placed and oriented and have the correct topography. If the spacecract position and camera pointing were correct at the image exposure time, then all maplets would align. If not all the images align, then the amount of mis-alignment can be used to correct the spacecraft state. This process performs the alignment, updating the pixel/line image-space landmark positions from their predicted values. Option 2 can be used to align the problem image data to a maplet by hand. The process asks you to move the image window in pixels (+right) and lines (+ down) in order to align it to the maplet. For example, if image extraction looks like the upper display, we would match it to the maplet display on the bottom the by moving its window in the negative pixel and positive line directions:

| | | O | <- | | image display

X

| O | | | | X | maplet display | | |---------—|

All windows in AUTOREGISTER are 99x99 pixels. If a maplet has QSZ $<$ 49, as sometimes happens very early in the SP!!process, then there will be dark space surrounding it. If QSZ $>$ 49, then only the central portion of the maplet will be aligned. Option 2 is only used occasionally. Usually, the window shifts are done with the auto align option (1). This option first asks for a spacing, the size of the search area for the correlation. '1' searches a 5x5 pixel area, '2' a 10x10 and so on. After a correlation, the process will ask whether you want to change the spacing. Simply enter 'y' and then the new spacing. When you are satisfied with the correlation, choose 'n' for new spacing. You will see:

```
0.  continue.
1.  halve shifts.
2.  quarter shifts.
```

'0' will shift all the maplets by the amount determined by the correlation, '1' by half that amount and '2' by a quarter. Finally, the procedure will ask whether you want to accept the shift. A 'n' answer returns things to the starting values. For the larger search area the data is binned, so after alignment is reached, we should always go back and do it with a spacing of 1 again. A typical set of keystrokes in a script might be:

```
1  <- auto align
3  <- spacing 3
n  <- no spacing change
0  <- shift by full amount
y  <- keep shift
1  <- auto align
1  <- spacing 1
n  <- no spacing change
0  <- shift by full amount
y  <- keep shift
```

The correlation subroutine will produce a display like:

18 DJ0003 -0.009 -0.025 0.954 +++++ 19 DF0002 -0.017 0.007 0.890 +++++ 20 EF0001 -0.006 0.006 0.809 ++++

where columns 1 and 2 are landmark number and name, columns 3 and 4 are the predicted pixel/line shift predicted by the correlation, column 5 is the correlation value and the last column is a goodness of fit indicator that ranges from 0 to 5. If there is no correlation result at all, columns 3 and 4 have values 0.0000. This value, with the extra decimal place is recognized by some diagnosti!!programs.

Because the maplets have different resolutions, some may correlate well and some not at all. AUTOREGISTER only uses correlations greater than CORLIM, nominally set to 0.5, to determine the s/!!state. This new state can be

used to repredict the image-space landmark locations for all maplets using option 3 of the final menu block, with the result that subsequent correlations with option 1 will be much better. Option l of the final block can be used to choose a different value for CORLIM. The final selection in the last menu block, option 4, adjusts the topo flags on the PICNM entry of each landmark (.LMK) file. When AUTOREGISTER adds new landmarks, it does so with an $*$ on the PICNM record, indicating that that image is not to be used for topography. There are several sub-options to option 4, but the one usually used is 'b' that removes the $*$ from the PICNM record for all landmarks.

AUTOREGISTER can be run in a batch mode, following a script set up by make_scriptA, but this is rarely done. Usually, the parallel process AUTOREGISTERP and its script maker make_scriptAP fulfills this role.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.165.1.2 subroutine clear_data (integer *QSZ,* integer *N,* real$*$4, dimension(1000,100000) *DN*)

2.165.1.3 subroutine correlate (integer$*$4 *LMX,* integer$*$4 *NPIX,* integer$*$4 *QSZ,* real$*$4, dimension(1000,100000) *DN1,* character$*$6, dimension(lmx) *LMKID, X0, Y0, Z0*)

Variable_I/O.

Variable I/O Description

PICNM I User supplied image name. ANS I User supplied answer to multiple menus selections CORLIM I User supplied correlation limit for aligning images. INVLIM I SLIM I CLIM I RSMN I RSMX I XNAME I User supplied landmark name to be removed. Z1 I User supplied correlation threshold. K I User supplied landmark number to move. DX(K) I User supplied X pixels to shift. DY(K) I User supplied Y pixels to shift. S I User supplied spacing to be used. K O Landmark number. LMKID O Landmark ID. Z1 O Correlation thershold. DX O X pixel shift DY O Y pixel shift. COV O Covariance. J O Number of pictures containing landmark.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./SUMFILES/<PICNM>.SUM I File containing image's summary information. ./LMRKFILES/<LMKNM>.LMK I Files containing landmark data.

./LMRKLISTR.TXT I File containing list of land- marks to be used. ./LMRKLIST.TXT I File containing list of land-marks to be used if LMRKLISTR.TXT does not exist.A ./LMRK_DISPLAY1.gray I&O Landmark image gray scale file ./LMRK_DISPLAY1.pgm O Landmark image gray scale file converted to pgm image file.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called MAXLEN PICRES PXCOUNT SLEN

SPC_subroutines_called DEBLEMISH DELETEINLMK_PIC DELETEINPIC_LMK DN2PGM EXTRACT_DATA_P↩ IC FIND_LAMBDA_PIC FIND_PICFILE GET_UL HGT2SLP IMGPL2VN INSERTINLMK_PIC INSERTINPIC_LMK IPL2SCOBJPTG PICINPT PREDICT_DATA_PIC RAW2PGM READ_HEADER READ_MAP V2IMGPL VISIBLE

SPICELIB_functions_called RPD VDOT VNORM

SPICELIB_subroutines_called VADD VHAT VSCL VSUB

History 2012_01_10 Weight WN in IPL2SCOBJPTG replaced by separate weights for camera (WC) and s/!!(W↩

S). in INIT_LITHOS.TXT use: PICWTS= 1, 1, 1, 1, 1 WB, WL, WC, WS, WT 2012_04_06 subroutine get_model now includes albedo 2012_12_01 Image frame toggle removed so GET_MAPX no longer called. 2013_01_12 font now enters as include file 2013_08_07 SP, CP order corrected in FIND_LAMBDA_PI!!call 2013_09_03 Reads L↩ MRKLISTX.TXT instead of LMRKLIST.TXT to find new lmks. MAKE_LMRKLISTX must be run if landmarks have been added or deleted. 2013_09_15 DEBLEMISH now included in image extraction. 2013_10_01 Capitalization and header added. 2014_02_24 Q0 fixed at Q0=49. 2014_02_26 0 spacing in CORRELATE asks for new input. 2014_03_07 Existence test for INIT_LITHOS.TXT removed. 2014_03_09 Prettied up menu. 2014_03_10 COV redefined.

2.165.1.4 subroutine correlate_lmk (real∗4, dimension(1000,100000) *DN1,* integer∗4 *QSZ,* integer∗4 *I0,* integer∗4 *J0, S, X0, Y0, Z0,* logical *FOUND*)

2.165.1.5 subroutine display (real∗4, dimension(1000,100000) *DN,* integer *QSZ,* integer *N*)

2.166 LITHOSPHERE/AUTOREGISTERP.f File Reference

Functions/Subroutines

- program __autoregisterp.f__
  Procedure AUTOREGISTERP.
- subroutine correlate (LMX, NPIX, QSZ, DN1, LMKID, X0, Y0, Z0)
  Variable_I/O.
- subroutine correlate_lmk (DN1, QSZ, I0, J0, S, X0, Y0, Z0, FOUND)
- subroutine clear_data (QSZ, N, DN)
- subroutine display (DN, QSZ, N)

2.166.1 Function/Subroutine Documentation

2.166.1.1 program __autoregisterp.f__ ( )

Procedure AUTOREGISTERP.

Abstract

This procedure aligns existing maplets with imaging data from a single image and uses subroutine IPL2SCOBJPTG to update the camera pointing and spacecraft-object vector in the corresponding SUMFILE. The program uses the file LMRKLISTX.TXT to pre- screen the maplets, so if maplets have been added or deleted recently, the procedure MAKE_LMRKLISTX should be run.

The procedure is designed to be run in a parallel batch mode. It uses lockout files to prevent two processes from reading or writine to a landmark file at the same time. Some of the menu options below are never used in the batch mode although they are fully functional if it is desired to run AUTOREGISTERP in interactive mode.

The program AUTOREGISTERP first asks for a two character USR name to distinguish between processes. It then asks for a picture name. It produces a list of existing landmarks, if any, and asks if more should be added. An initial filter for added images is set in INIT_LITHOS.TXT. RESLM is the maximum ratio of the image resolution to the maplet scale, while SIZLM is the maximum ratio of the linear maplet size to the image size. NUMLM sets the minimum number of maplets found in the image before the other two filters come into play. The user is asket for two more inputs:

The fractional width=0.5 (what I usually use) allows images that overlap any part of a window that is half the size of the maplet window. If 0.0 is chosen, the image must contain the landmark center.

If reject invisibles is chosen, the program uses the current shape to determine whether there is topography blocking the camera's view of the landmark center. Unless the object is bizarre such as Eros, choose 'n'.

IMPORTANT: When maplets are added in AUTOREGISTERP they immediately populate the SUMFILE. The remaining processing must be carried out to eliminate unwanted ones, unlike LITHOS where added images do not stay added until the landmark file is updated.

The remaining processing steps are shown in the main menu below:

```
...     MAIN MENU     ...

0. Exit

REMOVE MAPLETS
a. Auto remove
n. Auto remove new only
m. Manual remove                  <- only in interactive mode
p. Check peripheral visibility
o. Remove low-correlation lmks

IMAGE/MAPLET ALIGNMENT
1. Auto align
2. Manual align                   <- only in interactive mode

LANDMARK ADJUSTMENTS
3. Repredict px/ln
4. Change flags
l. Change repredict limit         <- only in interactive mode
```

The first entry, 0, exits the processing of the current image and asks for a new one. Entering 'q' for the image name quits the process altogether.

The next block of options remove landmarks according to a variety of filters. If any of the first four are chosen a table is produced with the column headers:

```
#       = landmark number
LMKNM   = landmark name
EMISS   = emission angle
COV     = maplet-image overlap
RES     = image resolution/maplet scale
INV     = 1000 x invisible fraction of maplet
#PI!!   = number of images in maplet
```

Option a and n remove landmarks according to a filter: INVLIM: maximum fraction (in thousandths) of invisible points in the maplet according to the current topography. For example, an obliquely viewed maplet may have part of a crater bottom that can't be seen. An INVLIM of 27 represents 2.7%. SLIM: maximum emission angle. CLIM: coverage limit. Minimum fraction of maplet covered by illuminated image data. RSMN: Minimum allowed ratio of image resolution (km/px) to maplet scale. RSMX: Maximum allowed ratio of image resolution (km/px) to maplet scale. The a option filters all landmarks and n filters only newly added ones. A display provided for the a and n options shows the number of images with resolutions from 0 to 3 times the maplet scale (column labels 00 - 30) and emission angles from 00 to 90 (rows). Option m allows the user to remove landmarks manually. Option p removes images in which part of the maplet is obscured by another part of the body. Option o eliminates images whose correlation with the illuminated maplet is less than a specified value. Option 1 must be run first to establish those correlation values.

The next menu block aligns extracted imaging data with the corresponding illuminated maplets. The EXTRACT_↩ DATA subroutines populate the landmark displays with image data projected onto the current maplet surface. We assume the maplet surfaces are correctly placed and oriented and have the correct topography. If the spacecract position and camera pointing were correct at the image exposure time, then all maplets would align. If not all the images align, then the amount of mis-alignment can be used to correct the spacecraft state. This process performs the alignment, updating the pixel/line image-space landmark positions from their predicted values. Option 2 can be used to align the problem image data to a maplet by hand. The process asks you to move the image window in pixels (+right) and lines (+ down) in order to align it to the maplet. For example, if image extraction looks like the upper display, we would match it to the maplet display on the bottom the by moving its window in the negative pixel and positive line directions:

| | | O | <- | | image display

X

| O | | | | X | maplet display | | |---------—|

All windows in AUTOREGISTERP are 99x99 pixels. If a maplet has QSZ $<$ 49, as sometimes happens very early in the SP!!process, then there will be dark space surrounding it. If QSZ $>$ 49, then only the central portion of the maplet will be aligned. Option 2 is only used occasionally. Usually, the window shifts are done with the auto align option (1). This option first asks for a spacing, the size of the search area for the correlation. '1' searches a 5x5 pixel area, '2' a 10x10 and so on. After a correlation, the process will ask whether you want to change the spacing. Simply enter 'y' and then the new spacing. When you are satisfied with the correlation, choose 'n' for new spacing. You will see:

```
0.  continue.
1.  halve shifts.
2.  quarter shifts.
```

'0' will shift all the maplets by the amount determined by the correlation, '1' by half that amount and '2' by a quarter. Finally, the procedure will ask whether you want to accept the shift. A 'n' answer returns things to the starting values. For the larger search area the data is binned, so after alignment is reached, we should always go back and do it with a spacing of 1 again. A typical set of keystrokes in a script might be:

```
1  <- auto align
3  <- spacing 3
n  <- no spacing change
0  <- shift by full amount
y  <- keep shift
1  <- auto align
1  <- spacing 1
n  <- no spacing change
0  <- shift by full amount
y  <- keep shift
```

The correlation subroutine will produce a display like:

18 DJ0003 -0.009 -0.025 0.954 +++++ 19 DF0002 -0.017 0.007 0.890 +++++ 20 EF0001 -0.006 0.006 0.809 ++++

where columns 1 and 2 are landmark number and name, columns 3 and 4 are the predicted pixel/line shift predicted by the correlation, column 5 is the correlation value and the last column is a goodness of fit indicator that ranges from 0 to 5. If there is no correlation result at all, columns 3 and 4 have values 0.0000. This value, with the extra decimal place is recognized by some diagnosti!!programs.

Because the maplets have different resolutions, some may correlate well and some not at all. AUTOREGISTERP only uses correlations greater than CORLIM, nominally set to 0.5, to determine the s/!!state. This new state can be used to repredict the image-space landmark locations for all maplets using option 3 of the final menu block, with the result that subsequent correlations with option 1 will be much better. Option l of the final block can be used to choose a different value for CORLIM. The final selection in the last menu block, option 4, adjusts the topo flags on the PICNM entry of each landmark (.LMK) file. When AUTOREGISTERP adds new landmarks, it does so with an $*$ on the PICNM record, indicating that that image is not to be used for topography. There are several sub-options to option 4, but the one usually used is 'b' that removes the $*$ from the PICNM record for all landmarks.

AUTOREGISTERP is almost always run in a batch mode, following a script set up by make_scriptAP.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.166.1.2 subroutine clear_data (integer *QSZ,* integer *N,* real$*$4, dimension(1000,100000) *DN*)

2.166.1.3 subroutine correlate (integer$*$4 *LMX,* integer$*$4 *NPIX,* integer$*$4 *QSZ,* real$*$4, dimension(1000,100000) *DN1,* character$*$6, dimension(lmx) *LMKID, X0, Y0, Z0*)

Variable_I/O.

Variable I/O Description

PICNM I User supplied image name. ANS I User supplied answer to multiple menus selections CORLIM I User supplied correlation limit for aligning images. INVLIM I SLIM I CLIM I RSMN I RSMX I XNAME I User supplied landmark name to be removed. Z1 I User supplied correlation threshold. K I User supplied landmark number to move. DX(K) I User supplied X pixels to shift. DY(K) I User supplied Y pixels to shift. S I User supplied spacing to be used. K O Landmark number. LMKID O Landmark ID. Z1 O Correlation thershold. DX O X pixel shift DY O Y pixel shift. COV O Covariance. J O Number of pictures containing landmark.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./SUMFILES/<PICNM>.SUM I File containing image's summary information. ./LMRKFILES/<LMKNM>.LMK I Files containing landmark data.

./LMRKLISTR.TXT I File containing list of land- marks to be used. ./LMRKLIST.TXT I File containing list of land-marks to be used if LMRKLISTR.TXT does not exist.A ./LMRK_DISPLAY1.gray I&O Landmark image gray scale file ./LMRK_DISPLAY1.pgm O Landmark image gray scale file converted to pgm image file.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called DN2PGM PICRES_LO PXCOUNT_LO SLEN

SPC_subroutines_called DEBLEMISH DELAY DELETEINLMK_PIC_LO DELETEINPIC_LMK DN2PGM EXTRA↩ CT_DATA_PIC FIND_LAMBDA_PIC FIND_PICFILE GET_UL HGT2SLP IMGPL2VN INSERTINLMK_PIC_LO I↩ NSERTINPIC_LMK IPL2SCOBJPTG_LO PICINPT PREDICT_DATA_PIC RAW2PGM READ_HEADER READ_↩ MAP V2IMGPL VISIBLE_LO

SPICELIB_functions_called RPD VDOT VNORM

SPICELIB_subroutines_called VADD VHAT VSCL VSUB

History

2012_01_10 Weight WN in IPL2SCOBJPTG replaced by separate weights for camera (WC) and s/!!(WS). in IN↩ IT_LITHOS.TXT use: PICWTS= 1, 1, 1, 1, 1 WB, WL, WC, WS, WT 2013_01_12 font now enters as include file 2013_08_07 Image frame toggle removed so GET_MAP no longer called. 2013_08_07 SP, CP order corrected in FIND_LAMBDA_PI!!call 2013_09_03 Reads LMRKLISTX.TXT instead of LMRKLIST.TXT to find new lmks. MA↩ KE_LMRKLISTX must be run if landmarks have been added or deleted. 2013_09_15 DEBLEMISH now included in image extraction. 2013_10_03 Capitalization and header added. 2014_02_01 Bug fix: ESZ set. 2014_02_↩ 01 Display of image commented out. 2014_02_24 Q0 fixed at Q0=49. 2014_03_10 0 spacing in CORRELATE asks for new input. 2014_03_10 Existence test for INIT_LITHOS.TXT removed. 2014_03_10 Prettied up menu. 2014_03_10 COV redefined.

2.166.1.4 subroutine correlate_lmk (real∗4, dimension(1000,100000) *DN1,* integer∗4 *QSZ,* integer∗4 *I0,* integer∗4 *J0, S, X0, Y0, Z0,* logical *FOUND*)

2.166.1.5 subroutine display (real∗4, dimension(1000,100000) *DN,* integer *QSZ,* integer *N*)

2.167 LITHOSPHERE/BLEMISHES.f File Reference

Functions/Subroutines

- program __blemishes.f__

  Proceduren BLEMISHES.

2.167.1 Function/Subroutine Documentation

2.167.1.1 program __blemishes.f__ ( )

Proceduren BLEMISHES.

Abstract

Thie procedure creates a blemish (.BLM) file for an image that lives in the BLEMISHES subdirectory. There are two kinds of blemish handled here:

The block option (b) masks a square block of the image specified by a minimum and maximum pixel and line. This type of blemish is typically used for missing or hashy lines due to downlink errors.

The spot option (s) masks a small region around a pixel/line center (p,l) from p-k to p+k and l-k to l+k. An additional input is a brightness threshold. If this is chocen as zero, everything in the spot will be masked. However, if there is a bright blemish such as a cosmi!!ray hit in the spot, the threshold can be set to remove only the affected pixls. In the latter case, a large region (large k) can be specified and all the bright blemishes in the region masked without affecting the rest of the data.

In the event that a camera has bad pixels common to all images, a template can be created that masks these pixels in all that cmera's images. The template is specified in the INIT_LITHOS.TXT file with the key word BLEMIS↩ H. For example, DAWN has two framing cameras with image names starting with FC1 and FC2. FC2 images all have several blemishes in the same places. A template in INIT_LITHOS specified by BLEMISH='FC2#########' will correct all of these mages without having thousands of individual blemish files. If an image contains additional blemishes it can have its own file with just those additional blemishes.

In order to create a TEMPLATE, a blemish file for one of the affected images is made with only the common bad areas masked. That file can be saved or not (the procedure will ask: 'Save blemish file? (y/n)'), and then the user will be asked 'Create/change template file? (y/n)'. If 'y' is chosen the user will be asked for a 12 character template name with the common characters of the image names in their proper positions and # everywhere else. If the PI↩ CNM used to make the original blemish file is not consistent with the template, the procedure will say so and STOP. Otherwise, it will craeate the template file and, if necessary, remind the user to define it in INIT_LITHOS.TXT.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.168 LITHOSPHERE/DYNAMICS.f File Reference

Functions/Subroutines

- program __dynamics.f__

  Procedure DYNAMICS.

2.168.1 Function/Subroutine Documentation

2.168.1.1 program __dynamics.f__ ( )

Procedure DYNAMICS.

Abstract

DYNAMICS augments the nominals file by appending the nominal inertial space position differences dW=W1-↩ W0 between the current image and up !! to two images before and after. This information is used to constrain the solution in the subroutine IPL2SCOBJPTG so that the s/!!position solution does not deviate too much from a dynamically realisti!!state. An entry ETLIM= in the input DYNAMICS.TXT gives a maximum time difference in seconds that limits the neighboring images to be used, and an estimate of the velocity uncertainty that goes into an uncertainty estimate also include in the added records. A typical record is:

```
ETLM=  1800, 1.D-6
```

A record FRAME= tells IPL2SCOBJPTG how to interpret the sigmas in spacecrafe position found in the nominals file. The choices are

```
FRAME= ['dxR_FRAME', 'Dxr_FRAME', 'BOD_FRAME']
```

The first one uses tha radial direction (R) as prime for the third component with the second component being the cross track direction (X) and the first component, roughly in the downtrack direction completing the right-handed coordinate system. This is used in orbital operations around large bodies, where the radial component is well known from from the Doppler data. The second choice is used during approach where the downtrack velocity is best known. The "radial" (impact parameter) direction is crossed with the downtrack to give the crosstrack direction. The third option simply uses the components in the body-fixed frame, generally with equal uncertainties in each direction.

A record VSIG= allows the user to change the position sigmas from those in the original nominals file. The record:

```
VSIG= 0.100, 0.100, 0.040
```

sets the uncertainties in the three components to 100 m, 100 m and 40 m respectively. If, during another mission phase, these sigmas change, another such record would be added to affect subsequent nominals files.

A record PSIG= allows the user to change the pointing sigmas in the nominal files. The three components refer to rotations about the three camera axes. For OSIRIS_REx we expect something like:

```
PSIG= 0.0003, 0.0003, 0.0005
```

with a larger twist uncertainty since there is only one star tracker.

The DYNAMICS.TXT file looks just like the PICTLIST.TXT file in that there is a space in the first column. Excerpts from such a file for !! DAWN at Vesta are:

```
FRAME='BOD_FRAME'      APPROACH
ETLM=  1800, 1.D-6
VSIG= 0.200, 0.200, 0.200
PSIG= 0.0001, 0.0001, 0.0001
 FC11A0001225
 FC11A0001241
 ...
 FC21A0003895
 FC21A0003910
FRAME='dxR_FRAME'      SURVEY
ETLM=  1800, 1.D-6
VSIG= 0.100, 0.100, 0.040
PSIG= 0.0001, 0.0001, 0.0001
 FC21A0003931
 FC21A0003932
 ...
 FC21A0032347
 FC21A0032348
END
```

There were a number of parameter changes as the mission porgressed from Approach to Survey to HAMO to LAMO to HAMO2 to Departure, with transitions between phases requiring parameters of their own.!!

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.169 LITHOSPHERE/GEOMETRY.f File Reference

Functions/Subroutines

- program __geometry.f__

  Procedure GEOMETRY.
- subroutine solve_vlm

  Variable_I/O.
- subroutine solve_scobjptg (PFLAG)

  .....

2.169.1 Function/Subroutine Documentation

2.169.1.1 program __geometry.f__ ( )

Procedure GEOMETRY.

Abstract

The procedure GEOMETRY is a wrapper that iterates the IPL2SCOBJPTG and IPL2VLM subroutines sequentially. The inputs are a list of operations where 1=IPL2VLM, 2=IPL2SCOBJPTG, 0=END, the number of iterations desired, a choice of whether to use limbs in determining camera pointing (y/n), and a option to do it all over again (y/n). For example,

```
  GEOMETRY
  120           <- do 1 followed by 2
  30            <- do them 30 times
  y             <- use limbs for pointing
  n             <- stop when done

  GEOMETRY
  20            <- do 2 only
  10            <- do it 10 times
  n             <- don't use limbs for pointing
  n             <- stop when done

The default is to do these operations for all landmarks in
LMRKLIST.TXT and all images in PICTLIST.TXT.  If INIT_LITHOS
contains a record

 GEOPIC='filename1'

or

 GEOMAP='filename2'

then the files used are reduced - filename1 instead of PICTLIST and
filename2 instead of LMRKLIST.

A third option has been commented out since it is not currently
operational.  This is the RA/DEC/TWIST option that computes the
transformation from inertial to body-fixed frame for each image.
This is part of the wobble study that is currently ongoing. Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies
      with imaging data", Meteoritics & Planetary Science 43,
      Nr 6, 1049-1061 (2008)
```

Declarations

2.169.1.2 subroutine solve_scobjptg (character*1 *PFLAG*)

.....

Parameters

pflag	.....

2.169.1.3 subroutine solve_vlm ( )

Variable_I/O.

Variable I/O Description

ENTRY I Menu item selection for: 1: landmark vectors. 2: camera pointing, scobj 3: RA/DEC/TWIST 0: end. NITER I User supplied number of iterations. PFLAG I User supplied use limbs for pointing flag. ANS I User supplied answer to menu selection.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!>toolkit programs. ./LMRKLIST.TXT I File containing a list of landmarks. ./PICTLIST.TXT I File containing a list of images.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called None

SPC_subroutines_called LPL2SCOBJPTG LPL2VLM SOLVE_RDT

SPICELIB_functions_called None

SPICELIB_subroutines_called RESLUN

History 2012_01_09: Weight WN in IPL2SCOBJPTG replaced by separate weights for camera (WC) and s/!>(W↩S). in INIT_LITHOS.TXT use: PICWTS= 1, 1, 1, 1, 1 WB, WL, WC, WS, WT 2012_01_15: If weight W↩T = 0 in IPL2SCOBJPTG trajectory loop ignored. 2012_01_16: IPL2VLM changed to remove map overlaps from sigma_lmk determination 2012_01_16: IPL2SCOBJ changed to remove image overlaps from sigma_↩vso determination 2012_10_03: PICTLIST.TXT is default for SOLVE_SCOBJPTG. If PICTLISTS.TXT is desired use GEOPIC='PICTLISTS.TXT' in INIT_LITHOS.TXT 2012_10_03: LMRKLIST.TXT is default for SO↩LVE_VLM. If LMRKLISTR.TXT is desired use GEOMAP='LMRKLISTR.TXT' in INIT_LITHOS.TXT 2013_07↩_30: Header added, capitalization & compiler warnings fixed. 2014_06_02: Command line Version option added. FIND OUT HOW MANY ARGUMENTS GIVEN ON THE COMMAND LINE = IARGC() ARGUM↩ENT -V PROVIDES VERSION NUMBER WRITE(6,∗) ' 3: RA/DEC/TWIST ' IF(ENTRY(K:K).EQ.'3') THE↩N CALL SOLVE_RDT ENDIF ..... PROGRAM SPECIFI!>SUBROUTINE↩S ....

2.170 LITHOSPHERE/LIMBER.f File Reference

Functions/Subroutines

- program __limber.f__
- real ∗8 function brt0 (z, npx, nln, dn)
- logical function bodpl (SHAPEFILE, PICNM, IMGPL)
- logical function mmon (SHAPEFILE, MM, MMFL, V0, CX, CY, CZ)

2.170.1 Function/Subroutine Documentation

2.170.1.1 program __limber.f__ ( )

2.170.1.2 logical function bodpl (character∗72 *SHAPEFILE,* character∗12 *PICNM,* double precision, dimension(2) *IMGPL*)

2.170.1.3 real∗8 function brt0 (real∗8, dimension(2) *z,* integer∗4 *npx,* integer∗4 *nln,* integer∗4, dimension(4096,4096) *dn*)

2.170.1.4 logical function mmon (character∗72 *SHAPEFILE,* double precision, dimension(2) *MM,* double precision *MMFL,* double precision, dimension(3) *V0,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ*)

2.171 LITHOSPHERE/LITHOS.f File Reference

Functions/Subroutines

- program __lithos.f__

  Procedure LITHOS.
- subroutine correlate (PMX, NPIX, QSZ, NSZ, DN1, PICID, HIDE, X0, Y0, Z0)

  Variable_I/O.
- subroutine correlate_pic (DN1, QSZ, I0, J0, S, X0, Y0, Z0, FOUND)
- subroutine clear_data (QSZ, NPIX, DN1)

2.171.1 Function/Subroutine Documentation

2.171.1.1 program __lithos.f__ ( )

Procedure LITHOS.

Abstract

This procedure is the main toolkit for the creation and refinement of landmarks and their associated maplets. A typical landmark file, CK0008.LMK, looks like:

CK0008 T 49 0.5000000 SIZE, SCALE(KM) -1 -1 -1 -1 HORIZON 0.2500000000D+00 0.2672841486D+00 S↩
IGKM, RMSLMK -0.1151582350D+03 -0.5701086949D+02 0.1512589957D+03 VLM -0.6234954596D+00 -0.↩
5246677995D+00 -0.5796353221D+00 UX 0.6024240851D+00 -0.7949613929D+00 0.7156561315D-01 UY -
0.4983358979D+00 -0.3045654297D+00 0.8117273450D+00 UZ 0.1559420789D+00 0.1046028345D+00 0.↩
7422983579D-01 SIGMA_LMK PICTURES N1644784749 786.74 820.27 <- lmk px/ln in image N1644785949
307.69 851.76N1675158257 602.02 730.42 MAP OVERLAPS BK0002 19.309 -1.587 -1.570 <- (V-V').(UX',UY',U↩
Z')/SCALE' CK0006 -66.478 6.974 -6.603 CK0009 -2.082 44.078 -4.142 BK0003 3.661 -43.752 -2.212 LIMB FITS
N1558945693 596.88 615.60 0.02 <- lmk px/ln, sigma from limb N1558946626 578.68 616.94 0.02W1644778567
604.73 555.56 0.07 END FILE

```
The first nine lines provide information about the landmark size, scale,
position and orientation.  HORIZON is an unused line, since we are no
longer using that functionality.  In the next line, SIGKM is always half
the scale and RMSLMK (RMS px/ln residual) is not used in LITHOS.  VLM is
the body-fixed landmark vector determined from GEOMETRY and SIGMA_LMK are
the formal uncertainties in each of the components.  UX, UY and UZ are the
unit vectors defining the orientation of the landmark reference plane.

The next block of data, PITCURES, contains the px/ln location of the
landmark in images.  These are initially set using menu option 1 by hand,
then by cross correlation among extracted image data, then by correlation
with a constructed landmark template (menu option 0) and finally by
correlation with the landmark maplet (menu option 2).  It is these
pixel/line locations that are used by GEOMETRY to determine the landmark
vector VLM.  The next two blocks, MAP OVERLAPS and LIMB FITS, are
determined only after a maplet has been constructed and contain
information about the relative locations of neighboring landmarks and the
pixel/line landmark locations in images where the maplet spans part of the
lit limb.  These data also contribute to the determination of landmark
```

```
location by GEOMETRY.

The main menu for LITHOS is shown below.  It is divided into a number of
processes, each documented separately.  The documentation for process P,
for example, can be found by searching on P^.

...     MAIN MENU     ...             For process abstract:

 Q. Quit LITHOS                              search Q^

LANDMARK/MAPLET CONSTRUCTION
 0. Find template                            search 0^
 1. Align landmarks                          search 1^
 2. Find heights                             search 2^
 O. Attach map to maps or limbs              search O^

LANDMARK/MAPLET I/O AND CREATION
 I. Input landmark                           search I^
 U. Update landmark files                    search U^
 C. Create new landmark                      search C^
 R. Replicate or Rename landmark             search R^
 S. Change scale, qsz or orientation         search S^
 G. Turn on/off rename                       search G^

ADJUST INPUT DATA AND NOMINALS
 N. Find normal                              search N^
 V. Find V, Z or PTG                         search V^
 A. Reset albedo or slopes                   search A^
 M. Get heights from shape model             search M^
 B. Get heights from surrounding map         search B^
 X. Turn on/off extract filter               search X^

DELETE, ELIMINATE OR IGNORE FILES
 D. Delete or Disconnect landmark            search D^
 E. Eliminate pictures from landmark         search E^
 P. Picture status                           search P^
 L. Turn on/off picture restriction          search L^

INFORMATION AND DISPLAY
 F. Find maplets containing surface point    search F^
 Z. Use zoom display                         search Z^
 H. Hide/Show screen output                  search H^

LITHOS can be run in batch mode in order to process a set of landmarks.  First, a
list of landmark names ending with END is saved as make_script.in, for example:

CK0001
CK0002
CK0003
CK0004
CK0005
CK0006
CK0007
CK0008
CK0009
CL0001
CL0002
CL0003
END

Then one of the appropriate seeds in the src/SEEDS directory,
Fiterate1.seed for example, is copied to make_scriptF.seed.  This will
tell each process what to do with the landmarks it is refining.  Unless the
landmarks have been chosen so that their overlaps lie in a specified region, we
should make sure that LMKLISTO.TXT does not exist.  In order to carry out the
iteration, the following script is copied into the terminal window:

./rem_script.b
rm -f ./TESTFILES/*          < Remove old inputs and clear output directories.
rm -f ./TESTFILES1/*
make_scriptF                 < Set up batch script.
chmod +x run_script.b        < Enable batch scritp.
./run_script.b &             < Run batch script.
```

```
find_nofit should be run from time to time to monitor the progress and to
see if any landmark needs extra work.  The TESTFILES directory will
accumulate maximally stretched landmark displays that can be viewed.  The
screen output of each of the LITHOS landmark processes will be saved as
LMKNM.OOT, a text file that cah be examined.  The hide option in the main
menu is included in the seed scripts to minimize the size of these files.

The most common batch process involving LITHOS is the tiling of an already
constructed (with BIGMAP) piece of surface with higher resolution maplets.  The
bigmap, which we usually call XXXXXX, is created at a resolution (km/px) of 1.25
times the maplet resolution.  XXXXXX is tiled with maplets centered every 50
pixels and lines using a setup script called make_scriptT.in with the form:

XXXXXX                  <- Bigmap name
XXX040.SEED             <- LITHOS script
      50   50           <- pixel/line of a tile
     100   50
     150   50
     200   50
    #250   50           <- This tile will not be made
     300   50
     350   50
      50  100
      ......
     200  350
     250  350
     300  350
     350  350
END

When BIGMAP is run, it keeps track of all the maplets that went into its
constructioncp in a file USED_MAPS.TXT.  If we

cp USED_MAPS.TXT LMRKLISTO.TXT

then LMRKLISTO.TXT will contain all maplets that could possibly overlap one of the
new ones.  Moreover, as LITHOS creats new maplets, these are automatically added
to  LMRKLISTO.TXT.  The script, in this case XXX040.SEED, creates a landmark (C^),
turns on rename (G^), inputs the landmark (I^), initializes the landmark with
topography and albedo from XXXXXX (B^), sets its local coordinate frame (N^), runs
a filter (E^) to eliminate images with too low resolution, too high emission
angle, possible invisibilities and insufficient coverage, and aligns the extracted
image data with the nominal maplet (0^).  The rest of the script looks like
Fiterate1.seed of the iterative batch above.  In order to carry out the tiling,
the following script is copied into the terminal window:

./rem_script.b
rm -f ./TESTFILES/*          < Remove old inputs and clear output directories.
rm -f ./TESTFILES1/*
make_scriptT                 < Set up batch script.
chmod +x run_script.b        < Enable batch scritp.
./run_script.b &             < Run batch script.

find_nofitT should be run from time to time to monitor the progress and to
see if any landmark needs extra work.  The TESTFILES directory will
accumulate maximally stretched landmark displays that can be viewed.  The
screen output of each of the LITHOS landmark processes will be saved as
LMKNM.OOT, a text file that cah be examined.  The hide option in the main
menu is included in the seed scripts to minimize the size of these files.

IMPORTANT: if find_nofitT finds maplets that need work, the user should clean
these up with LITHOS before exporting because the EXPORT/IMPORT process will
generally change the names of maplets.

When there are mnay maplets to be made, the process above is often farmed out to
several other machines.  In that case, we should rm -f LMRKlIST1.TXT before
beginning.  This file accumulates all the new maplets and, after te run, the
external user follows the procedures in EXPORT_TILES.TXT to produce a .tar file
that can be sent to the home computer.  The procedure in IMPORT_TILES.TXT will set
up a LITHOS script to import the new maplets into the central directory.

The MESSENGER spacecraft has a wide angle camera with 12 filters.  Since only the
```

```
clear filter is capable of being calibrated with star observations, and very few
unsaturated clear filter images of Mercury have been taken, all of WA!!images are
being used in order to calibrate all the filters in terms of the clear.  This can
lead to a large number of images in some landmarks, so a "pruning" option was
introduced.  If a maplet uses more than a certain number of images PRNLM, then
some "ignorable" images can be randomly removed in the p option of the E process
of LITHOS.  Ignorable images are set in INIT_LITHOS.TXT by a template of the form
IGNORE='W##########C' that ignores wide angle filter !!images.  Images will be
removed only if they are lower resolution than an IGNLM, however.  These extra
images provide duplication in slope/albedo estimation (0^) and in the differential
stereo option of the slope tpo height process (2^).  Any ignorable image will not
participate in those processes if the landmark has more than IGNUM images and the
image is lower resolution than an IGNLM.  The parameters PRNLM, IGNUM AND IGNLM
are all set in INIT_LITHOS.TXT.
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.171.1.2 subroutine clear_data (integer *QSZ,* integer *NPIX,* real∗4, dimension(5000,5000) *DN1*)

2.171.1.3 subroutine correlate (integer∗4 *PMX,* integer∗4 *NPIX,* integer∗4 *QSZ,* integer∗4 *NSZ,* real∗4, dimension(5000,5000) *DN1,* character∗12, dimension(pmx) *PICID,* logical *HIDE, X0, Y0, Z0*)

Variable_I/O.

Variable I/O Description

USR I ??? read in from INIT_LITHOS.TXT ASIG I ??? read in from INIT_LITHOS.TXT SSIG I ??? read in from INIT_LITHOS.TXT CHI0 I ??? read in from INIT_LITHOS.TXT SEED I ??? read in from INIT_LITHOS.TXT BO↩ DY I ??? read in from INIT_LITHOS.TXT BLIM I ??? read in from INIT_LITHOS.TXT RESLM I ??? read in from INIT_LITHOS.TXT PRNLM I ??? read in from INIT_LITHOS.TXT IGNLM I ??? read in from INIT_LITHOS.TXT IGNUM I ??? read in from INIT_LITHOS.TXT NOFIT I ??? read in from INIT_LITHOS.TXT NUMLM I ??? read in from INIT_LITHOS.TXT SIZLM I ??? read in from INIT_LITHOS.TXT PICLM I ??? read in from INIT_LITH↩ OS.TXT RECVR I ??? read in from INIT_LITHOS.TXT ALPAD I ??? read in from INIT_LITHOS.TXT RENAME I ??? read in from INIT_LITHOS.TXT RPLUSE I ??? read in from INIT_LITHOS.TXT REGFLG I ??? read in from INIT_LITHOS.TXT REG I ??? read in from INIT_LITHOS.TXT KB I ??? read in from INIT_LITHOS.TXT DENO↩ ISE I ??? read in from INIT_LITHOS.TXT IGNORE I ??? read in from INIT_LITHOS.TXT ANS I User input from multiple menu selection items NAME I User supplied landmark name. QSZ I Landmark size read from landmark file SCALE I Landmark scale (pix/km) from landmark file IHMN I Landmark I horizon minimum from landmark file INMX I Landmark I horizon maximum from landmark file JHMN I Landmark J horizon minimum from landmark file JHMX I Landmark J horizon maximum from landmark file SIG_KM I ???? from landmark file RMS_LMK I ???? from landmark file V(3) I Landmark distance from body center from landmark file UX(3) I Landmark X normal unit vector from landmark file UY(3) I Landmark Y normal unit vector from landmark file UZ(3) I Landmark Z normal unit vector from landmark file SIGMA_LMK I ???? from landmark file PICID I Image name read from PICTURE section of the landmark file IPL I Image pixel and line number read from PICTURE section of the landmark file. PFLAG I Image use flag read from PICTURE section of the landmark file. XNAME I Image name read from PICTLIST.TXT file. NPX I Number of image pixels from ./SUMFILES/<PICNM>.SUM NLN I Number of image lines from ./SUMFILES/<↩ PICNM>.SUM V(3) I Distance from S/!!to target center. CX(3) I Camera X unit pointing vector. CY(3) I Camera Y unit pointing vector. CZ(3) I Camera Z unit pointing vector. SZ(3) I Sun pointing vector. LMRKOLD I User supplied landmark name to replicate. LMRKNEW I User supplied landmark name to be replicated.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./LMKFILES/<LMKNM>.LMK I ./PICTLISTRX.TXT I File containing list of images ./PICTLISTX.TXT I File containing list of images to be used if ./PICTLISTRX.TXT does not exist. ./PICTLIST.TXT I File containing list of images to be used if ./PICTLISTX.TXT does not exist. ./SUMFILES/<PICNM>.SUM I Image summary file for <PICNM> ./MAPFILES/<LMKNM>.MAP I Landmark map file for landmark <LMKNM> ./OVERLAPS.TXT I ???????????? ./LMRKLIST.TXT I List of target

landmarks ./MAPFILES/<NAME>.MAP I Landmark map file for landmark <NAME> ./BIGLIST.TXT I ????????? ./SUMFILES/<PICID>.SUM I Image summary file for <PICID> ./NOMINALS/<PICNM>.NOM I Image nominal pointing file for <PICID> ./tmpl.gray O Temp gray scale map file. ./tmpl.pgm O Temp pgm map file converted by RAW ./LMKFILEs/<NAME>.LMK I ./MAPFILES/<NAME>.MAP I Landmark file where <NAME> is from OVE↩ RLAPS.TXT ./seeds.gray O Temp gray scale map file ./seeds.pgm O Temp pgm map file converted by RAW2PGM ./LMRK_DISPLAY0.gray O Temp gray scale map file. ./LMRK_DISPLAY0.pgm O Temp pgm map file converted by RAW ./LMRK_DISPLAY1.gray O Temp gray scale map file. ./LMRK_DISPLAY1.pgm O Temp pgm map file converted by RAW

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called ILLUM LMCOUNT MAXLEN PXCOUNT RANN SLEN

SPC_subroutines_called ATTACH COMPARE CREATE_LMFILE DEBLEMISH DELETE_LMFILE DELETEINL↩ MK_LMK DELETEINLMK_PIC DELETEINPIC_LIM DELETEINPIC_LMK DENOISE DETACH DISCONNECT_↩ LMFILE DISPLAY1 EXTRACT_DATA EXTRACT_DATA_PIC EXTRACT_GRAD FIND_ALBEDO FIND_LAMBDA FIND_PICFILE FIND_SLOPES GET_HEIGHTS GET_MAP HGT2SLP IMGPL2VN INSERTINPIC_LIM INSER↩ TINPIC_LMK INSERTINPIC_PTG INSERTINPIC_SIG INSERTINPIC_SOV INSERTINPIC_VFLG INVERTN IP↩ L2RDT IPL2SCOBJPTG IPL2VLM LIMB_HEIGHTS LOC2PIX ORIENT OVERLAPS PATCH_COORDS PICINPT POINT2MAPS PREDICT_DATA PXMM RAW2PGM READ_HEADER READ_MAP REPLICATE_LMFILE SHO↩ W_SLOPES SLP2HGT STEREO TUCK U2VN V2IMGPL VISIBLE WRITE_MAP ZOOM

SPICELIB_functions_called RPD VDOT VNORM SPICELIB_subroutines_called RECLAT RESLUN UCRSS VADD VEQU VHAT VMINUS VSCL VSUB

History 2011_12_21: OVERLAPS now reads limited LMRKLISTO.TXT if it exists. This speeds up tiling a region. Use cp USED_MAPS.TXT LMRKLISTO.TXT before starting and rm -f LMRKLISTO.TXT when done. 2011_12_24: Added subroutine INSERTINPIC_SIG to add sigmas in options v5,v6. 2011_12_25: Changed IPL2VLM to avoid counting notopo images as used. 2011_12_27: Option o/1 now autodeletes existing overlaps in case external tiling has introduced bogus LMKNM overlaps. 2011_12_27: Update option u now updates sumfiles with current limbs so external iterations can be imported without running limb fit. 2012_01_10: Weight WN in IPL2SCOBJPTG replaced by separate weights for camera (WC) and s/!!(WS). in INIT_LITHOS.TXT use: PICWTS= 1, 1, 1, 1, 1 WB, WL, WC, WS, WT 2012_01_15: If weight WT = 0 in IPL2SCOBJPTG trajectory loop ignored. 2012_01_16: IPL2VLM changed to remove map overlaps from sigma_lmk determination 2012_01_16: IPL2SCOBJ changed to remove image overlaps from sigma_vso determination 2012_01_23: Updates LMRKLISTO.TXT if it exists as new maplets are created 2012_02_14: Uhused and experimental subroutines removed. 2012_11_08: Code altered to correct type disagreements. 2012_11_11: Included LMRKLISTO warning 2012_12_02: GET_MAPX replaced with GET↩ _MAP in LIMB_HEIGHTS. 2012_12_07: Quit option included in overlaps procedure. 2013_01_28: Included OVE↩ RLAPS.TXT in option B search 2013_01_28: Included AUTOPICK in option B search 2013_08_05: Header added, capitalization & compiler warnings fixed. 2013_09_15: DEBLEMISH now acts on dn, not dnk. 2013_10_04: Redid the capitalization. 2014_02_15: If NPIX=0 in option '0' or '1' lmk deleted and process quits. 2014_02_15: Existence test for INIT_LITHOS.TXT removed. 2014_02_16: HIDE flag introduced to hide menu displays. 2014_02_17↩ : Option'O' completely overhauled. 2014_02_17: Numerical menu inputs changed to character. 2014_02_22: DISPLAY replaced with DISPLAY1. 2014_02_28: Image search cycles once if no ignored images. 2014_03_↩ 25: LMRKLISTX.TXT used in OVERLAPS if LMKLX=.TRUE. IN INIT_LITHOS.TXT. 2014_04_06: 'SEEDS.GRAY' changed back to 'seeds.gray'. 2014_04_06: 'SEEDS.GRAY' changed back to 'seeds.gray'. 2014_04_10: Included error checking changes from Eri!!Palmer. 2014_04_14: ANS=TMPANS changed to ANS=TMPANS(1:1). 2014_↩ 04_27: Smaller fraction for limb maps in 2^/3 eliminated. 2014_04_27: Command line version and seed options added. 2014_05_05: Some annoying EP changes commented out with ce. 2014_05_06: PFLAG changed to 'n' in option v/4. 2014_05_07: Some annoying changes put back in (if CLSEED=.TRUE.). 2014_05_10: PFLAG changed back to 'y' in option v/4.

2.171.1.4 subroutine correlate_pic (real∗4, dimension(5000,5000) *DN1,* integer∗4 *QSZ,* integer∗4 *I0,* integer∗4 *J0, S, X0, Y0, Z0,* logical *FOUND*)

2.172 LITHOSPHERE/LITHOSP.f File Reference

Functions/Subroutines

- program __lithosp.f__

  Procedure LITHOSP.
- subroutine correlate (PMX, NPIX, QSZ, NSZ, DN1, PICID, HIDE, X0, Y0, Z0)

  Variable_I/O.
- subroutine correlate_pic (DN1, QSZ, I0, J0, S, X0, Y0, Z0, FOUND)
- subroutine clear_data (QSZ, NPIX, DN1)
- subroutine display (DN, QSZ, NSZ)

2.172.1 Function/Subroutine Documentation

2.172.1.1 program __lithosp.f__ ( )

Procedure LITHOSP.

Abstract

This procedure is anabbreviation of the LITHOS toolkit for the refinement of landmarks and their associated maplets. It is designed to be run in parallel mode with LOCKOUT files to prevent processes from reading/writing to the same file at the same time. A typical landmark file, CK0008.LMK, looks like:

CK0008 T 49 0.5000000 SIZE, SCALE(KM) -1 -1 -1 -1 HORIZON 0.2500000000D+00 0.2672841486D+00 S↩
IGKM, RMSLMK -0.1151582350D+03 -0.5701086949D+02 0.1512589957D+03 VLM -0.6234954596D+00 -0.↩
5246677995D+00 -0.5796353221D+00 UX 0.6024240851D+00 -0.7949613929D+00 0.7156561315D-01 UY -↩
0.4983358979D+00 -0.3045654297D+00 0.8117273450D+00 UZ 0.1559420789D+00 0.1046028345D+00 0.↩
7422983579D-01 SIGMA_LMK PICTURES N1644784749 786.74 820.27 <- lmk px/ln in image N1644785949 307.69 851.76N1675158257 602.02 730.42 MAP OVERLAPS BK0002 19.309 -1.587 -1.570 <- (V-V').(UX',UY',U↩
Z')/SCALE' CK0006 -66.478 6.974 -6.603 CK0009 -2.082 44.078 -4.142 BK0003 3.661 -43.752 -2.212 LIMB FITS N1558945693 596.88 615.60 0.02 <- lmk px/ln, sigma from limb N1558946626 578.68 616.94 0.02W1644778567 604.73 555.56 0.07 END FILE

```
The first nine lines provide information about the landmark size, scale,
position and orientation.  HORIZON is an unused line, since we are no
longer using that functionality.  In the next line, SIGKM is always half
the scale and RMSLMK (RMS px/ln residual) is not used in LITHOS.  VLM is
the body-fixed landmark vector determined from GEOMETRY and SIGMA_LMK are
the formal uncertainties in each of the components.  UX, UY and UZ are the
unit vectors defining the orientation of the landmark reference plane.

The next block of data, PITCURES, contains the px/ln location of the
landmark in images.  These are initially set using menu option 1 by hand,
then by cross correlation among extracted image data, then by correlation
with a constructed landmark template (menu option 0) and finally by
correlation with the landmark maplet (menu option 2).  It is these
pixel/line locations that are used by GEOMETRY to determine the landmark
vector VLM.  The next two blocks, MAP OVERLAPS and LIMB FITS, are
determined only after a maplet has been constructed and contain
information about the relative locations of neighboring landmarks and the
pixel/line landmark locations in images where the maplet spans part of the
lit limb.  These data also contribute to the determination of landmark
location by GEOMETRY.

The main menu for LITHOSP is shown below.  It is divided into a number of
processes, each documented separately.  The documentation for process P,
for example, can be found by searching on P^.   This menu is an
abbreviation of that for LITHOS, and a number of the options in the
processes are also missing.  Its functionality is to refine the solutions
and it does not change the inputs on which those solutions are based.
```

```
...     MAIN MENU     ...             For process abstract:

 Q. Quit LITHOS                              search Q^

LANDMARK/MAPLET CONSTRUCTION
 0. Find template                            search 0^
 1. Align landmarks                          search 1^
 2. Find heights                             search 2^
 O. Attach map to maps or limbs              search O^

LANDMARK/MAPLET I/O AND CREATION
 I. Input landmark                           search I^
 U. Update landmark files                    search U^

ADJUST INPUT DATA AND NOMINALS
 N. Find normal                              search N^
 V. Solve for V                              search V^
 Z. Predict px/ln                            search Z^
 A. Reset albedo or slopes                   search A^
 M. Get heights from shape model             search M^
 B. Get heights from surrounding map         search B^
 X. Turn on/off extract filter               search X^

INFORMATION AND DISPLAY
 H. Hide/Show screen output                  search H^

There are several differences between LITHOSP and LITHOS.  First,
LITHOSP uses the extended LMRKLISTX.TXT (made by procedure MAKE_LMRKLISTX)
to allow a quick determination of what landmark files need to be used for
the overlap computation.  This is because LITHOSP is not making any new
landmarks, so we only need to run MAKE_LMRKLISTX once.  LITHOS, of course,
has its own shortcut in LMRKLISTO.TXT that can often be used.  Second,
each process running LITHOSP has its own copy of the shapemodel called
TSHP01.TXT, TSHP02.TXT, etc.  This dates from a computer that had an
unfortunate propensity to occasionally corrupt the shape model, and if
LITHOSP detects such a corruption, it will replace the bad model with a
good one.  This is no longer an issue, but the code persists.  Third,
LITHOSP reduces the search for limbs by using only those limbs that have
already been found spanning the maplet.  If a full search is desired, then
the line NEWLIM=.TRUE. should be included in INIT_LITHOS.TXT.  LITHOSP
also has its own DISPLAY subroutine that produces a maximally stretched
display file LMRK_DISPLAY01.PGM, LMRK_DISPLAY02.PGM, etc., one for each
process.  At the end of each landmark update, these are copied into the
TESTFILES1 directory with the name LMKNM.pgm for diagnosti!!purposes.

In order to run LITHOSP a list of landmark names ending with END is saved
as make_script.in, for example:

CK0001
CK0002
CK0003
CK0004
CK0005
CK0006
CK0007
CK0008
CK0009
CL0001
CL0002
CL0003
END
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.172.1.2 subroutine clear_data (integer *QSZ,* integer *NPIX,* real∗4, dimension(5000,5000) *DN1*)

2.172.1.3 subroutine correlate (integer∗4 *PMX,* integer∗4 *NPIX,* integer∗4 *QSZ,* integer∗4 *NSZ,* real∗4, dimension(5000,5000) *DN1,* character∗12, dimension(pmx) *PICID,* logical *HIDE, X0, Y0, Z0*)

Variable_I/O.

Variable I/O Description

USR I ??? read in from LITHOS_INIT ASIG I ??? read in from LITHOS_INIT SSIG I ??? read in from LITHO↩ S_INIT CHI0 I ??? read in from LITHOS_INIT SEED I ??? read in from LITHOS_INIT BODY I ??? read in from LITHOS_INIT BLIM I ??? read in from LITHOS_INIT RESLM I ??? read in from LITHOS_INIT PRNLM I ??? read in from LITHOS_INIT IGNLM I ??? read in from LITHOS_INIT IGNUM I ??? read in from LITHOS_INIT NOFIT I ??? read in from LITHOS_INIT NUMLM I ??? read in from LITHOS_INIT SIZLM I ??? read in from LITHO↩ S_INIT PICLM I ??? read in from LITHOS_INIT RECVR I ??? read in from LITHOS_INIT ALPAD I ??? read in from LITHOS_INIT RENAME I ??? read in from LITHOS_INIT RPLUSE I ??? read in from LITHOS_INIT REG↩ FLG I ??? read in from LITHOS_INIT REG I ??? read in from LITHOS_INIT KB I ??? read in from LITHOS_INIT DENOISE I ??? read in from LITHOS_INIT IGNORE I ??? read in from LITHOS_INIT USR I ??? read in from LITHOS_INIT ANS I User input from multiple menu selection items NAME I User supplied landmark name. QSZ I Landmark size read from landmark file SCALE I Landmark scale (pix/km) from landmark file IHMN I Landmark I horizon minimum from landmark file INMX I Landmark I horizon maximum from landmark file JHMN I Landmark J horizon minimum from landmark file JHMX I Landmark J horizon maximum from landmark file SIG_KM I ???? from landmark file RMS_LMK I ???? from landmark file V(3) I Landmark distance from body center from landmark file UX(3) I Landmark X normal unit vector from landmark file UY(3) I Landmark Y normal unit vector from landmark file UZ(3) I Landmark Z normal unit vector from landmark file SIGMA_LMK I ???? from landmark file PICID I Image name read from PICTURE section of the landmark file IPL I Image pixel and line number read from PICTURE section of the landmark file. PFLAG I Image use flag read from PICTURE section of the landmark file. XNAME I Image name read from PICTLIST.TXT file. NPX I Number of image pixels from ./SUMFILES/<PICNM>.SUM NLN I Number of image lines from ./SUMFILES/<PICNM>.SUM V(3) I Distance from S/!!to target center. CX(3) I Camera X unit pointing vector. CY(3) I Camera Y unit pointing vector. CZ(3) I Camera Z unit pointing vector. SZ(3) I Sun pointing vector. LMRKOLD I User supplied landmark name to replicate. LMRKNEW I User supplied landmark name to be replicated.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./TESTFILES/<LMKNM>.LMK O ./LMKFILES/<LMKNM>.LMK I ./TESTFILES/<LMKNM>.MAP O ./TMPLIST/<USR>.TXT ./TESTFILES/LMRK↩ LISTX.TXT O ./LMRKLISTX.TXT O ./LMKFILES/<LMKNM>.LMK I&O ./MAPFILES/<LMKNM>.MAP i ? ./SUM↩ FILES/<PICID>.SUM I Image summary file for <PICID> ./TESTFILES/<PICID>.SUM O Image summary file for <PICID> lockout. ./TESTFILES/<NAME>.LMK O ./LMKFILES/<NAME>.LMK I ./LMRK_DISPLAY.gray O Temp gray scale map file. ./LMRK_DISPLAY.pgm O Temp pgm map file converted by RAW

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called LMCOUNT_LO PICRES_LO PXCOUNT_LO RANN SLEN

SPC_subroutines_called ATTACH_LO DEBLEMISH DELAY DENOISE DISCONNECT_LMFILE_LO EXTRACT_↩ DATA_LO EXTRACT_DATA_PIC_LO FIND_ALBEDO FIND_LAMBDA FIND_SLOPES GET_HEIGHTS_LO GE↩ T_MAP_LO HGT2SLP INSERTINPIC_LIM_LO INSERTINPIC_LMK_LO IPL2VLM_LO LIMB_HEIGHTS_LO LO↩ C2PIX_LO MAKE_TMPLIST_LO PATCH_COORDS PICINPT_LO PREDICT_DATA RAW2PGM READ_MAP S↩ LP2HGT STEREO_LO V2IMGPL WRITE_MAP

SPICELIB_functions_called RPD VDOT VNORM

SPICELIB_subroutines_called RECLAT RESLUN VHAT VSCL VSUB

History 2011_12_25 Changed IPL2VLM to avoid counting notopo images as used 2011_12_27 Update option u now updates sumfiles with current limbs so external iterations can be imported without running limb fit. 2012_↩ 01_16 IPL2VLM changed to remove map overlaps from sigma_lmk determination 2013_09_15 New DEBLEMISH subroutine now used. 2013_09_18 LIMB_HEIGHTS_LO replaces LIMB_HEIGHTS 2013_10_04 Capitalization, compiler warnings fixed and header added. 2014_02_25 If NPIX=0 in option '0' or '1' process quits. 2014_↩ 02_25 Existence test for INIT_LITHOS.TXT removed. 2014_02_25 HIDE flag introduced to hide menu displays. 2014_02_25 Numerical menu inputs changed to character. 2014_02_26 0 spacing in CORRELATE asks for new input. 2014_04_27 Smaller fraction for limb maps in 2^/3 eliminated. 2014_05_06: Command line version and seed options added. 2014_05_12: RECENT set to LMKNM on successful input so update not necessary to invoke. 2014_06_06: RECENT now a valid input in scripts.

2.172.1.4 subroutine correlate_pic (real∗4, dimension(5000,5000) *DN1,* integer∗4 *QSZ,* integer∗4 *I0,* integer∗4 *J0, S, X0, Y0, Z0,* logical *FOUND*)

2.172.1.5 subroutine display (real∗4, dimension(5000,5000) *DN,* integer *QSZ,* integer *NSZ*)

2.173 LITHOSPHERE/MAKE_SUMFILES.f File Reference

Functions/Subroutines

- program __make_sumfiles.f__

  Procedure MAKE_SUMFILES.

2.173.1 Function/Subroutine Documentation

2.173.1.1 program __make_sumfiles.f__ ( )

Procedure MAKE_SUMFILES.

Abstract

This procedure creates the initial PICNM.SUM files and PICNM.NOM files from the current set of SPICE kernels. It reads the file make_sumfiles.in that is created by the PROCESS_IMG procedure and whose records contain:

PICNM, s/!!clocK_count, Camera #, s/!!#, binning factor, optional information

The picture name PICNM is a 12 character identifier. If the project image names are too long, we make up our own in the PROCESS_IMG procedure. On Rosetta, for example, the name consists of a letter designating a camera (there are 12 on board), seconds past J2000 (nine digits), and two digits representing the filter wheel combination. The spacecraft clock is a string that can have different formats, depending on the mission. The procedure can handle any clock string format up to 32 characters in length. The next column specifies the number of the camera used to take the image. This number is cross referenced to data in the make_sumfiles.txt file discussed below. It is becoming increasingly common for a single mission to carry more than one spacecraft. The s/!!# column was introduced for the Deep Impact mission that had two independent spcaecraft (for a short time). The ROSETTA mission has two as well, the ROSETTA spacecraft the the PHILAE lander. The last active column refers to the binning factor n, assumed to be always square (nxn). Note that some ROSETTA images can be windowed, with only a portion of the full size image downlinked. The PROCESS_IMG procedure puts the window in the proper full-frame context and fills the remaining space with 0, representing "no data". The remainder of the record, optional information, is not read by MAKE_SUMFILES. On ROSETTA, this optional space is used to record the original full project image name, so that make_sumfiles.in provides a dictionary to translate from our image designations to the project's.

In many test simulations, image headers contain UT!!but not SCLK. The procedure now recognizes UT!!by a blank or a - as the fifth character and adjusts accordingly.

The second main input file for MAKE_SUMFILES, make_sumfiles.txt, contains information about the cameras, spacecraft, target and all the necessary SPICE kernels. The first record is the number of cameras:

```
12                                              NCAM
```

```
followed by a block for each camera looking like:

 'ROS_OSIRIS_NAC'                                    CAMERA NAME
 2048  2048                                          NPX, NLN
 717.32  1024.5  1024.5                             MMFL, PX0, LN0
 -74.0741, 0, 0, 0, 74.0741, 0                       KMAT
 -2, 1, 3                                            PROJECT CAMERA AXES
  400, 65535                                         T1,T2

with records for camera name, image size, focal length and central pixel/line,
K-matrix, camera axes and thresholds.  The camera axes (Cx,Cy,Cz) in the SP!!software
are defined in space to align with the sample (pixel), line and boresight directions:

                          V
                        .
                       .
                      .
                     .
                    .
                   .
                  .
                 .
                .       Cz
               .      /
             o       /
           --|--    -----> Cx
             |      |
            / \     |
                    v
                    Cy
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.174 LITHOSPHERE/NEW_POLE.f File Reference

Functions/Subroutines

- program __new_pole.f__

2.174.1 Function/Subroutine Documentation

2.174.1.1 program __new_pole.f__ ( )

2.175 LITHOSPHERE/OMEGA.f File Reference

Functions/Subroutines

- program __omega.f__

  Procedure OMEGA.
- subroutine mmr2v (MMFL, MM, V0, CX, CY, CZ, R, V, U, H)

  Variable_I/O.
- subroutine va2vb (VA, ALPHA, VB)
- subroutine partials (VA, ALPHA, U, CX, CY, CZ, DZ)

2.175.1 Function/Subroutine Documentation

2.175.1.1 program __omega.f__ ( )

Procedure OMEGA.

Abstract

This program solves for the angular velocity of a body's rotation by observing the motion of landmarks in nearby images. The pointing scobj originally determined in body-fixed space from a nominal .pck file are transformed back to inertial space. If V is the vector of a landmark at time = 0 (PICNM0) then the inertial space vector V' at time = t (PICNM1) is given by

```
V' = Vcos(a) +(AxV)sin(a)/a +A(A.V)(1-cos(a))/a^2
```

where A is the inertial space vector OMEGA x t and a is its magnitude with OMEGA = angular velocity. The program solves for A, for an offset of image PICNM1 due to relative pointing errors and for the inertial space landmark positions at t=0. Many iterations ($\sim$100) are required for convergence (but it is very fast) and OMEGA is printed both as an inertial space vector and in RA, DEC,|OMEGA| in deg and deg/da.

Although a small pointing offset is solved for, the images should be at least roughly regitered (REGISTER) before starting.

The program can now update the SUMFILE if desired.

The program can now produce a file called OMEGA_SUM.TXT that keeps track of Inertial space RA/DE!!of omega, Inertial space RA/DE!!of body frame z-axis, and the LAT/LON of the omega vector in the body frame. Over time, omega in the body-fixed frame will trace out an ellipse. The center of the ellipse will be the z-principal axis of the body, while the major axes of the ellipse will be in the directions of the other principla moments. The ration of the axes of the ellipse in the x and y directions will be

```
x/y = sqrt(I2*(I3-I2))/sqrt(I1*(I3-I1))
```

Disclaimer

Abondon hope, ye who enter here.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.175.1.2 subroutine mmr2v (double precision *MMFL,* double precision, dimension(2) *MM,* double precision, dimension(3) *V0,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* double precision *R,* double precision, dimension(3) *V,* double precision, dimension(3) *U,* double precision *H*)

Variable_I/O.

Variable I/O Description

PICNM0 I User supplied image name to be processed. PICNM1 I User supplied image name to be processed. PXDIAM I User supplied pixel diameter of PICNM0. ANS I User supplied answer to multiple menu selections.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./SUMFILES/<PICNM>.SUM I File containing image's summary information. Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called PXMM INVERTN WHOLEPIC RAW2PGM V2IMGPL

SPICELIB_functions_called CLIGHT HALFPI RESLUN RPD SPD VDOT VNORM

SPICELIB_subroutines_called FURNSH EUL2M M2EUL MTXV PXFORM UTC2ET VCRSS VHAT RECRAD IN↩
VERT

LOCAL_subroutines_called MMR2V VA2VB PARTIALS

History 2014_01_15: First functioning version. Tested with Mimas 2014_01_16: Added option to fix rotation rate. 2014_01_16: Dimensions increased to 5000 from 1000. 2014_01_17: Outputs inertial and body-fixed landmark vectors. 2014_01_17: Solves for RA/DEC/TW of body-fixed frame. 2014_05_08: Major surgery. 2014_05_17: Produces OMEGA_SUM.TXT file for display

2.175.1.3 subroutine partials (double precision, dimension(3) *VA,* double precision, dimension(3) *ALPHA,* double precision, dimension(3) *U,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* double precision, dimension(3,0:3) *DZ*)

2.175.1.4 subroutine va2vb (double precision, dimension(3) *VA,* double precision, dimension(3) *ALPHA,* double precision, dimension(3) *VB*)

2.176 LITHOSPHERE/POLE.f File Reference

Functions/Subroutines

- program __pole.f__

  Procedure POLE.

2.176.1 Function/Subroutine Documentation

2.176.1.1 program __pole.f__ ( )

Procedure POLE.

Abstract

This program solves for the pole (RA/DEC) and rotation rate in DEG/DA for a body in principal axis rotation. It simultaneously solves for the body-fixed locations of all landmark centers.

There are two important simplifications that make this solution possible:

First, changes inthe predicted pixel/line locations of landmarks in an image depend only on the orientation of the landmark vectors relative to the camera pointing. The orientation of the spacecraft-object vector to the camera pointing is invarient under rotations.

Second, equations for the changes to the rotational parameters r and the landmark vectors l are

```
| A    B   B   B| | dr |     | wr |
|                | |    |     |    |
| Bt   D        | | dl |     | wl |
|                | |    |   = |    |
| Bt       D    | | dl |     | wl |
|                | |    |     |    |
| Bt           D| | dl |     | wl |
```

with many many more landmark terms. wr and wl are the residuals multiplied by appropriate partials. These can be recast as

```
Adr + Bdl = wr
Btdr + Ddl = wl  (one for each landmark)
```

and using the second equation to eliminate the dl from the first

```
(A-BDiBt)dr= wr-BDiwl  (Di = inverse of D)
```

so to find the changes in the rotational parameters, we only have to invert a small matrix.

This new version of POLE continues with the solution for the landmark vectors. First, the new values of the rotation parameters is applied and fixed, so only the equations

```
Ddl = wl  (one for each landmark)
```

need to be solved. The updaded walues of the landmark vectors are retained for the next iteration.

Program has been tested against old POLE routine (that had no iterative capability) and gives identical results in the first iteration.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.177 LITHOSPHERE/PROCESS_FITS.f File Reference

Functions/Subroutines

- program __process_fits.f__

  Procedure PROCESS_FITS.

2.177.1 Function/Subroutine Documentation

2.177.1.1 program __process_fits.f__ ( )

Procedure PROCESS_FITS.

Abstract

This procedure reads a FITS image file, extracts relevant information from the header, adds a record to the make_sumfiles.in file read by the MAKE_SUMFILES program and adds a raw 8- or 16- bit image to the IMAGEFILES sub-directory for SP!!processing. make_sumfiles.in records are:

PICNM, UT!!OR SCLK, CAMERA #, S/!!#, BINNING, FITS NAME

The 12 character PICNM consists of a one character camera identifier, a 9-character ephemeris time, and a two-character filter identifier. SCLK is used if available, but UT!!can also be read by MAKE_SUMFILES. There are five camera's currently included, polycam, mapcam, samcam, and two nav cams. Both the navcams are have the N prefix, but are distinguished by filter numbers of 1 and 2. OSIRIS-REx has only one spacecraft (unlike DI, ROSETTA, GALILEO and CASSINI), and there will be no binning, so the fourth and fifth entries are always 1. The final column keeps track of the original FITS name for the image. It is not used by MAKE_SUMFILES but it is useful to preserve this information.

The times given in the make_sumfiles.in file correspond to the middle of the exposure. We are assuming that the times in the FITS header are the start times. Assuming that this is the case, a flag TSHIFT is set to .TRUE. near the beginning of the program. If it turns out that the times are midpoints, then TSHIFT should be changed to .FALSE.

Provision exists for the possibility of windowing, an option that will not be used on OSIRIS-REx, but will on ROSETTA. If only a portion of the image is send down, that portion will be imbedded in a full size image, with unused pixels having dn number of zero.

The mission has the option of downloading only 8-bit data and providing a look-up table (LUT) to approximate the full 16 bits of the image. As yet, this has yet to be implemented. If and when it is, the program will be updated using code similar to that used for CASSINI.

The data from 8 and 16 bit images is included unchange. If 32-bit images are provided (a pretty stupid thing to do) the program maximally stretches the data and creates 16 bit files. The problem here is that the threshold values for each image must be determined by hand and entered via the Display program. The output files, either raw 8- or 16-bit data, are saved as PICNM.DAT in the IMAGEFILES sub-directory.

Disclaimer

Abandon hope, ye who enter here.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.178 LITHOSPHERE/PROCESS_IMG.f File Reference

Functions/Subroutines

- program __process_img.f__

  Procedure PROCESS_IMG.

2.178.1 Function/Subroutine Documentation

2.178.1.1 program __process_img.f__ ( )

Procedure PROCESS_IMG.

Abstract

2.179 LITHOSPHERE/REGISTER.f File Reference

Functions/Subroutines

- program __register.f__

  Procedure REGISTER.
- subroutine correlate (dni, dnr, qsz, x0, y0, z0, found)

  Variable_I/O.

2.179.1 Function/Subroutine Documentation

2.179.1.1 program __register.f__ ( )

Procedure REGISTER.

Abstract

REGISTER provide an initial estimate for the spacecraft state (camera pointing and s/c-object vector) by aligning an image with a known object - either the shape model, a high resolution map or another (already registered) image. The user enters the image name and the object to align with:

```
s = shape
i = reference image
m = reference map
```

If i or m is chosen, you will be prompted for an image name or a map name. If '0' is chosen for the map name, the program will search a set of "Zmaps" for the one most likely to oderlap the image. Zmaps are labeled Z(N/S)#### where the first two numbers are the latitude center divided by 5 (00-18) and the second pair is east longitude divided by 5 (00-71). Thus ZS0837 is a map with a center at 40 degrees south and 185 degrees east. Maps are only used after a detailed shape model and high-resolution maps have been constructed. At that time we are registering new images for navigation or improving the topography.

A final entry is the scale in km. The basi!!display for REGISTER is 600x600 pixels. If the body in question is, say 500 m across, then if the scale is chosen to be 5 m (.005) the image will be 100 pixels across in the display.

The display and the arrays of image and reference data are not the actual imaging data, but that data projected on a "substrate". When we are just starting processing and looking at low-resolution images of the body, the substrate is simply a plane through the body center oriented parallel to the camera's focal plane. This flat 'f' substrate is the default value. Once a decent shape model is obtained and our images cover a small fraction of the body's surface, the substrate is taken to be that surface itself, either in the form !! of the shape 's' or a high-resolution map 'm'. In this case, the topography is represented as a DTM whose reference plane is the same !! as the flat substrate with heights in the negative camera bore sight direction. Note that if the reference is also an image, this data is projected on the same substrate as the image being registered.

The main menu looks like:

```
0. Quit
1. Change scale
2. Global shift
3. Shift unknown (LEFT/RED) image
4. Rotate unknown (LEFT/RED) image
5. Change reference
6. Change RANGE of (LEFT/RED) image
7. Revert to nominal
8. Change substrate
9. Update nominal and quit
a. Toggle bkg
b. Toggle image for Vlm
c. Change correlation limit
d. Fix/Unfix scobj
e. Fix/Unfix pointing
t. Tuck picture
```

Options 1, 5 and 8 allow you to change the scale, reference object and substrate, respectively.

Options 3, 4 and 6 make changes to the .SUM file in camera pointing and/or cross line-of sight scobj, camera twist and s/!!range, respectively. The 3 option is the one most used. There is an autocorrelate that, if chosen, will estimate the offset between the image and the reference. If the correlation is less than a preset limit it will ask for a manual input. That limit, initially set to 0.25 can be changed with option c. It should be noted that if the a batch or other automati!!run is being performed, then if the correlaton fails there is a provision for the procedure to end gracefully. If after 3 and then y are entered, the line XSTOP is input then if the correlation fails the .SUM data will be reset to the input values and the program stopped. If XSKIP# is input, the program will skip # lines of the script, reset the .SUM data and ask for the next image to be input. A recent make_scriptR.seed for a batch run registering 450 Vesta images to Zmaps was:

```
m               <- reference = map
0               <- Auto-choose Zmap
1.0             <- Scale = 1 km
a               <- Turn off abckground
3               <- Shift image
y               <- Autocorrelate
XSTOP           <- Stop on no correlation
1               <- Change scale
0.25            <- New scale = 250 m
3               <- Shift image
y               <- Autocorrelate
XSTOP           <- Stop on no correlation
0               <- Quit
y               <- Save new .SUM data
n               <- Don't save rotation history
```

```
n                <- Don't save nominal
q                <- Quit procedure
END
```

The last two 'n' are "always" so. The rotation history file was introduced to keep track of pointing errors in Clementine data during Lunar orbits in an attempt to quantify systemati!!shifts and it is rare that we want to set the nominal file equal to the .SUM solution. The 'a' on the fourth line deserves some expalnation. When we are correlating small images to a nominal shape, we want to use all the data, so the space off the body counts just as much as the body itself. The procedure wakes up with a "background" turned on so that this correlation can be performed. By typing 'a', we toggle this background off, so it is only the common topography that is correlated between the image anf the Zmap.

Sometimes when we are trying to align an image to a reference, the entire display is off center. The '2' option moves both the image and reference displays by the same amount so that we can more conveniently align them, usually at smaller scale.

When an image shift has been determined, either manually or through autocorrelation, The camera pointing ans spacecraft-object (scobj) vector in the .SUM file are changed in a manner weighted by their respective sigmas in the nominals (.NOM) file. If we want to keep one or the other unchanged, we use the 'd' or 'e' option to fix it.

The '7' option populates the working .SUM file. Thie is sometimes a baleout procedure after having screwed up the .SUM file in some way. However, we have now introduced a new option '9' that lets us save the current result as the nominal without changing the .SUM file from its original value. If for example, the spacecraft range is out to lunch (as it was on Hayabusa) the working .SUM file can be populated with the nominal, a change made to the range with option '6', and the new nominal saved with option '9'.

There are two other options that are included in REGISTER for convenience. Option 'b' allows a flag to be set on the image that will enable its brightness variations to be used to determine topography but keeps it from participating in the geometry solution for the landmark vector. This is used to keep Mariner 10 images of Mercury, which have questionable nominals, from messing up the vector but still, with their sometimes unique sun angles, helping with the topography determination. The 't' option allows an image to be tucked so it will not participate in the SP!!process at all. It could be tucked from LITHOS, but it often happens that as REGISTER is used to cycle through new images, problems are easily seen and dealt with immediately.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.179.1.2 subroutine correlate (real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *dni,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *dnr,* integer∗4 *qsz, x0, y0, z0,* logical *found*)

Variable_I/O.

Variable I/O Description

PICNM I User supplied image name to be processed. REF I User supplied reference type (shape, image, or map) REFNM I User supplied reference name. SCALE I User supplied map scale in km/px. ANS I User supplied answer to multiple menu selections. CORLM I User supplied correlation limit. RA I User supplied RA of the pole. De!! I User supplied declination of the pole. TW I User supplied twist of the pole. SUBSTRATE I User supplied substrate type (shape, flat, map) SUBST I User supplied substrate name. ROT(3) I User supplied offset to the C-matrix. VNORM(W) O Range between S/!!and target. PICNM O Current image name. SHAPE O Current shape. REFNM O Current reference name.

File_I/O

Filename I/O Description

./INIT_LITHOS.TXT I File containing default values to be read in by SP!!toolkit. ./ROTATION.TXT I ./SUMFILE↩ S/<PICNM>.SUM I File containing image's summary information. Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called CORRELATE EXTRACT_DATA_PI!! GET_HEIGHTS GET_MAP HGT2SLP IMGPL2VN INSERTINPIC_POLE INSERTINPIC_PTG INSERTINPIC_SOV INSERTINPIC_SZ INSERTINPIC_VFLG INSERTINPI!! INSERTINNOM_SOV INVERTN PICINPT PREDICT_DATA_PI!! RAW2PGM RAWPPM READ_HEADER U2VN V2IMGPL

SPICELIB_functions_called CLIGHT HALFPI RESLUN RPD VDOT VNORM

SPICELIB_subroutines_called EUL2M FURNSH LATRE!! M2EUL MTXV MXV PXFORM UCRSS UTC2ET VADD VCRSS VEQU VHAT VMINUS VROTV VSCL VSUB

History 2012_04_10: Subroutine U2V replaced by U2VN 2012_11_08: Code altered to correct type disagreements including change of eps to real∗4. 2012_12_01: Bug fixed by replacing GET_MAPX with GET_MAP. 2013_07_31: Header added, capitalization & compiler warnings fixed. 2013_06_02: Bug fixed by including previously existing V2IMGPL before image extraction. This was inadvertently left out during the transition tothe common PICINPT subroutine. 2013_09_11: NOMINAL file update option reinstated. 2013_09_11: Corrections made to INSERTINNOM calls. 2013_09_13: Unused W(I) lines removed in option 3 2013_09_13: S1 and S2 redefined as rms sigmas. 2013_11_20: dnx<0 values set to 0. 2013_11_20: eps not used in display. 2014_03_11: Existence test for INIT_LITHOS.TXT removed. 2014_04_23: Find Pole experimental stuff removed (opt 9) 2014_04_23: Current values can update nominal. No .SUM change. 2014_05_06: Command line Version option added.

2.180 LITHOSPHERE/REGRES.f File Reference

Functions/Subroutines

- program __regres.f__

  Procedure REGRES.

2.180.1 Function/Subroutine Documentation

2.180.1.1 program __regres.f__ ( )

Procedure REGRES.

Abstract

This program produces a set of REGRES files as an interface between SP!!and the KinetX nagigation software. A file is made for each image in the restricted list PICTLISTR.TXT, and only landmarks from a restricted list LMKRLISTR.TXT are used. For each image in PICTLISTR.TXT the process produces a REGRES file in the REGRES_FILES directory called PICNM.TXT. Each file contains:

```
REGRES FILE.  CREATED           [time of creation]
PARTIALS UNITS: PX/KM, PX/DEG

CAMERA ID
SPACECRAFT ID
TARGET ID
ABCORR

UT!!                            [Image time]
ET SE!!PAST J2000

NPX, NLN                        [Image size ib pixels and lines]
MMFL, CTR                       [Camera focal length and image center]
```

```
K-MATRIX

TITV                            [Inertial to camera frame rotation]
TPMI                            [Body-fixed to inertial rotation]

LITHOS SOLUTION:

     S!!- OBJ VECTOR
     CAMERA RA, DC, TW
     S/!!POSITION SIG
     PNT SIGMA (DEG)
     TWIST SIGMA (DEG)

NOMINAL SOLUTION:

     S!!- OBJ VECTOR
     CAMERA RA, DC, TW
     SOLAR UNIT VECTOR

LANDMARKS:                      [One for each landmark]

     BODY-FIXED VLM
     B-F VLM SIGMA              [Formal uncertainty in LM position]
     PX, LN OBSERVED            [Landmark location in image]
     PX, LN PREDICTED           [Uses nominal scobj, solved pointing]
     dpx/dW1, dln/dW1
     dpx/dW2, dln/dW2           [Partials wrt scobj]
     dpx/dW3, dln/dW3
     dpx/dRA, dln/dRA
     dpx/dDC, dln/dD!!          [Partials wrt pointing Euler angles]
     dpx/dTW, dln/dTW

 END LANDMARKS

 KERNEL LIST

     [list of all spice kernels used]

 END FILE
```

Because the navigation software does not solve for camera pointing, the SP!!camera solution is used rather than the nominal.

Disclaimer

Abandon hope, ye who enter here.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.181 LITHOSPHERE/RESIDUALS.f File Reference

Functions/Subroutines

- program __residuals.f__

  Procedure RESIDUALS.

2.181.1 Function/Subroutine Documentation

2.181.1.1 program __residuals.f__ ( )

Procedure RESIDUALS.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.182 LITHOSPHERE/SHIFT.f File Reference

Functions/Subroutines

- program __shift.f__

2.182.1 Function/Subroutine Documentation

2.182.1.1 program __shift.f__ ( )

2.183 MAPMAKER/LIST_MMTILES.f File Reference

Functions/Subroutines

- program __list_mmtiles.f__

2.183.1 Function/Subroutine Documentation

2.183.1.1 program __list_mmtiles.f__ ( )

2.184 MAPMAKER/MAKE_MMTILES.f File Reference

Functions/Subroutines

- program __make_mmtiles.f__

2.184.1 Function/Subroutine Documentation

2.184.1.1 program __make_mmtiles.f__ ( )

2.185 MAPMAKER/MAPMAKER.f File Reference

Functions/Subroutines

- program __mapmaker.f__
- subroutine orient (UX, UY, UZ)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine write_map (LMRKFILE, LMKNM, NTMP, QSZ, SCALE,
- subroutine flip (n, lflag, ch1, ch2)
- subroutine u2v (UZ, V)
- subroutine get_heights (ntmp, qsz, scale, ux, uy, uz, v0, infile,
- subroutine get_model (infile, q, vec)
- subroutine raw2pgm (infile, outfile, npx, nln)

- subroutine raw2ppm (infile, outfile, npx, nln)
- integer ∗4 function slen (STRING)
- subroutine imgpl2v (PICNM, IMGPL, USE, V)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine pxmm (KMAT, CTR, D, PX, MM)
- subroutine mm2v (MM, MMFL, V0, CX, CY, CZ, USE, V)
- subroutine display (NTMP, Q0, S0, V, UX, UY, UZ,
- subroutine patch_coords (qsz, huse, ht, ux, uy, uz)
- subroutine invertn (N, M, MINV)
- double precision function vnorm (V1)

  Procedure VNORM (Vector norm, 3 dimensions)
- double precision function vdot (V1, V2)

  Procedure VDOT (Vector dot product, 3 dimensions)
- double precision function rpd ()

  Procedure RPD (Radians per degree)
- double precision function spd ()

  Procedure SPD (Seconds per day)
- subroutine ucrss (V1, V2, VOUT)

  Procedure UCRSS (Unitized cross product, 3x3)
- subroutine vhat (V1, VOUT)

  Procedure VHAT ("V-Hat", unit vector along V, 3 dimensions)
- subroutine latre

  Procedure LATRE!!(Latitudinal to rectangular coordinates)
- subroutine vequ (VIN, VOUT)

  Procedure VEQU (Vector equality, 3 dimensions)
- subroutine vscl (S, V1, VOUT)

  Procedure VSCL (Vector scaling, 3 dimensions)
- subroutine vsub (V1, V2, VOUT)

  Procedure VSUB (Vector subtraction, 3 dimensions)

2.185.1 Function/Subroutine Documentation

2.185.1.1 program __mapmaker.f__ ( )

2.185.1.2 subroutine display (integer *NTMP,* integer *Q0,* double precision *S0,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.185.1.3 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.185.1.4 subroutine get_heights (integer∗4 *ntmp,* integer∗4 *qsz,* real∗8 *scale,* real∗8, dimension(3) *ux,* real∗8, dimension(3) *uy,* real∗8, dimension(3) *uz,* real∗8, dimension(3) *v0,* character∗72 *infile*)

2.185.1.5 subroutine get_model (character∗72 *infile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.185.1.6 subroutine imgpl2v (character∗12 *PICNM,* double precision, dimension(2) *IMGPL,* logical *USE,* double precision, dimension(3) *V*)

2.185.1.7 subroutine invertn (integer∗4 *N,* real∗8, dimension(6,6) *M,* real∗8, dimension(6,6) *MINV*)

2.185.1.8 subroutine latre ( )

Procedure LATRE!!(Latitudinal to rectangular coordinates)

Abstract

Copyright

Required_Reading

Keywords

Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Exceptions

Files

Particulars

Examples

Restrictions

Author_and_Institution

Literature_References

Version

Index_Entries

Revisions

2.185.1.9 subroutine mm2v (double precision, dimension(2) *MM,* double precision *MMFL,* double precision, dimension(3) *V0,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(3) *V*)

2.185.1.10 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.185.1.11 subroutine orient (double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ*)

2.185.1.12 subroutine patch_coords (*qsz,* logical, dimension(-ntmp:ntmp,-ntmp:ntmp) *huse,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ht, ux, uy, uz*)

2.185.1.13 subroutine pxmm (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *PX,* double precision, dimension(2) *MM*)

2.185.1.14 subroutine raw2pgm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.185.1.15 subroutine raw2ppm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.185.1.16 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.185.1.17 double precision function rpd ( )

Procedure RPD (Radians per degree)

Abstract

Copyright

Required_Reading

Keywords

Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Exceptions

Files

Particulars

Examples

Restrictions

Literature_References

Author_and_Institution

Version

Index_Entries

2.185.1.18 integer∗4 function slen (character∗(∗) *STRING*)

2.185.1.19 double precision function spd ( )

Procedure SPD (Seconds per day)

Abstract

Copyright

Required_Reading

Keywords

Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Exceptions

Files

Particulars

Examples

Restrictions

Literature_References

Author_and_Institution

Version

Index_Entries

2.185.1.20 subroutine u2v (double precision, dimension(3) *UZ,* double precision, dimension(3) *V*)

2.185.1.21 subroutine ucrss (double precision, dimension (3) *V1,* double precision, dimension (3) *V2,* double precision, dimension (3) *VOUT*)

Procedure UCRSS (Unitized cross product, 3x3)

Parameters

v1	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

Revisions

2.185.1.22 double precision function vdot (double precision, dimension (3) *V1,* double precision, dimension (3) *V2*)

Procedure VDOT (Vector dot product, 3 dimensions)

Parameters

v1	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

2.185.1.23 subroutine vequ (double precision, dimension (3) *VIN,* double precision, dimension (3) *VOUT*)

Procedure VEQU (Vector equality, 3 dimensions)

Parameters

vin	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

2.185.1.24 subroutine vhat (double precision, dimension (3) *V1,* double precision, dimension (3) *VOUT*)

Procedure VHAT ("V-Hat", unit vector along V, 3 dimensions)

Parameters

v1	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

Revisions

2.185.1.25 double precision function vnorm (double precision, dimension (3) *V1*)

Procedure VNORM (Vector norm, 3 dimensions)

Parameters

v1	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

2.185.1.26 subroutine vscl (double precision *S,* double precision, dimension (3) *V1,* double precision, dimension (3) *VOUT*)

Procedure VSCL (Vector scaling, 3 dimensions)

Parameters

s	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

2.185.1.27 subroutine vsub (double precision, dimension (3) *V1,* double precision, dimension (3) *V2,* double precision, dimension (3) *VOUT*)

Procedure VSUB (Vector subtraction, 3 dimensions)

Parameters

v1	Declarations

Brief_I/O

Detailed_Input

Detailed_Output

Parameters

Particulars

Examples

Restrictions

Exceptions

Files

Author_and_Institution

Literature_References

Version

Index_Entries

2.185.1.28 subroutine write_map (character∗72 *LMRKFILE,* character∗6 *LMKNM,* integer *NTMP,* integer *QSZ,* double precision *SCALE*)

2.186 RECOVERY/endless.f File Reference

Functions/Subroutines

- program __endless.f__

2.186.1 Function/Subroutine Documentation

2.186.1.1 program __endless.f__ ( )

2.187 RECOVERY/GHOSTS.f File Reference

Functions/Subroutines

- program __ghosts.f__

2.187.1 Function/Subroutine Documentation

2.187.1.1 program __ghosts.f__ ( )

2.188 RECOVERY/REFRESH_SUMFILES.f File Reference

Functions/Subroutines

- program __refresh_sumfiles.f__

2.188.1 Function/Subroutine Documentation

2.188.1.1 program __refresh_sumfiles.f__ ( )

2.189 SCRIPT_MAKERS/find_nofit.f File Reference

Functions/Subroutines

- program __find_nofit.f__

  Procedure find_nofit.

2.189.1 Function/Subroutine Documentation

2.189.1.1 program __find_nofit.f__ ( )

Procedure find_nofit.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.190 SCRIPT_MAKERS/find_nofitP.f File Reference

Functions/Subroutines

- program __find_nofitp.f__

  Procedure find_nofitP.

2.190.1 Function/Subroutine Documentation

2.190.1.1 program __find_nofitp.f__ ( )

Procedure find_nofitP.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.191 SCRIPT_MAKERS/find_nofitT.f File Reference

Functions/Subroutines

- program __find_nofitt.f__

  Procedure find_nofitT.

2.191.1 Function/Subroutine Documentation

2.191.1.1 program __find_nofitt.f__ ( )

Procedure find_nofitT.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.192 SCRIPT_MAKERS/make_scriptA.f File Reference

Functions/Subroutines

- program __make_scripta.f__

  Procedure make_scriptA.

2.192.1 Function/Subroutine Documentation

2.192.1.1 program __make_scripta.f__ ( )

Procedure make_scriptA.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.193 SCRIPT_MAKERS/make_scriptAP.f File Reference

Functions/Subroutines

- program __make_scriptap.f__

  Procedure make_scriptAP.

2.193.1 Function/Subroutine Documentation

2.193.1.1 program __make_scriptap.f__ ( )

Procedure make_scriptAP.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.194 SCRIPT_MAKERS/make_scriptF.f File Reference

Functions/Subroutines

- program __make_scriptf.f__

  Procedure make_scriptF.

2.194.1 Function/Subroutine Documentation

2.194.1.1 program __make_scriptf.f__ ( )

Procedure make_scriptF.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.195 SCRIPT_MAKERS/make_scriptP.f File Reference

Functions/Subroutines

- program __make_scriptp.f__

  Procedure make_scriptP.

2.195.1 Function/Subroutine Documentation

2.195.1.1 program __make_scriptp.f__ ( )

Procedure make_scriptP.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.196 SCRIPT_MAKERS/make_scriptR.f File Reference

Functions/Subroutines

- program __make_scriptr.f__

  Procedure make_scriptR.

2.196.1 Function/Subroutine Documentation

2.196.1.1 program __make_scriptr.f__ ( )

Procedure make_scriptR.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.197 SCRIPT_MAKERS/make_scriptT.f File Reference

Functions/Subroutines

- program __make_scriptt.f__

  Procedure make_scriptT.

2.197.1 Function/Subroutine Documentation

2.197.1.1 program __make_scriptt.f__ ( )

Procedure make_scriptT.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.198 SCRIPT_MAKERS/MAKE_TILES.f File Reference

Functions/Subroutines

- program __make_tiles.f__

  Procedure MAKE_TILES.

2.198.1 Function/Subroutine Documentation

2.198.1.1 program __make_tiles.f__ ( )

Procedure MAKE_TILES.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.199 SCRIPT_MAKERS/MAKE_TILESP.f File Reference

Functions/Subroutines

- program __make_tilesp.f__

  Procedure MAKE_TILESP.

2.199.1 Function/Subroutine Documentation

2.199.1.1 program __make_tilesp.f__ ( )

Procedure MAKE_TILESP.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.200 SCRIPT_MAKERS/sigmalist.f File Reference

Functions/Subroutines

- program __sigmalist.f__

2.200.1 Function/Subroutine Documentation

2.200.1.1 program __sigmalist.f__ ( )

2.201 SHAPER/bigmap.f File Reference

Functions/Subroutines

- program __bigmap.f__

  Procedure bigmap.
- subroutine pixinlmx (INFILE)

  Variable_I/O.

2.201.1 Function/Subroutine Documentation

2.201.1.1 program __bigmap.f__ ( )

Procedure bigmap.

Abstract

This procedure creates a large topo/albedo map with the same file structure as the maplets (see subroutine REA↩ D_MAP or WRITE_MAP abstract). The location of the map canter is specified by one of three choices:

```
p:  pixel/line location in a picture
l:  latitude and west longitude
m:  pixel line location in a map or maplet.
```

There is a fourth choice (i) that is experimental and is hidden from the menu. It will play no role in mission operations.

A second set of inputs contains the bigmap scale in km/px, the half-size (qsz), an integer random seed, and a maximum maplet scale in case lower resolution maplets are to be excluded from the

The program first determines the body-fixed vector to the map center and the approximate surface normal Uz. It then projects the shape model onto this surface and determines a second approximation to the normal Uz by fitting a plane to the heights. It repeats this process one more time. The new map coordinate frame is then oriented so that East is to the right.

The maplets to be used in the construction are taken from the first of the files BIGMAP.IN, LMRKLISTR.TXT or LMRKLIST.TXT found to exist. However, if LMKLX=.TRUE. in the INIT_LITHOS.TXT file, the program will read LMRKLISTX.TXT. If there are a great many maplets, the latter choice will speed up the procedure by determining whether a maplet needs to be used without having to open its .LMK file.

The bigmap reference plane has $(2*qsz+1)^{\wedge}2$ points at locations:

```
p(i,j) = V + j*Ux +i*Uy   (i,j = -qsz,qsz)
```

A line from each of these points in the normal (Uz) direction will pierce a number of maplets. For each i,j we keep track of the weighted accumulation of the heights to the piercing points, the squares of those heights, and the slopes and albedos of the maplets at the piercing points.

Once these arrays are filled, the average heights and albedos at each point of the reference plane are computed, as well as the standard deviation of the heights. This last provides a convenient measure ot the height uncertainty at each point, and a display SIGMAS.pgm provided a quick means of identifying possible problem areas.

If we choose to use the height averaged map just constructed, we are done now. If we are constructind bigmaps from fake CreatorP data, this is recommended. A display showing the gradients of the bigmap can be viewed to see if all looks well. If it dose, we may just as well stop here. However, we often continue on to determine the slope_averaged bigmap. We choose a small fraction of our average heights (usually 0.005) as conditioning heights and integrate over the averaged slopes as we did in the slope to height integration in LITHOS (see subroutines SLP2HGT and NNEIGHBORS). This tends to eliminage "cliffs" that sometimes occur at maplet boundaries. We are asked for a weight for the conditioning heights and we usually choose 0.025. We then iterate perhaps 5 to 10 times before exiting. A typical run (this one from Vesta) looks like this:

```
bigmap
1                          <- specify by lat/lon
-1.6, 4.0                  <- lat/lon values
.05, 500, 1234, 1          <- scale, qsx, seed, max maplet res
CLAUDI                     <- bigmap name
1                          <- choose slope integration
.005                       <- fraction of heights for conditioning
.025                       < conditioning weight
1                          <- iterate
1
1
1
1
1
1
0                          <- stop iteration
0                          <- no template
```

The last entry is one of three choices, the others letting the user mask out all but a circular or rectangular portion of the map - a rarely used feature but useful for presentation slides.

In addition to the bigmap, that goes into the MAPFILES directory, and the SIGMAS.pgm image, bigmap produces other files. USED_MAPS.TXT is a list of all maplets that overlap the bigmap and were used in its creation. INSI↩ DE.TXT is a list of all maplets that completely ovwerlap the bigmap. A file MAPNAME.LMK, of the same form as a maplet's landmark file goes into the BIGFILES directory. Finally, the file SIGMAS.TXT is updated to include a record showing the maximum and average standard deviation for the BIGMAP heights.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.201.1.2 subroutine pixinlmx (character∗72 *INFILE*)

Variable_I/O.

Variable I/O Description

ANS I User response to menu selections PICMN I User supplied picture name SEED I User supplied seed MXSCL I User supplied max scale BIGMAP I User supplied map name IMGPL I User supplied patch center Z1 I User supplied latitude in deg. Z2 I User supplied W longitude in deg. NAME I User supplied map name S0 I User supplied map scale QSZ I User supplied map pixel size ∗ 2 Q0 I User supplied half size

File_I/O

Filename I/O Description

SUMFILES/<PICNM>.SUM I Image <PICNM> summary file USED_MAPS.TXT INSIDE.TXT BIGMAP.IN I LM↩ RKLISTR.TXT I Used if BIGMAP.IN doesn't exist LMRKLIST.TXT I Used if LMRKLISTR.TXT doesn't exist SIGMA↩ S.TXT I SIGMAS.DAT O Temporary file used by RAW2PGM MAPFILES/<BIGMAP>.MAP O Big map file created by WRITE_MAP BIGFILES/<BIGMAP>.LMK O BIGLIST.TXT I USED_MAP.TXT I LMKFILES/<LMKFILE>.LMK I USED_PICS.TXT O

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called SLEN

SPC_subroutines_called GET_HEIGHTS IMGPL2VN ORIENT PATCH_COORDS RAW2PGM READ_HEADE↩ R READ_MAP SLP2HGT SHOW_SLOPES U2VN V2IMGPL WRITE_MAP

SPICELIB_functions_called RPD VDOT VNORM

SPICELIB_subroutines_called LATREC VHAT VMINUS VSUB

History 2012_10_31: INSIDE maplets with holes recognized 2013_01_11: Only maplets with scale <= MXSCL included in Bigmap 2013_01_11: All maplets included in INSIDE.TXT and USED_MAPS.TXT 2013_07_23: Header added, capitalization & compiler warnings fixed. 2013_09_26: Parameter EPS set to 1.D-05 to avoud speckling. 2013_11_18: TMPL(I,J,3) computed earlier in case average only map wanted. 2014_03_24: LMRKLISTX.TXT used as list if LMKLX=.TRUE. IN INIT_LITHOS.TXT. 2014_04_11: Albedo problem introduced by -Wall fixes fixed. 2014_05_01: Y and Z prefixes removed as prohibited inputs. 2014_05_05: Command line version and seed options added.

2.202 SHAPER/bigmapL.f File Reference

Functions/Subroutines

- program __bigmapl.f__

  Procedure bigmapL.

2.202.1 Function/Subroutine Documentation

2.202.1.1 program __bigmapl.f__ ( )

Procedure bigmapL.

Abstract

This procedure creates a large topo/albedo map with the same file structure as the maplets (see subroutine REA↩ D_MAP or WRITE_MAP abstract). This is a "light" versiomn of bigmap, designed to be run in parallel to make many bigmaps at once with the script maker MAKE_TILESP. Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.203 SHAPER/densify.f File Reference

Functions/Subroutines

- program __densify.f__

  Procedure densify.
- subroutine put_model (OUTFILE, Q, VEC)

  Variable_I/O.

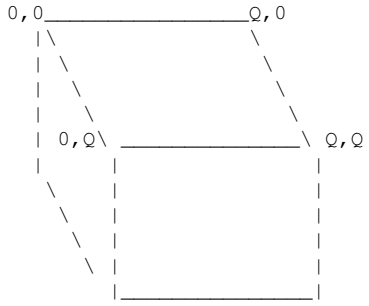
2.203.1 Function/Subroutine Documentation

2.203.1.1 program __densify.f__ ( )

Procedure densify.

Abstract

In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!software, the surface points of the model are topologically equivalent to grid points on the faces of a cube. The grid labels run from 0 to Q, with Q a power of 2.




```
The DENSIFY program multiplies an initial Q by a factor K=2^n, so that Q -> KQ
and points of the grid are now spaced by DQ=K.  A cell of the original model on
one of the new grids has a spacing of one unit, but has vectors defined every DQ
units (o).  We usually take K=2 and construct several models at increasingly
high resolution.

           ---------------------> DQ
           o    .    x    .    o

           .    .    .    .    .

           x    .    x    .    x

           .    .    .    .    .

           o    .    x    .    o

The subroutine INTERPOLATE defines new vectors at points spaced by DQ/2 (x),
```

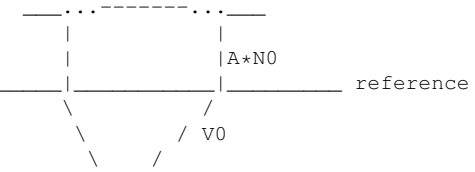
averaging the four corner vectors to obtain the central one, and the two
endpoints of each edge to get the midpoint vector. The subroutine returns these
vectors and divides DQ by 2. DENSIFY tests to see if DQ equal 1 and, if
not, calls the subroutine again. Now the upper left quarter of the
previous diagram looks like:

```
          ---------> DQ
          o    x    o

          x    x    x

          o    x    o
```

and when the interpolation is complete, all vectors are defined, the
subroutine returns a value DQ=1 and DENSIFY goes on about its business.

It should be pointed out that the shape coming out of INTERPOLATE is
the starting shape for the DENSIFY program, that uses the maplet ensemble
to create a denser (ie higher resolution) shape model.

The program densify first constructs a reference surface by interpolating
the surface points of a lower resolution shape model. At each point of
the reference there is a vector V0 from the model center to that point and
a normal N0 to the surface. That normal is extended some distance until
it pierces one or more of the ensemble of maplets, and the average A of
those distances is taken to represent the piercing point on the new
model's surface, so the new surface vector is V = V0 + A*N0. Because,



especially at the early stages, there are mismatches in maplet locations
simply due to the formal uncertainties of the estimation process, we have
found it better to average the maplet normals N at each point, keeping a
small randomly selected set of the A as conditioning heights. If Wa is
the average maplet normal between two neighboring points a and o, then
(Va - Vo).Wa = 0 = (Va0+Aa*Na0).Wa - (Vo0+Ao*No0).Wa so, as we did in the
slope to height determination in subroutines SLP2HGT and NNEIGHBORS, we
determine Ao from the average of Ao = (Va0.Wa-Vo0.Wa+Aa*Na0.Wa)/NoO.Wa
over the four neighboring points (a,b,d,d), including a random
constraining altitude Ax with a weight wx:

```
Ao = [(Va0-Vo0+Aa*Na0).Wa/No0.Wa +(Vb0-Vo0+Ab*Nb0).Wb/No0.Wb
   +(Vc0-Vo0+Ac*Nc0)/No0.W!!+(Vd0-Vo0+Ab*Nb0).Wd/NoO.Wd + wx*Ax]/(4+wx)
```

This is only one of many iterations involved in the relaxation process.

The initial inputs to the procedure are the names of the input and output SHAPE
file along with a line with K (usually 2), a limit in km specifying how far
along the surface normal the program should search for a maplet, and a random
seed in the form of a large integer. This seed will be superceeded if one is
entered on the command line following the densify invocation, eg

```
     densify 19372
```

In most cases, the maplets will cover most of the surface. Where it is not
covered the normals to the input model provide the the "slopes" and the
integration procedes without any randomly chosen conditioning heights from these
areas. In some cases, such as fast flybys of small bodies, only a small fraction
of the surface is visible - vast areas are unknown. In these cases, conditioning
heights are also taken from the input model as well. This option is specified by
using a negative value for K (usually -2).

The procedure now determines the average height along each surface normal from
each reference point of the densified shape. It also determines the average
maplet surface normal and the standard deviation of the heights, used as a
measure of uncertainty. It produces the output SHAPE file and a similar file
called SIGMAS.TXT that has an extra column representing the uncertainty. This
latter file can be displayed as an image to show areas that might need further

```
work.  The program now gives the options:

     0. end program
     1. proceed to iteration

If '0' is chosen, the output shape model will be the height averaged result.
The entire script, with the output model called SHAPEX.TXT, is:

     densify
     SHAPEFILES/SHAPE1.TXT        < input shape
     2, 1.0, 5639                 < K, search range (km), random SEED
     SHAPEFILES/SHAPEX.TXT        < output shape
     0                            < end program

If '1' is chosen, the program asks for:

     input fraction

the fraction of averaged heights (and empty heights if K<0) used to condition
the integration (usually .005) and:

     input weight

the weight given to the conditioning heights (usually .025).  There is an
experimental feature in this version of densify.  If the program LIMBER has been
run to generate limb points, the file LIMBVECS.TXT will be read to provide a
conditioning height at each limb point.  If this feature is not desired, simply
type "mv LIMBVECS.TXT LIMBVECS0.TXT" or "rm -f LIMBVECS.TXT" before starting.

The program produces an output SHAPE file and gives the options:

     0. end program
     1. more iteration
     2. change weight

The interim shape can be viewed to see whether we want to change the weight given to
the conditioning heights using a procedure described below.  Usually, we just
continue iterating or, finally, exit the program.  The entire script, with the
output model called SHAPEX.TXT, is:

     densify
     SHAPEFILES/SHAPE1.TXT        < input shape
     2, 1.0, 5639                 < K, search range (km), random SEED
     SHAPEFILES/SHAPEX.TXT        < output shape
     1                            < more iteration
     .005                         < fraction of points used for conditioning
     .025                         < conditioning weight
     1
     1
     1                            < more iterations
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     0                            < end program

After the program terminates, we can run:

     shape_info
     SHAPEFILES/SHAPEX.TXT

that will output the surface area, volume, moment of inertia tensor per unit mass
and center of figure - center of mass offset.

At any time we can also run:
```

```
     shape2maps
     SHAPEFILES/SHAPEX.TXT

that will produce input files for the program view_shape so we can see what the
shape looks like.  This is what we would do if we wanted to check whether the
conditioning weights needed to be changed.

By tradition, we usually use SHAPE0 for Q=64, SHAPE1 for Q=128, SHAPE2 for Q = 256
and SHAPE3 for Q=512, so after this run, if there have been no problems, we want to:

     cp SHAPEFILES/SHAPEX.TXT SHAPEFILES/SHAPE2.TXT

After another densification, when SHAPEX.TXT is Q=512, we would:

     cp SHAPEFILES/SHAPEX.TXT SHAPEFILES/SHAPE3.TXT

SHAPE.TXT is also Q=512 and we want to update that as well.  For historical reasons,
some scripts change permissions on SHAPE.TXT to read only, so we want to:

     chmod +w SHAPE.TXT
     cp SHAPEFILES/SHAPEX.TXT SHAPEFILES/SHAPE.TXT
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.203.1.2 subroutine put_model (character∗72 *OUTFILE,* integer∗4 *Q,* double precision, dimension(3,0:512,0:512,6) *VEC*)

Variable_I/O.

Variable I/O Description

INFILE I User supplied name of shape model file to be read. KDNS I TOL I SEED I OUTFILE I User supplied name of new shape model file. K I User supplied choice to menu selection. Z1 I User supplied fraction or weight depending on menu LMRKNM O Landmark name being processed (written to console)

File_I/O

Filename I/O Description

LMRKLISTR.TXT I List of landmarks. LMRKLIST.TXT I List of landmarks to be used if LMRKLISTR.TXT does not exist. SHAPEFILES/SIGMA.TXT O List of sigma values associated with shape model. OUTFILE O New shape model file in ICQ format written out by PUT_MODEL LMFILE I Landmark file to be read in by READ_MAP. INFILE I Current shape mode to be read by GET_MODEL.

Restrictions None

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called MAXLEN RANN

SPC_subroutines_called CORNERS EXTEND GET_MODEL INTERPOLATE MATCHUP NORM2ALT PUT_MO↩ DEL READ_MAP

SPICELIB_functions_called VDOT VNORM

SPICELIB_subroutines_called UCRSS VCRSS VADD VEQU VHAT VSUB

History 2013_08_07: Program modified to used SP!!library subroutines. 2013_08_19: Header added, capitalization,

compiler warnings fixed. 2014_05_02: Conditioning with LIMBVECS.TXT from LIMBER 2014_05_06: Command line version and seed options added.

2.204 SHAPER/densifya.f File Reference

Functions/Subroutines

- program __densifya.f__

  Procedure densifya.
- subroutine put_model (OUTFILE, Q, VEC, A)

  Variable_I/O.

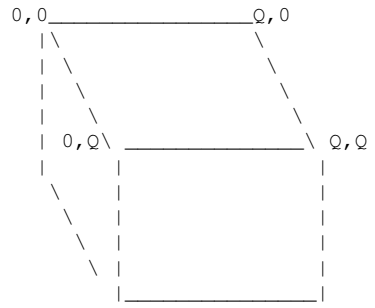
2.204.1 Function/Subroutine Documentation

2.204.1.1 program __densifya.f__ ( )

Procedure densifya.

Abstract

In the labeling scheme for the implicitly connected quadrilateral (ICQ) representation used by the shape model in the SP!!softwate, the surface points of the model are topologically equivalent to grid points on the faces of a cube. The grid labels run from 0 to Q, with Q a power of 2.



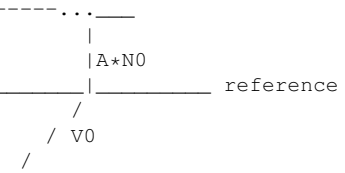

```
The DENSIFYA program multiplies an initial Q by a factor K=2^n, so that Q -> KQ
and points of the grid are now spaced by DQ=K.  A cell of the original model on
one of the new grids has a spacing of one unit, but has vectors defined every DQ
units (o).  We usually take K=2 and construct several models at increasingly
high resolution.

           --------------------> DQ
           o    .    x    .    o

           .    .    .    .    .

           x    .    x    .    x

           .    .    .    .    .

           o    .    x    .    o

The subroutine INTERPOLATE defines new vectors at points spaced by DQ/2 (x),
averaging the four corner vectors to obtain the central one, and the two
endpoints of each edge to get the midpoint vector.  The subroutine returns these
vectors and divides DQ by 2.  DENSIFY tests to see if DQ equal 1 and, if
not, calls the subroutine again.  Now the upper left quarter of the
previous diagram looks like:

           ---------> DQ
           o    x    o
```

```
          x    x    x

          o    x    o
```

and when the interpolation is complete, all vectors are defined, the
subroutine returns a value DQ=1 and DENSIFY goes on about its business.

It should be pointed out that the shape coming out of INTERPOLATE is
the starting shape for the DENSIFY program, that uses the maplet ensemble
to create a denser (ie higher resolution) shape model.

The program densify first constructs a reference surface by interpolating
the surface points of a lower resolution shape model. At each point of
the reference there is a vector V0 from the model center to that point and
a normal N0 to the surface. That normal is extended some distance until
it pierces one or more of the ensemble of maplets, and the average A of
those distances is taken to represent the piercing point on the new
model's surface, so the new surface vector is V = V0 + A*N0. Because,



especially at the early stages, there are mismatches in maplet locations
simply due to the formal uncertainties of the estimation process, we have
found it better to average the maplet normals N at each point, keeping a
small randomly selected set of the A as conditioning heights. If Wa is
the average maplet normal between two neighboring points a and o, then
(Va - Vo).Wa = 0 = (Va0+Aa*Na0).Wa - (Vo0+Ao*No0).Wa so, as we did in the
slope to height determination in subroutines SLP2HGT and NNEIGHBORS, we
determine Ao from the average of Ao = (Va0.Wa-Vo0.Wa+Aa*Na0.Wa)/NoO.Wa
over the four neighboring points (a,b,d,d), including a random
constraining altitude Ax with a weight wx:

```
Ao = [(Va0-Vo0+Aa*Na0).Wa/No0.Wa +(Vb0-Vo0+Ab*Nb0).Wb/No0.Wb
   +(Vc0-Vo0+Ac*Nc0)/No0.W!!+(Vd0-Vo0+Ab*Nb0).Wd/NoO.Wd + wx*Ax]/(4+wx)
```

This is only one of many iterations involved in the relaxation process.

The initial inputs to the procedure are the names of the input and output SHAPE
file along with a line with K (usually 2), a limit in km specifying how far
along the surface normal the program should search for a maplet, and a random
seed in the form of a large integer. This seed will be superceeded if one is
entered on the command line following the densify invocation, eg

```
     densifya 19372
```

In most cases, the maplets will cover most of the surface. Where it is not
covered the normals to the input model provide the the "slopes" and the
integration procedes without any randomly chosen conditioning heights from these
areas. In some cases, such as fast flybys of small bodies, only a small fraction
of the surface is visible - vast areas are unknown. In these cases, conditioning
heights are also taken from the input model as well. This option is specified by
using a negative value for K (usually -2).

The procedure now determines the average height along each surface normal from
each reference point of the densified shape. It also determines the average
maplet surface normal and the average albedo from the maplets,. It produces the
output SHAPEA file in the standard ICQ format with an extra column for albedo.
The program now gives the options:

```
     0. end program
     1. proceed to iteration
```

If '0' is chosen, the output shape model will be the height averaged result.
The entire script, with the output model called SHAPEA.TXT, is:

```
     densifya
     SHAPEFILES/SHAPE2.TXT        < input shape
```

```
     2, 1.0, 8937                < K, search range (km), random SEED
     SHAPEFILES/SHAPEA.TXT       < output shape
     0                           < end program
```

If '1' is chosen, the program asks for:

```
     input fraction
```

the fraction of averaged heights (and empty heights if K<0) used to condition
the integration (usually .005) and:

```
     input weight
```

the weight given to the conditioning heights (usually .025). There is an
experimental feature in this version of densify. If the program LIMBER has been
run to generate limb points, the file LIMBVECS.TXT will be read to provide a
conditioning height at each limb point. If this feature is not desired, simply
type "mv LIMBVECS.TXT LIMBVECS0.TXT" or "rm -f LIMBVECS.TXT" before starting.

The program produces an output SHAPE file and gives the options:

```
     0. end program
     1. more iteration
     2. change weight
```

The interim shape can be viewed to see whether we want to change the weight given to
continue iterating or, finally, exit the program. The entire script, with the
output model called SHAPEA.TXT, is:

```
     densifya
     SHAPEFILES/SHAPE2.TXT       < input shape
     2, 1.0, 8937                < K, search range (km), random SEED
     SHAPEFILES/SHAPEA.TXT       < output shape
     1                           < more iteration
     .005                        < fraction of points used for conditioning
     .025                        < conditioning weight
     1
     1
     1                           < more iterations
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     1
     0                           < end program
```

After the program terminates, we can run:

```
     shape_info
     SHAPEFILES/SHAPEA.TXT
```

that will output the surface area, volume, moment of inertia tensor per unit mass
and center of figure - center of mass offset.

At any time we can also run:

```
     shape2mapsA
     SHAPEFILES/SHAPEA.TXT
```

that will produce input files for the program view_shapeA so we can see what the
shape looks like. This is what we would do if we wanted to check whether the
conditioning weights needed to be changed.

By tradition, we usually use SHAPE0 for Q=64, SHAPE1 for Q=128, SHAPE2 for Q = 256
and SHAPE3 for Q=512. We hardly ever make a SHAPEA that is not Q=512. This is the
prefered input into LITHOS for filling new maplets with shape data, since it
approximates the albedo as well. In order to update SHAPE3 from SHAPEA, we use the

```
dumber porgram that usually lowers the Q of the model by factors oy 2.  In the script
below, it leaves Q unchanged but removes the fourth "albedo" column:

     dumber
     SHAPEFILES/SHAPEA.TXT
     SHAPEFILES/SHAPE3.TXT
     1                           < reduce Q by factor of 1
     y                           < remove albedo

SHAPE.TXT is also Q=512 and we want to update that as well.  For historical reasons,
some scripts change permissions on SHAPE.TXT to read only, so we want to:

     chmod +w SHAPE.TXT
     cp SHAPEFILES/SHAPE3.TXT SHAPEFILES/SHAPE.TXT
```

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.204.1.2 subroutine put_model (character∗72 *OUTFILE,* integer∗4 *Q,* double precision, dimension(3, 0:512, 0:512,6) *VEC,* double precision, dimension(0:512, 0:512,6) *A*)

Variable_I/O.

Variable I/O Description

INFILE I User supplied name of shape model file to be read. KDNS I TOL I SEED I OUTFILE I User supplied name of new shape model file. K I User supplied choice to menu selection. Z1 I User supplied fraction or weight depending on menu LMRKNM O Landmark name being processed (written to console)

File_I/O

Filename I/O Description

LMRKLISTR.TXT I List of landmarks. LMRKLIST.TXT I List of landmarks to be used if LMRKLISTR.TXT does not exist. SHAPEFILES/SIGMA.TXT O List of sigma values associated with shape model. OUTFILE O New shape model file in ICQ format written out by PUT_MODEL LMFILE I Landmark file to be read in by READ_MAP. INFILE I Current shape mode to be read by GET_MODEL.

Restrictions

Software_Documentation

OSIRIS-REx Stereophotoclinometry Software Design Document OSIRIS-REx Stereophotoclinometry Software User's Guide

Author_and_Institution

R.W. Gaskell (PSI)

Version

SPC_functions_called RANN

SPC_subroutines_called CONRNER EXTEND GET_MODEL INTERPOLATE MATCHUP NORM2ALT PUT_MO↩ DEL READ_MAP

SPICELIB_functions_called UCRSS VCRSS VEQU VHAT VSUB

SPICELIB_subroutines_called VDOT VNORM

History 2013_08_07: Program modified to used SP!!library subroutines. 2013_08_19: Header added, capitalization, compiler warnings fixed. 2014_05_01: Y and Z prefixes removed as prohibited inputs. 2014_05_02: Conditioning with LIMBVECS.TXT from LIMBER 2014_05_06: Command line version and seed options added.

2.205 SHAPER/dumber.f File Reference

Functions/Subroutines

- program __dumber.f__
  ftn dumber.f -o dumber.e

2.205.1 Function/Subroutine Documentation

2.205.1.1 program __dumber.f__ ( )

ftn dumber.f -o dumber.e

2.206 SHAPER/ll2vec.f File Reference

Functions/Subroutines

- program __ll2vec.f__

2.206.1 Function/Subroutine Documentation

2.206.1.1 program __ll2vec.f__ ( )

2.207 SHAPER/shape2maps.f File Reference

Functions/Subroutines

- program __shape2maps.f__
  ftn shape2maps.f -o shape2maps.e

2.207.1 Function/Subroutine Documentation

2.207.1.1 program __shape2maps.f__ ( )

ftn shape2maps.f -o shape2maps.e

2.208 SHAPER/shape2mapsA.f File Reference

Functions/Subroutines

- program __shape2mapsa.f__
  ftn shape2mapsA.f -o shape2mapsA.e

2.208.1 Function/Subroutine Documentation

2.208.1.1 program __shape2mapsa.f__ ( )

ftn shape2mapsA.f -o shape2mapsA.e

2.209 SHAPER/shape_info.f File Reference

Functions/Subroutines

- program __shape_info.f__
  Procedure shape_info.

2.209.1 Function/Subroutine Documentation

2.209.1.1 program __shape_info.f__ ( )

Procedure shape_info.

Abstract

The procedure shape_info produces useful information about a shape model by integrating over (adding) the pyramids with bases of quadrilateral facets with appropriate weights. The volume, surface area and center of figure position (odffset) are provided, all in kilometer units. Note that an offset does not necessarily mean that we want to run the procedute SHIFT to bring the center of figure to the center of the coordinate system. For example, Vesta showed such !! an offset of about 1.5 km that really was an offset, because Vesta is not a homogeneous body.

Assuming that the body is homogeneous, the procedure also computes the moment of inertial tensor (per unit mass). It makes no attempt to diagonalize the tensor and rotate to the principal axis frame.

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.210 SHAPER/spheremapsA.f File Reference

Functions/Subroutines

- program __spheremapsa.f__
  ftn spheremapsA.f /Users/rgaskell/toolkit/lib/spicelib.a -o spheremapsA.e
- real ∗8 function rann (seed)
- subroutine read_map (LMRKFILE, NTMP, QSZ, SCALE, V, UX, UY, UZ, HT, ALB)
- subroutine flip (n, lflag, ch1, ch2)
- integer ∗4 function slen (STRING)
- subroutine raw2ppm (infile, outfile, npx, nln)
- subroutine v2imgpl (V, V0, PICNM, LS, RD, NPX, NLN, MMFL, CTR, KMAT, D, CX, CY, CZ, USE, IMGPL)
- subroutine mmpx (KMAT, CTR, D, MM, PX)
- subroutine getmm (picnm, mmfl, z1, z2, z3, z)
- subroutine find_picfile (PICNM, PICFILE, EX)
- subroutine scobj (R0, U0, TRJ_AXIS, TRJ, POL_AXIS, POL, LN0, LN, V)
- subroutine pointing (CX, CY, CZ, CAX, IMAX, TRJ_AXIS, TRJ, POL_AXIS, POL, LN0, LN, WX, WY, WZ)
- subroutine rot2mat (V, M)
- double precision function cpx (PICNM, NPX, NLN, CPIC, L)
- logical function onpx (PICNM, NPX, NLN, P, L)

2.210.1 Function/Subroutine Documentation

2.210.1.1 program __spheremapsa.f__ ( )

ftn spheremapsA.f /Users/rgaskell/toolkit/lib/spicelib.a -o spheremapsA.e

2.210.1.2 double precision function cpx (character∗12 *PICNM,* integer *NPX,* integer *NLN,* double precision, dimension(2) *CPIC,* integer *L*)

2.210.1.3 subroutine find_picfile (character∗12 *PICNM,* character∗72 *PICFILE,* logical *EX*)

2.210.1.4 subroutine flip (integer∗4 *n,* logical *lflag,* character∗(∗) *ch1,* character∗(∗) *ch2*)

2.210.1.5 subroutine getmm (character∗12 *picnm,* double precision *mmfl,* double precision *z1,* double precision *z2,* double precision *z3,* double precision, dimension(2) *z*)

2.210.1.6 subroutine mmpx (double precision, dimension(2,3) *KMAT,* double precision, dimension(2) *CTR,* double precision, dimension(4) *D,* double precision, dimension(2) *MM,* double precision, dimension(2) *PX*)

2.210.1.7 logical function onpx (character∗12 *PICNM,* integer *NPX,* integer *NLN,* integer *P,* integer *L*)

2.210.1.8 subroutine pointing (double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* double precision, dimension(20,3) *CAX,* integer *IMAX,* double precision, dimension(3) *TRJ_AXIS,* double precision, dimension(3) *TRJ,* double precision, dimension(3) *POL_AXIS,* double precision *POL,* double precision *LN0,* double precision *LN,* double precision, dimension(3) *WX,* double precision, dimension(3) *WY,* double precision, dimension(3) *WZ*)

2.210.1.9 real∗8 function rann (real∗8 *seed*)

2.210.1.10 subroutine raw2ppm (character∗72 *infile,* character∗72 *outfile,* integer∗4 *npx,* integer∗4 *nln*)

2.210.1.11 subroutine read_map (character∗72 *LMRKFILE,* integer *NTMP,* integer *QSZ,* double precision *SCALE,* double precision, dimension(3) *V,* double precision, dimension(3) *UX,* double precision, dimension(3) *UY,* double precision, dimension(3) *UZ,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *HT,* real∗4, dimension(-ntmp:ntmp,-ntmp:ntmp) *ALB*)

2.210.1.12 subroutine rot2mat (real∗8, dimension(3) *V,* real∗8, dimension(3,3) *M*)

2.210.1.13 subroutine scobj (double precision, dimension(3) *R0,* double precision, dimension(3) *U0,* double precision, dimension(3) *TRJ_AXIS,* double precision, dimension(3) *TRJ,* double precision, dimension(3) *POL_AXIS,* double precision *POL,* double precision *LN0,* double precision *LN,* double precision, dimension(3) *V*)

2.210.1.14 integer∗4 function slen (character∗(∗) *STRING*)

2.210.1.15 subroutine v2imgpl (double precision, dimension(3) *V,* double precision, dimension(3) *V0,* character∗12 *PICNM,* logical *LS,* logical *RD,* integer *NPX,* integer *NLN,* double precision *MMFL,* double precision, dimension(2) *CTR,* double precision, dimension(2,3) *KMAT,* double precision, dimension(4) *D,* double precision, dimension(3) *CX,* double precision, dimension(3) *CY,* double precision, dimension(3) *CZ,* logical *USE,* double precision, dimension(2) *IMGPL*)

2.211 SHAPER/spheremapsB.f File Reference

Functions/Subroutines

- program __spheremapsb.f__

  Procedure spheremapsB.

2.211.1 Function/Subroutine Documentation

2.211.1.1 program __spheremapsb.f__ ( )

Procedure spheremapsB.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.212 SHAPER/triax.f File Reference

Functions/Subroutines

- program __triax.f__
- subroutine xyf2u (q, x, y, face, ax, v)
- subroutine put_model (q, vec, shapefile)
- subroutine defu (u)

2.212.1 Function/Subroutine Documentation

2.212.1.1 program __triax.f__ ( )

2.212.1.2 subroutine defu (integer∗4, dimension(3,3,6) *u*)

2.212.1.3 subroutine put_model (integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec,* character∗72 *shapefile*)

2.212.1.4 subroutine xyf2u (integer∗4 *q,* real∗8 *x,* real∗8 *y,* integer∗4 *face,* real∗8, dimension(3) *ax,* real∗8, dimension(3) *v*)

2.213 SHAPER/vecs2cube.f File Reference

Functions/Subroutines

- program __vecs2cube.f__
- subroutine invertn (N, M, MINV)
- real ∗8 function dot (v1, v2)
- subroutine ucross (v1, v2, v, mag)
- subroutine unit (v, mag)
- subroutine xyf2u (q, x, y, face, ax, v)
- subroutine put_model (outfile, q, vec)
- subroutine defu (u)

2.213.1 Function/Subroutine Documentation

2.213.1.1 program __vecs2cube.f__ ( )

2.213.1.2 subroutine defu (integer∗4, dimension(3,3,6) *u*)

2.213.1.3 real∗8 function dot (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2*)

2.213.1.4 subroutine invertn (integer∗4 *N,* real∗8, dimension(6,6) *M,* real∗8, dimension(6,6) *MINV*)

2.213.1.5 subroutine put_model (character∗72 *outfile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.213.1.6 subroutine ucross (real∗8, dimension(3) *v1,* real∗8, dimension(3) *v2,* real∗8, dimension(3) *v,* real∗8 *mag*)

2.213.1.7 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.213.1.8 subroutine xyf2u (integer∗4 *q,* real∗8 *x,* real∗8 *y,* integer∗4 *face,* real∗8, dimension(3) *ax,* real∗8, dimension(3) *v*)

2.214 SHAPER/vecs2shape.f File Reference

Functions/Subroutines

- program __vecs2shape.f__
- subroutine put_model (outfile, q, vec)
- subroutine xyf2u (sz, x, y, face, ax, v)
- subroutine defu (u)
- real ∗8 function p (L0, M0, PHI)
- subroutine unit (v, mag)
- subroutine binv (N, M, MINV)
- subroutine matchup (sz, i1, j1, f1, i2, j2, f2)
- subroutine corners (sz, i1, j1, f1, i2, j2, f2, i3, j3, f3)

2.214.1 Function/Subroutine Documentation

2.214.1.1 program __vecs2shape.f__ ( )

2.214.1.2 subroutine binv (integer∗4 *N,* real∗8, dimension(2500,2500) *M,* real∗8, dimension(2500,2500) *MINV*)

2.214.1.3 subroutine corners (integer∗4 *sz,* integer∗4 *i1,* integer∗4 *j1,* integer∗4 *f1,* integer∗4 *i2,* integer∗4 *j2,* integer∗4 *f2,* integer∗4 *i3,* integer∗4 *j3,* integer∗4 *f3*)

2.214.1.4 subroutine defu (integer∗4, dimension(3,3,6) *u*)

2.214.1.5 subroutine matchup (integer∗4 *sz,* integer∗4 *i1,* integer∗4 *j1,* integer∗4 *f1,* integer∗4 *i2,* integer∗4 *j2,* integer∗4 *f2*)

2.214.1.6 real∗8 function p (integer *L0,* integer *M0,* real∗8 *PHI*)

2.214.1.7 subroutine put_model (character∗72 *outfile,* integer∗4 *q,* real∗8, dimension(3,0:512,0:512,6) *vec*)

2.214.1.8 subroutine unit (real∗8, dimension(3) *v,* real∗8 *mag*)

2.214.1.9 subroutine xyf2u (integer∗4 *sz,* real∗8 *x,* real∗8 *y,* integer∗4 *face, ax,* real∗8, dimension(3) *v*)

2.215 UTILITIES/duplicates.f File Reference

Functions/Subroutines

- program __duplicates.f__

  Procedure duplicates.

2.215.1 Function/Subroutine Documentation

2.215.1.1 program __duplicates.f__ ( )

Procedure duplicates.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.216 UTILITIES/EUL_I2PM.f File Reference

Functions/Subroutines

- program __eul_i2pm.f__

  Procedure EUL_I2PM.

2.216.1 Function/Subroutine Documentation

2.216.1.1 program __eul_i2pm.f__ ( )

Procedure EUL_I2PM.

2.217 UTILITIES/EXPORT.f File Reference

Functions/Subroutines

- program __export.f__

  Procedure EXPORT.

2.217.1 Function/Subroutine Documentation

2.217.1.1 program __export.f__ ( )

Procedure EXPORT.

Abstract The program will generate two files, EXPORT.B and EXPORT.TXT that are used to create the tar file, NEW_FILES.TAR, of new landmark and maplet files for exporting to a different computer system.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.218 UTILITIES/IMGLATLON.f File Reference

Functions/Subroutines

- program __imglatlon.f__

  PROGRAM IMGLATLON.

2.218.1 Function/Subroutine Documentation

2.218.1.1 program __imglatlon.f__ ( )

PROGRAM IMGLATLON.

FINDS IMAGES WHOSE CENTER LIES IN A GIVEN RANGE OF LATITUDES AND EAST LONGITUDES.

2.219 UTILITIES/IMPORT.f File Reference

Functions/Subroutines

- program __import.f__

  Procedure ftn IMPORT.

2.219.1 Function/Subroutine Documentation

2.219.1.1 program __import.f__ ( )

Procedure ftn IMPORT.

Abstract This program is used in conjunction with EXPORT to move newly generated files from one computer system to another.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.220 UTILITIES/make_list.f File Reference

Functions/Subroutines

- program __make_list.f__

  Procedure.

2.220.1 Function/Subroutine Documentation

2.220.1.1 program __make_list.f__ ( )

Procedure.

ftn make_list

Abstract This routine will read the image names from the LIMBLIST1.TXT file and document any matches of those images names with the images names or PICTURES in the current list of landmark. The documentation consists of the landmark name and the number of matching images with the LIMBLIST1.TXT

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.221 UTILITIES/MAKE_LMRKLISTX.f File Reference

Functions/Subroutines

- program __make_lmrklistx.f__
  Procedure ftn MAKE_LMRKLISTX.

2.221.1 Function/Subroutine Documentation

2.221.1.1 program __make_lmrklistx.f__ ( )

Procedure ftn MAKE_LMRKLISTX.

Abstract This routine reads the landmarks in the LMRKLIST.TXT file and then from the corresponding landmark file read the size, scale, landmark position vector from the center of the body to center of the landmark and the number of pictures containing the landmark and writes these values out to the LMRKLISTX.TXT

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.222 UTILITIES/MAKE_PICTLISTX.f File Reference

Functions/Subroutines

- program __make_pictlistx.f__
  Procedure ftn MAKE_PICTLISTX.

2.222.1 Function/Subroutine Documentation

2.222.1.1 program __make_pictlistx.f__ ( )

Procedure ftn MAKE_PICTLISTX.

Abstract

This routine reads the picture names in the PICTLIST.TXT file and then from the data in the corresponding PIC↩ NM.SUM file computes the range to the body center (Z2), the cosine of half the maximum field of view (Z1), the unit vector to the body center (W) and the unit bore sight direction of the camera (C). These are wrttten to the PICTLISTX.TXT that has records:

```
1X,PICNM,Z2,Z1,W(1),W(2),W(3),C(1),C(2),C(3)
```

if a restricted list PICTLISTR.TXT the program produces an analagous file PICTLISTRX.TXT. The program LITHOS will search for these files in the order PICTLISTRX, PICTLISTR, PICTLISTX, PICTLIST and use the first one found unless the line RPLUSE=.FALSE. (RPL = Restricted Picture List) appears in the INIT_LITHOS.TXT file, in which care the restricted lists are ignored.

The purpose of the eXtended PICTLISTX file is to extract all the information necessary from the PICNM.SUM files so that those potentially thousands of files do not have to be opened in order to determine whether a maplet is visible in an image.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.223 UTILITIES/make_tilefile.f File Reference

Functions/Subroutines

- program __make_tilefile.f__

  Procedure make_tilefile.

2.223.1 Function/Subroutine Documentation

2.223.1.1 program __make_tilefile.f__ ( )

Procedure make_tilefile.

Abstract The make_tilefile analyses a coverage_m.pgm file, which is the output of map_coverage, for area that still need to be tiled. The output of this program is then put in the make_scriptT.in file.

Disclaimer None

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.224 UTILITIES/OMEGA_PLOT.f File Reference

Functions/Subroutines

- program __omega_plot.f__

2.224.1 Function/Subroutine Documentation

2.224.1.1 program __omega_plot.f__ ( )

2.225 UTILITIES/rem_done.f File Reference

Functions/Subroutines

- program __rem_done.f__

  Procedure rem_done.

2.225.1 Function/Subroutine Documentation

2.225.1.1 program __rem_done.f__ ( )

Procedure rem_done.

Abstract

Required_Reading

R.W. Gaskell, et.al, "Characterizing and navigating small bodies with imaging data", Meteoritics & Planetary Science 43, Nr 6, 1049-1061 (2008)

Declarations

2.226 UTILITIES/ROUGHNESS.f File Reference

Functions/Subroutines

- program __roughness.f__

  ftn -Wall ROUGHNESS.f /Users/rgaskell/toolkit/lib/spicelib.a -o ROUGHNESS.e

2.226.1 Function/Subroutine Documentation

2.226.1.1 program __roughness.f__ ( )

ftn -Wall ROUGHNESS.f /Users/rgaskell/toolkit/lib/spicelib.a -o ROUGHNESS.e

2.227 UTILITIES/SUM2NOM.f File Reference

Functions/Subroutines

- program __sum2nom.f__

2.227.1 Function/Subroutine Documentation

2.227.1.1 program __sum2nom.f__ ( )

2.228 UTILITIES/TEST_IPL2RDT.f File Reference

Functions/Subroutines

- program __test_ipl2rdt.f__

2.228.1 Function/Subroutine Documentation

2.228.1.1 program __test_ipl2rdt.f__ ( )

2.229 UTILITIES/Z_PLOT.f File Reference

Functions/Subroutines

- program __z_plot.f__

2.229.1 Function/Subroutine Documentation

2.229.1.1 program __z_plot.f__ ( )