



InSight
InSight (NSYT)
Software Interface Specification

SIS Title: *Camera Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products*

SIS ID: NSYT-PLD001

Project internal SIS Location (for life of mission):

<https://alpha-lib.jpl.nasa.gov/docushare/dsweb/View/Collection-161615>

Project Version: v3.3

Release Date: June 26, 2019

JPL Doc ID: JPL D-56937

Prepared by: Robert Deen, Payam Zamani, Hallie Abarca, Justin Maki

Custodian: Robert Deen

Mission Reference: InSight

Approvals

Generating Element

Hallie Abarca MIPL

Receiving Elements

Ashitey Trebi-Ollennu IDS

Project

Rick Welch MOS/GDS

Suzanne Smrekar
Project Science/Archive

Concurrence

Justin Maki Science Team Co-I

Elizabeth Barret Science Rep.

Susan LaVoie PDS Imaging Node

Raymond Arvidson
Susan Slavney PDS Geo-Sciences
Node

InSight Project

Software Interface Specification (SIS)

Camera Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products

Version 3.3

Custodian:

Robert Deen, JPL-MIPL

Authors:

Robert Deen, Payam Zamani, Hallie Abarca, Justin Maki

The research described in this document was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

© 2018, 2019 California Institute of Technology. Government sponsorship acknowledged.

JPL D-92454



Jet Propulsion Laboratory
California Institute of Technology

CHANGE LOG

DATE	SECTIONS CHANGED	REVISION
2014-04-12	All	V0.0
2014-06-16	All	V0.2
2014-08-10	Included feedback from Amy for PDS label and product levels. removed camera descriptions in anticipation of color camera.	V0.3
2015-05-11	Added new product codes, changed file names based on new IMAGE_ID. Added sections on camera details.	V0.4
2015-05-31	Added details about camera design	V0.5
2015-07-21	Added “group” column to product type table to allow sorting by group, or alphabetically Added “TAU” product type for tau reports.	V0.6
2015-08-25	Added diagrams for coordinate systems, added sample imaging labels	V1.0
2015-10-21	Included sample VICAR label and keyword table	V1.0
2016-02-08	All, updates to all sections made by Hallie, Bob, Justin and Payam	V 1.1
2016-3-30	Extensive updates throughout the document	V2
2017-10-12	Updated PDS levels for the product types per input from Cristina and Bob Added SRD to product type table	V2.1
2017-10-23	Minor changes throughout. Significant revisions to 2.3, 2.4, 4.1.4, 5.2.2.3, 5.2.2.4, 5.2.3.6.2, 5.2.4.1.6, 6.3, 7.1, 7.2, and Appendices.	V2.2
2017-11-20	Minor revisions to: 4.1.4, 5.1.1, 5.2.1.7, 5.2.3.2	V2.3
2017-11-21	Redesign of color products (affects 2.4.1, 2.4.2, table 4.4, 5.2.2.2.1, 5.2.2.3)	V2.3
2017-11-28	Revised description of XYZ frames in 5.2.3.2	V2.3
2017-11-28	Added XRM product type; changed several Derived to Calibrated in table 4.4 and adjusted colors of XYZ-related products	V2.3
2017-12-15	Revisions throughout to respond to PDS Peer Review (mostly typo/wording/clarification issues)	V2.4
2018-01-13	Fixed document references throughout and list in section 1.3	V2.4
2018-01-17	Updated acronym list	V2.4
2018-01-18	Updated several figures, reset change tracking	V2.4
2018-04-10	Updated WSH and WSS products to reflect new workspace polygon with new additional values (affects 5.2.4.2, and figure 5-24)	V2.5
2018-04-19	Redesign (again) of color products (affects 2.4.2, table 4.2 (filter), table 4.4, 5.1.1.2, 5.2.1.2, and an extensive rewrite of 5.2.2.3). Added figure 2-3.	V2.5
2018-04-21	Wrote new Calibration section (Appendix B); clarified rotation and companding of EDR’s (5.1.1).	V2.5
2018-05-07	Clarified that UIx and ZIx are in Site frame; changed ZZZ to Green in 4-4.	V2.5
2018-05-21	Added section 5.5 describing Browse images; changed DEM to Site frame.	V2.5
2018-05-22	Clarified that .xml replaces VIC or IMG but adds to other types	V2.5
2018-05-30	Rewrote mesh section to describe PDS mesh products (and better describe	V2.5

DATE	SECTIONS CHANGED	REVISION
	ops mesh products)	
2018-06-05	Revised the lander mech coord sys; added arm diagram and fixed text in 6.3	V2.5
2018-06-25	Added UP*, XI*, and RUK products to table 4-4 (and moved/renamed the instrument products to consolidate them), revamped 5.2.4.5 and 5.2.4.6 to include the new products, added 5.2.4.1.7 (on wts offset)	V2.5
2018-06-25	Updated figures 5-18 (point-persp mosaic), 5-8 and 5-25 (WSS overlay)	V2.5
2018-06-06	Bumped version, reset change bars. Replaced Figs 5-1, 5-16, 5-19, 5-20, 5-21, 5-23. Added Fig 5-8 (SRD) (which rennumbers all figure 5-x references above).	V2.6
2018-07-02	Updated 2.4 and 5.2.1.1, rewrote color (again) in 5.2.2.3. Changed names of P color products in table 4-4. Updated bandpasses in table 2-1	V2.6
2018-07-03	Resolved final TBDs. Reset change bars, bump version for Peer Review	V2.7
2018-07-18	In table 4-4 and 5.2.4.5, fixed XIx product types to be XPx, clarified definitions of UPx and XPx, and fixed typo on SEN	V2.7
2018-07-24	Clarified use of Sol/DOY and Epoch in filename conventions, and mosaic projection frame in filename.	V2.7
2018-07-26	Clarified use of version numbers in PDS (section 3.5)	V2.7
2018-07-30	Clarified mesh skin types in 5.3. Fixed keyword table description in App E	V2.7
2018-08-01	Added HTML tables to Appendix E.	V2.7
2018-08-01	Bumped version to 2.8 for the SIS, cleared change bars. Peer Review version.	V2.8
2018-08-22	Title page changes per docrev office: added acknowledgment, copyright, and NASA logo; removed JPL logo and "Paper copies..." note.	V2.8
2018-10-02	Numerous editorial revisions (mostly wording, clarity, and typos) based on Peer Review feedback. Added new section 7.2.	V3.0
2018-11-16	Removed TDS as a component of GUS; clarified which images are made into browse images (section 5.5); added more explanation of the keyword tables in Appendix E. Bumped version to 3.1.	V3.1
2018-11-17	Noted that EJPs have no PDS4 labels but are instead in the EDR label	V3.1
2018-12-03	Table 4-4: Added Derived Image Type column; added IDX, ICM products, changed description on CPx/MPx, fixed typos. Added section 5.14 (IDX,ICM). Revised Fig 5-4 to match post-landing pipeline change. Bumped version to 3.2.	V3.2
2019-05-02	Removed appendices C and D (example labels) as redundant, since there are "samples" in the data dir	V3.2
2019-05-31	Revised radiometric and color processing. Reformatted Table 4-4, added RAY, RZF, RZD, RZS, RZY, and the Dxx/Zxx color products. Removed DSG (disparity grid). Added 5.2.2.2.2 (zenith scaling) and 5.2.2.2.3 (float to int scaling). Rewrote 5.2.2.3 (new color products)	V3.3

TBD ITEMS

ITEM	ASSIGNEE
All TBD items are highlighted in yellow. None currently.	

TABLE OF CONTENTS

1.1 Purpose and Scope.....15

1.2 Contents.....15

1.3 Constraints and Applicable Documents16

 1.3.1 Relationships with Other Interfaces19

2. Instrument Overview 20

2.1 Instrument Context Camera (ICC)21

2.2 Instrument Deployment Camera (IDC).....22

2.3 Camera CCD Layout25

2.4 Camera Flight Software Processing26

 2.4.1 Shutter Subtraction26

 2.4.2 Responsivity Balancing26

 2.4.3 Bayer demosaicking (de-Bayering)26

 2.4.4 Color Correction Matrix27

 2.4.5 Companding27

3. Data Processing Overview 28

3.1 Data Processing Level28

3.2 Product Label and Header Descriptions.....29

3.3 EDR Data Product30

3.4 Data Flow30

3.5 RDR Data Product31

3.6 Data Validation32

4. Data Product Overview 34

4.1 Data Product Structure34

 4.1.1 VICAR Image Format34

 4.1.2 PDS4 Label Format35

 4.1.3 Mapping of PDS and VICAR Labels36

 4.1.4 Label References.....36

4.2 Binary Data Storage Conventions36

 4.2.1 Bit and Byte Ordering36

4.3 File Naming37

 4.3.1 EDR and Single-Frame RDR Filename Convention37

 4.3.2 Product Identifiers42

 4.3.3 Mosaic Filename Convention46

 4.3.4 Terrain Mesh Filename Convention50

5. Detailed Data Product Specifications 55

5.1 EDR Data Product Specification55

 5.1.1 Image EDR (EDR).....55

 5.1.1.1 Raw EDRs56

 5.1.1.2 Color EDRs.....57

5.1.1.3	Subframe, Downsampling, and Thumbnails	57
5.1.2	Original JPEG (EJP)	57
5.1.3	Reference Pixels (ERP)	57
5.2	Single-Image RDR Data Product Specification	59
5.2.1	Common Processing	59
5.2.1.1	Bayer color processing (BAY)	59
5.2.1.2	Color RDRs	59
5.2.1.3	Geometrically Corrected Images (Linearization).....	60
5.2.1.4	Image Overlays (**O).....	63
5.2.1.5	Filled Images (**F).....	63
5.2.1.6	Lander Volume Exclusion Image Masks (MXY, M**)	63
5.2.1.7	Masked Images (**M)	64
5.2.1.8	EDR-like RDRs	64
5.2.2	Image RDRs	64
5.2.2.1	Inverse LUT RDR (ILT, ILM, ILC, ILP).....	64
5.2.2.2	Radiometrically Corrected RDR (RA*, IO*, RI*, RZ*)	65
5.2.2.2.1	MIPLRAD Method.....	65
5.2.2.2.2	Zenith Scaling.....	66
5.2.2.2.3	Float to Integer Scaling	67
5.2.2.3	Color-Corrected Products (C**/D**/Z**/M**)	67
5.2.2.4	Anaglyph/Colorglyph RDR	71
5.2.3	Geometry RDRs.....	71
5.2.3.1	Disparity RDR (DSR, DSP, DFF, DDD, MDS)	71
5.2.3.1.1	Stereo Pair Matching Method.....	72
5.2.3.2	XYZ RDR (XYZ, XYR, XYE, DEM).....	73
5.2.3.3	Range RDR (RNG, RNR, RNE)	75
5.2.3.4	Surface Normal RDR (UVW, UVS).....	76
5.2.3.5	“Surface Roughness” RDR (RUF).....	78
5.2.3.6	Slope RDRs (SLP, SRD, SHD, SMG, SNT, SEN).....	78
5.2.3.6.1	Slope (nominal) RDR (SLP).....	79
5.2.3.6.2	Slope Radial Direction (SRD)	79
5.2.3.6.3	Slope Heading RDR (SHD)	80
5.2.3.6.4	Slope Magnitude RDR (SMG)	81
5.2.3.6.5	Slope Northerly Tilt RDR (SNT)	81
5.2.3.6.6	Solar Energy RDR (SEP).....	81
5.2.3.7	Photometry RDR (IEP)	82
5.2.4	Instrument Placement RDRs	82
5.2.4.1	Common Concepts	82
5.2.4.1.1	Instrument Placement Modeling.....	82
5.2.4.1.2	Tether Effects.....	83
5.2.4.1.3	Instrument Clocking	83
5.2.4.1.4	Instrument Footpad Settling.....	84
5.2.4.1.5	Holes in the Data	84
5.2.4.1.6	Grapple Deflection	84
5.2.4.1.7	WTS Offset	84

5.2.4.2	Workspace Reachability RDR (WSH, WSS)	85
5.2.4.3	Instrument Tilt RDR (TFH, TFS, TFW).....	86
5.2.4.4	Delta Tilt RDR (TDS).....	88
5.2.4.5	Instrument Normal and Instrument Location RDRs (UIH, UIS, UIW, UPH, UPS, UPW, ZIH, ZIS, ZIW, XPH, XPS, XPW)	89
5.2.4.6	Instrument Roughness RDR (RUH, RUS, RUW, RUK).....	90
5.2.4.7	Instrument Placement Goodness RDR (GUH, GUS)	93
5.2.4.8	Arm Reachability RDRs (ARM)	95
5.3	Terrain Model (Mesh) Data Product Specification	95
5.3.1	Per-XYZ Terrain RDRs, Operations format	96
5.3.2	“Unified” Terrain RDRs, Operations format	97
5.3.3	Per-XYZ Terrain RDRs, PDS format.....	97
5.3.4	“Unified” Terrain RDRs, PDS format	99
5.4	Mosaic RDR Data Product Specification.....	101
5.4.1	Overview of Mosaics	101
5.4.2	Pointing Correction.....	102
5.4.3	Brightness Correction	103
5.4.4	Mosaicking Method for Most Mosaics	104
5.4.5	Mosaicking Method for Orthorectified Mosaics	106
5.4.6	Mosaic Ancillary Files	107
5.4.7	Cylindrical Projection Mosaic RDR.....	109
5.4.8	Camera Point Perspective Mosaic RDR.....	110
5.4.9	Cylindrical-Perspective Projection Mosaic RDR	111
5.4.10	Polar Projection Mosaic RDR.....	114
5.4.11	Vertical Projection Mosaic RDR	115
5.4.12	Orthographic Projection Mosaic RDR	117
5.4.13	Orthorectified Projection Mosaic RDR.....	118
5.4.14	Non-image Mosaics	119
5.4.15	Mosaic Index Files (IDX, ICM)	121
5.5	Browse Images.....	121
6.	Standards Used in Generating Products.....	123
6.1	PDS Standards	123
6.2	Time Standards	123
6.3	Coordinate Frame Standards.....	123
6.3.1.1	Lander (IDA) Frame	124
6.3.1.2	Lander Mechanical Frame	125
6.3.1.3	Site Frame	125
6.3.1.4	Local Level Frames	125
6.3.1.5	Arm Frame	125
6.3.1.6	“Rover” Motion Counter and Quaternion.....	125
7.	Applicable Software	127
7.1	Data Generation Programs	127
7.2	Data Viewing and Conversion Software.....	128
7.3	Applicable PDS Software Tools.....	129

7.4 Software Distribution and Update Procedures 129

Appendix A. Detailed Structure of the IMAGE_ID field in IDPH 130

Appendix B. Calibration Files 131

Appendix C. Label Keywords 134

LIST OF FIGURES

Figure 2-1: Location of IDC and ICC cameras and Instrument Deployment System (IDS) components..... 20

Figure 2-2: ICC image from the InSight testbed. The lander footpad can be seen in the lower right, and the SEIS instrument can be seen in the workspace. 22

Figure 2-3: Lander horizontal IDC mosaic 23

Figure 2-4: Lander vertical IDC mosaic..... 24

Figure 2-5: IDC image acquired in the InSight testbed (left). Corresponding 3-dimensional terrain shown as X (red), Y (green), and Z (blue) contours (right)..... 24

Figure 2-6: Acquisition and Readout of Camera Image Data..... 25

Figure 2-7: Bayer filter pattern..... 26

Figure 3-1: Image RDR Processing Flow (simplified) 32

Figure 4-1: VICAR Image File Structure..... 34

Figure 4-2: EDR/RDR File Naming Convention..... 38

Figure 4-3: Mosaic File Naming Convention..... 46

Figure 4-4: Mesh/DEM File Naming Convention..... 50

Figure 5-1: EJP showing orientation of camera (left). EDR after rotation (right)..... 56

Figure 5-2: ICC Image EDR (left) vs “Linearized” ICC Image RDR (right)..... 61

Figure 5-3: Left and Right Eye IDC Images Linearized to each other for stereo viewing..... 63

Figure 5-4: Color and Radiometric Processing Flow for InSight. The code contains the 3-letter product type and the Filter/Color code (indicating the color type)..... 70

Figure 5-5: XYZ Data Masked and Overlaid onto Image EDR. Red, green, blue indicate lines of constant X, Y, Z respectively at 0.1m spacing (blue lines are thus traditional elevation contours)..... 75

Figure 5-6: Range (Distance) data overlay on image EDR..... 76

Figure 5-7: Surface Normal Data Overlaid onto Image EDR. RGB colors map directly to XYZ components of the surface normal..... 78

Figure 5-8: Slope Data Overlaid onto Image. Blue is less slope, green into red is more slope..... 79

Figure 5-9: Slope Radial Direction (SRD). Data ranges from -7 degrees (blue) to +8 degrees (red) 80

Figure 5-10: Slope Heading Overlaid onto Image EDR. Heading is arbitrarily mapped to hue 81

Figure 5-11: Workspace Reachability for HP3 (WSH) Data Overlaid onto Image EDR. Green indicates nominal zone for HP3 deployment (5), yellow is ICC partially occluded (6), orange is the kinematically reachable zone (4), blue is HP3 footprint boundary (2), and red is outside the HP3 boundary (1). Numbers refer to left column of Table 5-2..... 86

Figure 5-12: Instrument Tilt for SEIS (TFS) Data Overlaid onto Image EDR. Green indicates good (state=5)..... 88

Figure 5-13: Instrument Delta Tilt for SEIS/WTS (TDS) Data Overlaid onto Image EDR. Green indicates good (state=5). 89

Figure 5-14: Instrument Roughness for SEIS (RUS) Data Overlaid onto Image EDR. Green indicates good (state=5)..... 93

Figure 5-15: Goodness for HP3 (GUH) Data Overlaid onto Image EDR. Green indicates good (state=5)..... 95

Figure 5-16: DeploymentWorkspace Rendered in Unified Terrain Mesh 100

Figure 5-17: Instrument Deployment Location Rendered in Unified Terrain Mesh..... 101

Figure 5-18: Cylindrical Projection Mosaic of the Lander Environment (from IDC)..... 109

Figure 5-19: Cylindrical Projection Mosaic of Lander Deck..... 110

Figure 5-20: Point-perspective mosaic..... 111

Figure 5-21: Cylindrical-Perspective Mosaic..... 114

Figure 5-22: Polar Projection Mosaic of Lander and Workspace..... 115

Figure 5-23: Vertical Projection Mosaic of Instrument Deployment Workspace. Arm is not masked off for purposes of illustration..... 116

Figure 5-24: Vertical Projection Mosaic of Lander Deck..... 117

Figure 5-25: Orthorectified Projection Mosaic of Workspace 119

Figure 5-26: XYZ Mosaic Overlaid onto Orthorectified Mosaic. Red, green, blue are lines of constant X, Y, Z respectively (so blue is a traditional contour map)..... 120

Figure 5-27: WSS Orthorectified Mosaic Overlaid onto Orthorectified Mosaic. Green, yellow, an magenta indicate deployable area; see Table 5-2 for meanings of colors. 121

Figure 6-1: InSight Coordinate Systems. Red captions show axes for Lander frame, blue for Site frame, and gray for Lander Mechanical. 123

Figure 6-2: Arm coordinate systems (side view) 126

Figure 6-3: Arm coordinate systems (top view)..... 126

LIST OF TABLES

Table 1-1: Product and Software Interfaces to this SIS..... 19

Table 2-1: ICC and IDC camera description..... 20

Table 3-1: APIDs assigned to cameras..... 28

Table 3-2: PDS and NASA Processing Levels..... 29

Table 4-1: Camera EDR and RDR Byte Ordering..... 37

Table 4-2: EDR/RDR Filename Fields..... 38

Table 4-3: Product ID Color Codes..... 43

Table 4-4: EDR/RDR Product Types..... 43

Table 4-5: Mosaic Filename Fields..... 46

Table 4-6: Mesh/DEM Filename Fields..... 50

Table 5-1: Camera EDR Data Products..... 55

Table 5-2: Workspace Reachability Values..... 85

Table 5-3: Band Assignments for Tilt Products..... 87

Table 5-4: Instrument Tilt State Assignments..... 87

Table 5-5: Band Assignments for Roughness Products..... 92

Table 5-6: Instrument Roughness State Assignments..... 92

Table 5-7: Placement Goodness Values..... 94

Table 6-1: Coordinate Frames Used for Surface Operations..... 124

Table 7-1: Key Software Tools for Camera Payload Downlink Processing..... 127

Table 7-2: Software Used to View Data..... 128

ACRONYMS AND ABBREVIATIONS

Note: The 3-letter product codes listed in Table 4-4 are not repeated in this table. Label keywords are also not listed in this table, although many of them contain abbreviations. See the Appendices for label definitions.

3D	Three Dimensional
APID	Application Process ID (telemetry packet application identifier)
APSS	Activity Planner SubSystem
ASCII	American Standard Code for Information Interchange
BSQ	Band SeQuential
CAD	Computer Aided Design
CAHV	Center, Axis, Horizontal, Vertical (camera model type)
CAHVOR	CAHV plus Optical, Radial
CAHVORE	CAHVOR plus Entrance (pupil)
CCD	Charge-Coupled Device
CIE	International Commission on Illumination
CODMAC	Committee on Data Management and Computation
CCSDS	Consultative Committee for Space Data Systems
DEM	Digital Elevation Map (or Model)
DN	Data Number
DOY	Day Of Year
DSN	Deep Space Network
EDR	Experiment Data Record
EM	Engineering Model
EOL	End Of file Labels
ERT	Earth Received Time
FEI	File Exchange Interface
FGICD	Flight-Ground Interface Control Document
FM	Flight model
FOV	Field Of View
FSW	Flight Software
GDS	Ground Data System
HiRISE	High Resolution Imaging Science Experiment (camera on MRO)
HP3	Heat Flow and Physical Properties Probe
HSI	Hue, Saturation, Intensity
HSV	Hue, Saturation, Value

ICC	Instrument Context Camera
ICD	Interface Control Document
ID	Identifier
IDA	InSight Deployment Arm
IDC	Instrument Deployment Camera
IDPH	Image Data Product Header
IDS	Instrument Deployment System
IEEE	Institute of Electrical and Electronics Engineers
IFOV	Instantaneous Field Of View
ILUT	Inverse Look-Up Table
InSight	Interior Exploration using Seismic Investigations, Geodesy, and Heat Transport
IOF	Irradiance Over Flux (often I/F; radiance factor measurement)
IOT	Instrument Operations Teams
iRGB	Instrument RGB (color space)
JPEG	Joint Photographic Experts Group (sometimes JPG)
JPL	Jet Propulsion Laboratory
LUT	Look-Up Table
LOD	Level Of Detail
MAHLI	Mars Hand Lens Imager (MSL camera)
MARDI	Mars Descent Imager (MSL camera)
Mastcam	Mast Camera (MSL camera)
MER	Mars Exploration Rovers
MIPL	Multimission Instrument Processing Laboratory
MIPLRAD	MIPL Radiometry
MMM	Mastcam, MAHLI, MARDI
MOS	Mission Operations System
MRO	Mars Reconnaissance Orbiter
MSB	Most Significant Byte
MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
NSYT	Abbreviation for the “InSight” project
OBJ	Name of mesh file format originally defined by Wavefront (not acronym)
ORT	Operational Readiness Test
PDS	Planetary Data System
PDS3	PDS Version 3 (older standard)
PDS4	PDS Version 4 (current standard)

PNG	Portable Network Graphics
RDR	Reduced Data Record
RGB	Red, Green, Blue
RMC	Rover Motion Counter
RSVP	Rover Sequencing and Visualization Program
SCLK	Spacecraft Clock
SEIS	Seismometer (instrument on InSight)
SFDU	Standard Format Data Unit
SGI	Silicon Graphics, Inc.
SIS	Software Interface Specification
SOL	Mars Solar day
SPeX	Science Product EXchange
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events kernels
sRGB	Standard RGB (color space)
TAU	Atmospheric opacity measure (Greek letter, not an acronym)
TBD	To Be Determined/Defined
TDS	Telemetry Delivery Subsystem
VICAR	Video Image Communication And Retrieval
WTS	SEIS Wind and Thermal Shield
XML	eXtensible Markup Language
Xpath	XML Path Language
xyY	Color space defined by CIE

INTRODUCTION

1.1 Purpose and Scope

The purpose of this Data Product Software Interface Specification (SIS) is to provide consumers of InSight instrument Experiment Data Record (EDR) and Reduced Data Record (RDR) data products with a detailed description of the products and how they are generated, including data sources and destinations.

Within the scope of this documents, the terms “EDR” and “PDS raw” refer to the same product and are used interchangeably.

Content in this document supports EDR and RDR data products generated by the MIPL for the following instruments:

- InSight Context Camera (ICC)
- InSight Deployment Camera (IDC)

The Project users for whom this SIS is intended include the Instrument Deployment System (IDS), the Rover Sequencing and Visualization Program (RSVP), Science Instrument teams that may need imaging products to use in conjunction with their experiments, and the InSight Ground Data System (GDS). This SIS is also intended for archival documentation of these products in the Planetary Data System (PDS).

In this document, the EDR data product is the raw, un-calibrated, uncorrected image data acquired by the cameras. It may include decompression if there was data product compression performed onboard the lander by the instrument. Within the group of camera instruments, the full frame image EDR data products are identical in format, except for some product label differences.

The RDR data products described in this document derived directly from one or more image EDR or image RDR data product(s). They are comprised of radiometrically calibrated and/or camera-model corrected and/or geometrically altered (including re-projected) versions of the raw camera data, as well as higher-order processed results (such as XYZ points and surface normals).

1.2 Contents

This Data Product SIS describes how the EDR data product is acquired by the camera and how it is processed, formatted, labeled, and uniquely identified, and how the RDR data product is derived from EDR or RDR data products. The document discusses standards used in generating the product and software that may be used to access the product. The EDR and RDR data product structure and organization is described in sufficient detail to enable a user to read the product. Finally, examples of composite EDR/RDR labels are provided, along with the definitions of the keywords in the label.

1.3 Constraints and Applicable Documents

This SIS is meant to be consistent with the contract negotiated between the InSight Project and the science team in which reduced data records and documentation are explicitly defined as deliverable products. Because this SIS governs the specification of camera-related products used during operations, any proposed changes to this SIS must be vetted by all affected software subsystems observing this SIS in support of operations (e.g., APSS, IDS and MIPL).

Keywords may be added to future revisions of this SIS. Therefore, it is recommended that software designed to process EDRs and RDRs specified by this SIS should be robust toward (new) unrecognized keywords.

The following documents are cited in this SIS, or contain relevant background information:

1. InSight Archive Generation, Validation and Transfer Plan, JPL D-75289, May 30 2014.
2. InSight Camera Flight to Ground Interface Control Document (FGICD), Lockheed Martin Space Systems, Rev-A, 12/17/2015, Lockheed Martin Space System, Denver Colorado, CAGE Code 04236
3. Maki, J. N., et al., “The Color Cameras on the InSight Lander”, Space Science Reviews, Pre-landing InSight Issue, 2018 (in preparation).
4. Abarca, H., et al., “Image and Data Processing for InSight Lander Operations and Science”, Space Science Reviews, Pre-landing InSight Issue, 2018 (in press).
5. Maki, J. N., et al., Mars exploration rover engineering cameras, J. Geophys. Res. Planets (1991–2012), 108, E12, 2003.
6. Maki, J., et al., The Mars science laboratory engineering cameras, Space science reviews, 170.1-4, 77-93, 2012.
7. Bell, J. F., et al., Mars exploration rover Athena panoramic camera (Pancam) investigation, J. Geophys. Res.: Planets (1991–2012), 108.E12, 2003.
8. Herkenhoff, K. E., et al., Athena microscopic imager investigation, J. Geophys. Res. Planets (1991–2012), 108, E12, 2003.
9. Interior Exploration Using Seismic Investigations, Geodesy, and Heat Transport (InSight) Participating Scientists Program Proposal Information Package, September 3, 2015
10. Alexander, D. and R. Deen, “Mars Science Laboratory Project Software Interface Specification (SIS); Camera & LIBS Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products, version 4.0”, JPL Doc. D-38107, data set MSL-M-NAVCAM-2-EDR-V1.0, NASA Planetary Data System, 2017.
11. Chen, A. et al, “Mars Exploration Rover Project Software Interface Specification (SIS); Camera Experiment Data Record (EDR) and Reduced Data Record (RDR) Operations and Science Data Products, version 4.4”, JPL Doc. D-22846, data set MER1-M-NAVCAM-2-EDR-OPS-V1.0, NASA Planetary Data System, 2014.
12. Alexander, D., R. Deen, and P. Zamani, “Phoenix Project Software Interface Specification (SIS); Camera Experiment Data Record (EDR) and Reduced Data Record (RDR) Data Products, version 1.1.2”, JPL D-33231, data set PHX-M-SSI-2-EDR-V1.0, NASA Planetary Data System, 2008.

13. Deen, R., "The VICAR File Format",
http://www-mipl.jpl.nasa.gov/external/VICAR_file_fmt.pdf .
14. Malvar, H.S., Li-Wei He, and R. Cutler, "High-quality linear interpolation for demosaicing of Bayer-patterned color images", Proceedings, IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), doi: 10.1109/ICASSP.2004.1326587, 2004.
15. Yakimovsky, Y and R. Cunningham, "A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras", January 7, 1977.
16. Gennery, D., "Camera Calibration, JPL IOM 347/86/10, February 5, 1986.
17. Gennery, D., et al., "Sensing and Perception Research for Space Telerobotics at JPL", Proceedings of the IEEE Intern. Conf. on Robotics and Automation, March 31 - April 3, 1987.
18. Gennery, D., "Camera Calibration Including Lens Distortion", JPL D-8580, May 31, 1991.
19. Bon, B., "Algorithm for Using CAHV to Determine SGI Graphics Viewpoint and Perspective", JPL IOM 3472-91-057, August 6, 1991.
20. Gennery, D., "Inclusion of Old Internal Camera Model in New Calibration", JPL IOM 386.3-94-001, February 22, 1994.
21. Gennery, D., "Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points", Calibration and Orientation of Cameras in Computer Vision, ISBN 3-540-65283-3, 2001.
22. Gennery, D., "Computations for Generalized Camera Model Including Entrance, Part 1 and Part 2", unpublished, May 23, 2001.
23. Gennery, D., "Generalized Camera Calibration Including Fish-Eye Lenses", JPL D- 03-0869, 2002.
24. Deen, R., "Issues with Linearization", JPL Docushare Collection 2700, File 75670, 2003.
25. Deen, R.G. and J.J. Lorre, Seeing in Three Dimensions: Correlation and Triangulation of Mars Exploration Rover Imagery, IEEE International Conf. on Systems, Man, and Cybernetics, Vol 1, pp 911-916, 2005. Doi: 10.1109/ICSMC.2005.1571262.
26. Marsviewer: <https://pds-imaging.jpl.nasa.gov/tools/marsviewer/>
27. Golombek, M., et al, "Geology and Physical Properties Investigations by the InSight Lander", Space Science Reviews, Pre-landing InSight issue, 2018 (in preparation).
28. Wright, J.R., A. Trebi-Ollenu, F. Hartman, B. Cooper, S. Maxwell, J. Yen, J. Morrison, "Terrain Modelling for In-situ Activity Planning and Rehearsal for the Mars Exploration Rovers", IEEE International Conf. on Systems, Man, and Cybernetics, Vol 2, pp 1372-1377, 2005. Doi: 10.1109/ICSMC.2005.1571338.
29. <http://paulbourke.net/dataformats/obj>
30. <http://www.meshlab.net>
31. CIE (1932). "Commission internationale de l'eclairage proceedings, 1931." Cambridge University Press Cambridge, 1932.
32. ITU (2015). International Telecommunication Union Radiocommunication Sector, Recommendation ITU-R BT.709-6, "Parameter values for the HDTV standards for production and international programme exchange", June 2015, BT Series, Broadcasting Service, Television, Electronic Publication Geneva, 2015.
<http://www.itu.int/rec/R-REC-BT.709-6-201506-I/en>
33. Wyszecki, G., Stiles, W.S. (1982). "Color Science: Concepts and Methods, Quantitative Data and Formulae", 2nd ed., John Wiley, New York, 1982.

This SIS is also consistent with the following Planetary Data System (PDS) documents, relating to PDS release 4 (PDS4). The PDS documents with version numbers are periodically updated. The specific release number used by a given data product is listed in the prologue of that data product label.

34. Planetary Data System Standards Reference, version 1.10.1, May 10, 2018
35. Data Provider's Handbook, Archiving Guide to the PDS4 Data Standards, version 1.10.1, May 10, 2018.
36. Planetary Science Data Dictionary Document, Abridged, version 1.10.1, May 7, 2018
37. Instrument Deployment Camera (IDC) / Instrument Context Camera (ICC) Science Team and PDS Imaging Node INTERFACE CONTROL DOCUMENT, JPL D-55781
38. InSight Context and Deployment Cameras (ICC/IDC) PDS Archive Bundle Software Interface Specification, JPL D-55604

1.3.1 Relationships with Other Interfaces

Changes to this EDR/RDR data product SIS document affect the following products, software, and/or documents. See Section 7 for more on the software.

Table 1-1: Product and Software Interfaces to this SIS

Name	Type P = product S = software D = document	Owner
MIPL database schema	P	MIPL (JPL)
Camera Payload EDRs <ul style="list-style-type: none"> • ICC • IDC 	P	MIPL (JPL)
Mars Program Suite <ul style="list-style-type: none"> • CRUMBS • MARSAUTOTIE • MARSBRT • MARSCAHV • MARSCOR3 • MARSDEBAYER • MARSDISPCOMPARE • MARSDISPINVERT • MARSINVERTER • MARSJPLSTEREO • MARSMAP • MARSMASK • MARSMCAULEY • MARSMOS • MARSNAV • MARSORTHO • MARSRAD • MARSRANGE • MARSRELABEL • MARSSLOPE • MARSTIE • MARSUNLINEARIZE • MARSUVW • MARSXYZ • NSYTFOOT • NSYTGOD • NSYTROUGH • NSYTTILT • NSYTWKSP 	S	MIPL (JPL)
RSVP	S	RSVP team (JPL)
IDS	S	IDS team JPL

2. INSTRUMENT OVERVIEW

The InSight imaging payload is comprised of two cameras, the Instrument Deployment Camera (IDC) and the Instrument Context Camera (ICC). The cameras share the identical electronics, but are equipped with different lenses and the detectors have been swapped out with Bayer-pattern color detectors (Maki et al, 2018) [3]. Because of the electronics commonality, image data from both cameras are functionally identical. The raw image data formats are also similar to the MSL Engineering Cameras (Maki et al, (2012), [6]) and MER Science and Engineering Cameras (Maki et al, (2003) [5], Bell et al, (2003) [7], Herkenhoff et al, (2003) [8]), with the main difference being the presence of a Bayer-pattern CCD and derived RGB 3-banded images on InSight.

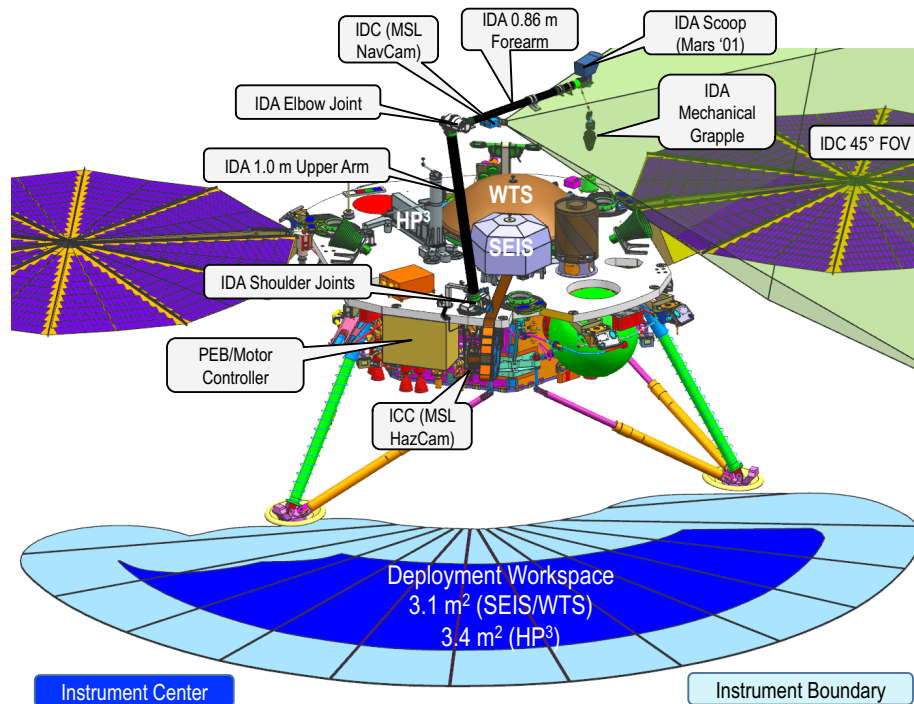


Figure 2-1: Location of IDC and ICC cameras and Instrument Deployment System (IDS) components

Table 2-1: ICC and IDC camera description

Parameter	ICC Value	IDC Value
Horizontal Field of View (degrees)	124	45
Vertical Field of View (degrees)	124	45
Diagonal Field of View (degrees)	180	67

Pixel Field of View at image center (mrad/pixel)	2.1	0.82
Focal Length (mm)	5.58	14.67
f/number	15	12
Readout Style	Frame transfer	
Imaging area (pixels)	1024 × 1024	
Pixel size (microns)	12 × 12	
Gain (electrons/DN)	50	
Full well capacity (electrons)	170,000	
Digitization (bits/pixel)	12	
System readout noise (electrons)	25	
Spectral wavelength/bandpass of Red Bayer filter	587 +/- 50 nm	
Spectral wavelength/bandpass of Green Bayer filter	550 +/- 50 nm	
Spectral wavelength/bandpass of Blue Bayer filter	528 +/- 85 nm	

2.1 Instrument Context Camera (ICC)

The ICC is a wide-angle color camera that is hard-mounted to the lander (see Figure 2-1). The primary purpose of the ICC is to provide contextual views of the entire deployment area during all phases of instrument deployment (see Figure 2-2). The ICC will acquire the first image of the workspace on Sol 0 and will continue to monitor the workspace throughout the pre-and post-deployment stages of the mission. The ICC will also provide a backup means of imaging the atmosphere for dust optical depth (τ) determinations to assist in power predictions. The ICC is a flight spare MSL Hazcam with an upgraded Bayer color filter array detector. Table 2-1 lists the main characteristics of the ICC. Note that stereo imaging is not possible with the ICC.

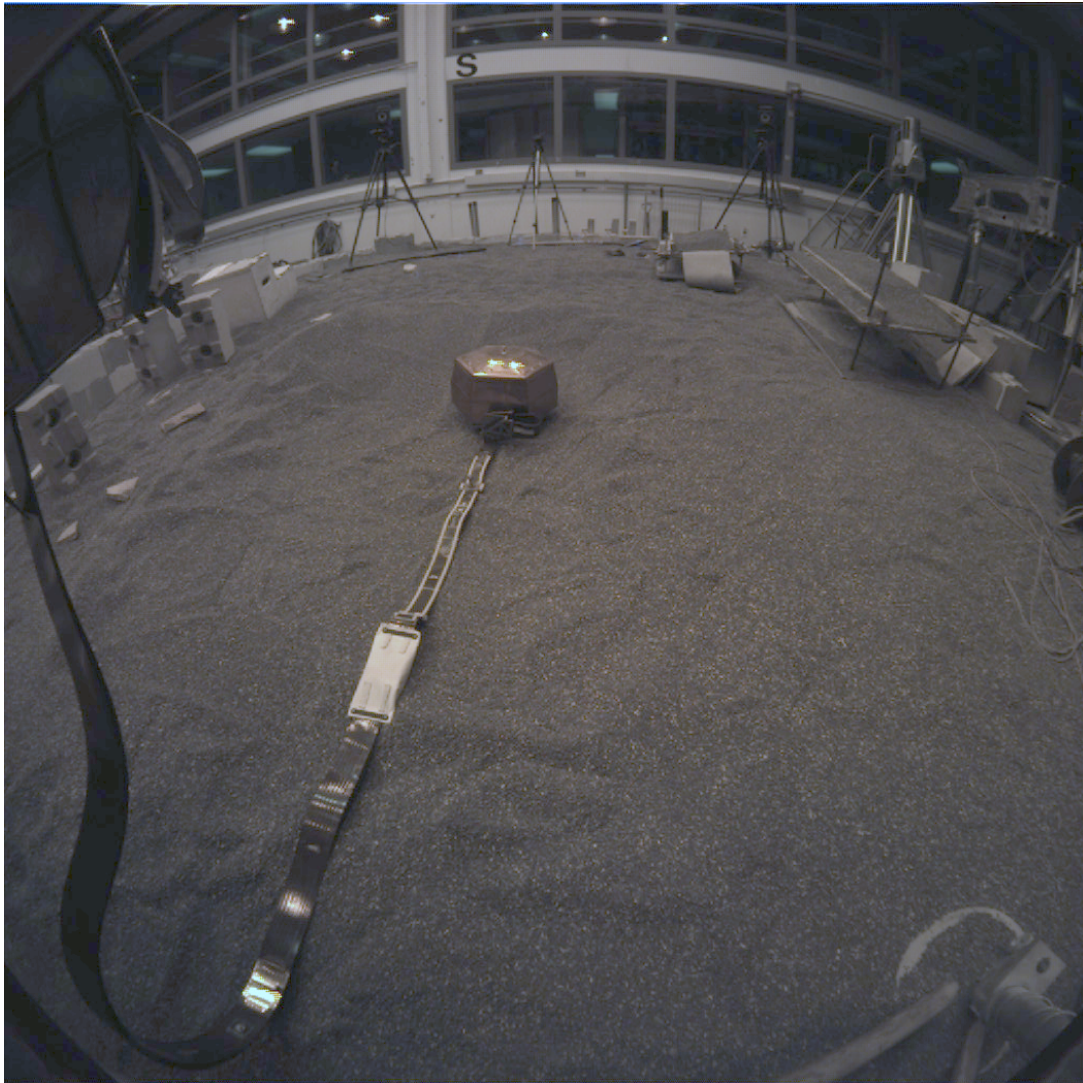


Figure 2-2: ICC image from the InSight testbed. The lander footpad can be seen in the lower right, and the SEIS instrument can be seen in the workspace.

2.2 Instrument Deployment Camera (IDC)

The IDC is a color camera that is mounted on the forearm segment of the Instrument Deployment Arm (see Figure 2-1). The main purposes of the IDC are to provide detailed stereo views of the deployment area and to verify arm interactions with the instrument hardware (e.g., grapple, lifting, and placement). The IDC will also acquire images of the sky; these images will be used to determine the dust optical depth (τ) in the Martian atmosphere as a primary input to solar panel power modeling. The IDC is a flight spare MSL Navcam. Like the ICC, the IDC has been upgraded with a Bayer color filter array detector. Table 2-1 lists the main characteristics of the IDC. Note that due to the placement of the camera, a portion of the arm is visible in the same location in every IDC image (seen in Figure 2-5).

After arm deployment, the IDC will image the lander feet, solar panels, payload deck, and deck fiducial targets (see Figure 2-3, Figure 2-4). The IDC will then be used to image the entire workspace in stereo at varying spatial resolutions; these images will be used to generate three-dimensional Digital Elevation Models (DEMs) of the deployment workspace (see Figure 2-5). Once the instrument placement sites are chosen, the IDC will image each site at a higher spatial resolution. During instrument deployment, IDC confirmation images will be acquired of the pre-grapple positions, grapple engagements, lifts, placements, and release of the instruments. A final IDC stereo pair will document the actual placement of the instruments. IDC stereo imaging of the SEIS instrument head assembly after placement and disengagement will also be used to determine its orientation (azimuth) on the surface. Although not required for instrument placement, the IDC will likely be used to acquire a 360-degree panorama of the landing site as well as other images of the surface and atmosphere.

The IDC is a single camera, not a traditional stereo pair. Stereo images are acquired by moving the arm between two exposures, thus seeing the same features from a different point of view. There is no stereo toe-in, so the overlapping field of view is somewhat limited. Typically only the shoulder joint is moved, in order to reduce stereo baseline error (due to having only one moving joint). Each frame is marked as a “Left” or “Right” frame in the acquisition command, in order to facilitate ground stereo processing. However, there is nothing fundamentally different between the “Left” and “Right” images. More information on stereo processing can be found in Abarca et al. (2018) [4].

More information on the IDC (Navcam) and ICC (Hazcam) hardware can be found in Maki et al (2018) [3], Maki et al. (2003) [5] and Maki et al. (2012) [6]. Additional information on the MER/MSL camera hardware can be found in Bell et al. (2003) [7] and Herkenhoff et al. (2003) [8].



Figure 2-3: Lander horizontal IDC mosaic

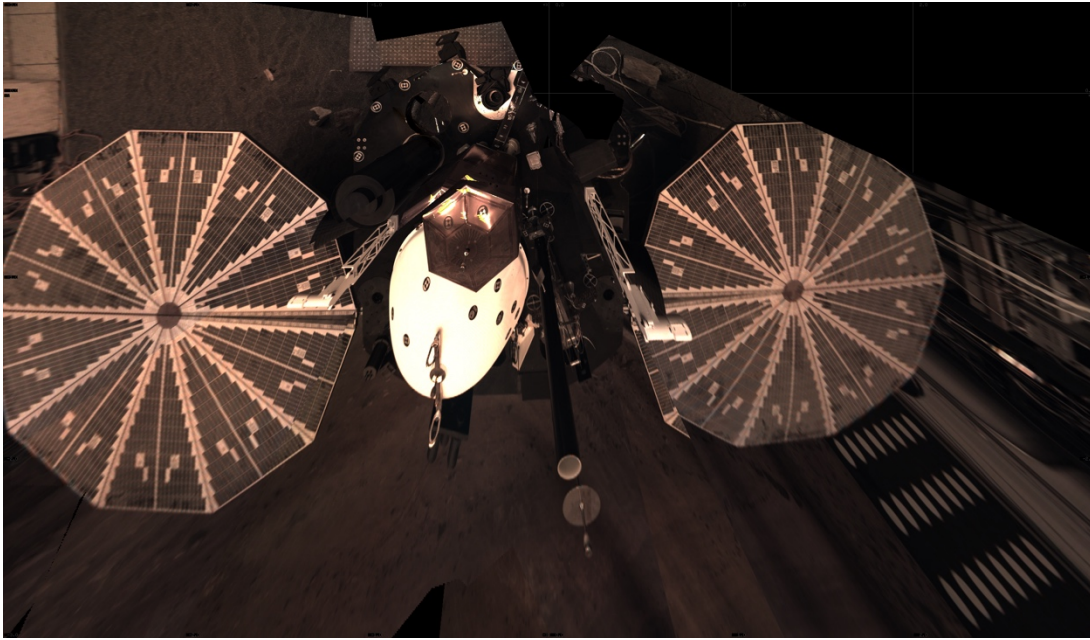


Figure 2-4: Lander vertical IDC mosaic

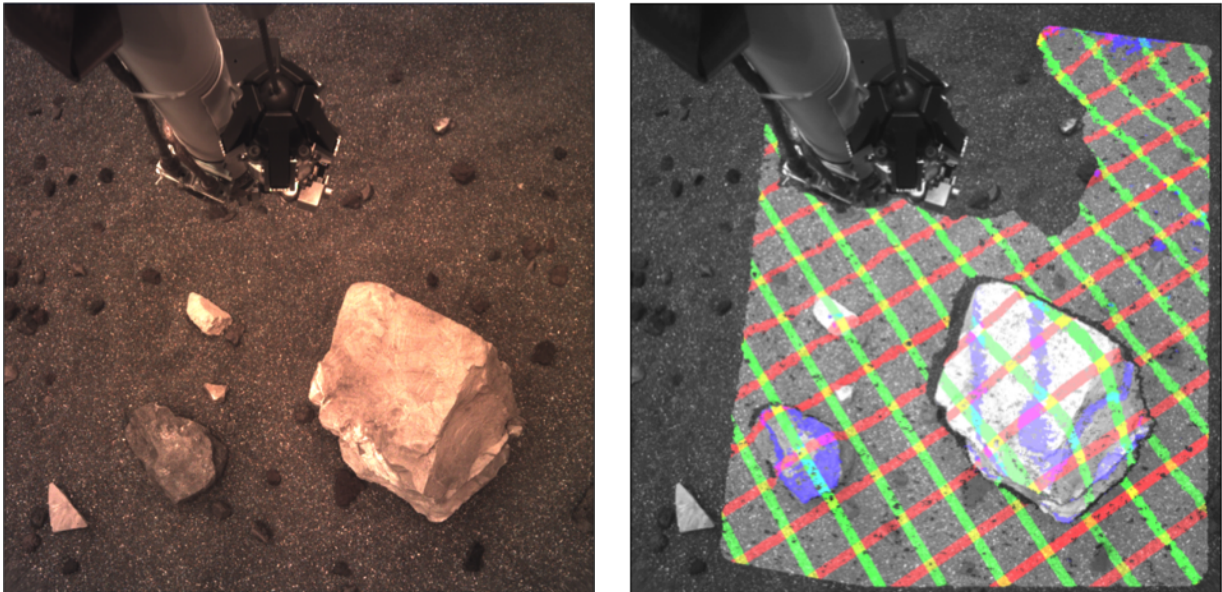


Figure 2-5: IDC image acquired in the InSight testbed (left). Corresponding 3-dimensional terrain shown as X (red), Y (green), and Z (blue) contours (right).

2.3 Camera CCD Layout

The camera reads out 1024x1056 pixels x 12 bits, as shown in Figure 2-6. This data is then transferred to the ground in one of two formats:

1. Raw data: 1024x1056 pixels x 12 bits, stored as 16 bit integers, Bayer pattern not demosaicked.
2. JPEG compressed: 1024x1024 pixels x 8 bits per color, reference pixels removed, Bayer pattern demosaicked.

In both cases, a 1024x1024 image is produced on the ground; for the raw case the reference pixels are stored in a separate ERP product (Section 5.1.3).

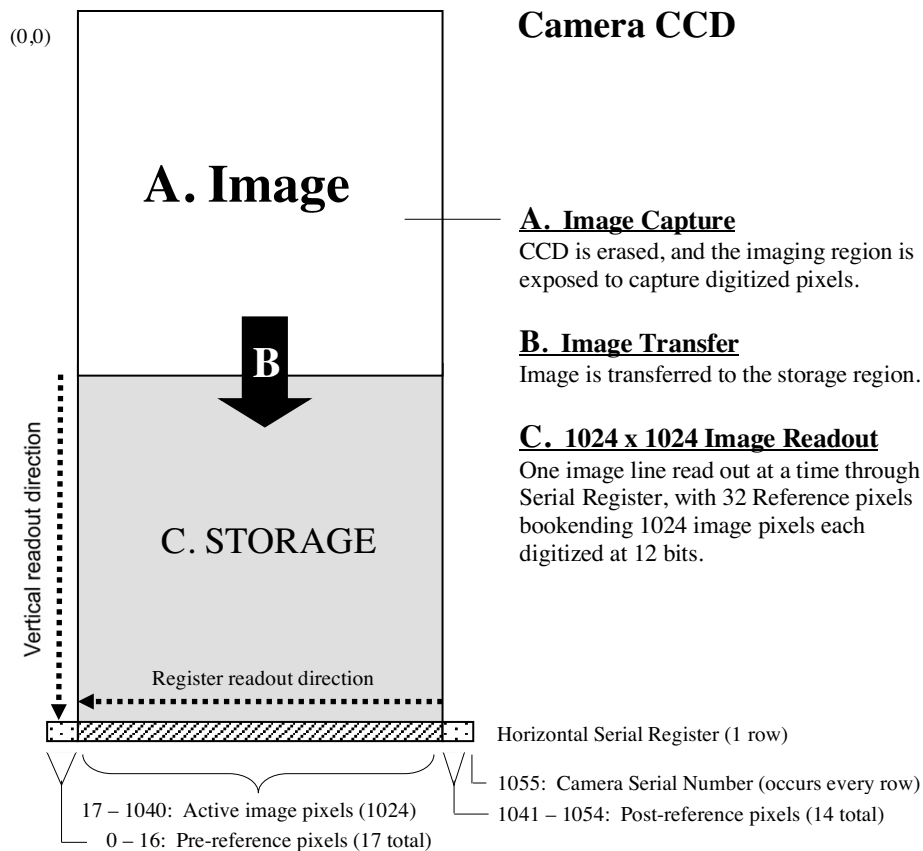


Figure 2-6: Acquisition and Readout of Camera Image Data

A Bayer filter/pattern is overlaid on top of the CCD (physical microfilters over each pixel), as shown in Figure 2-7, which produces color without the use of a filter wheel.

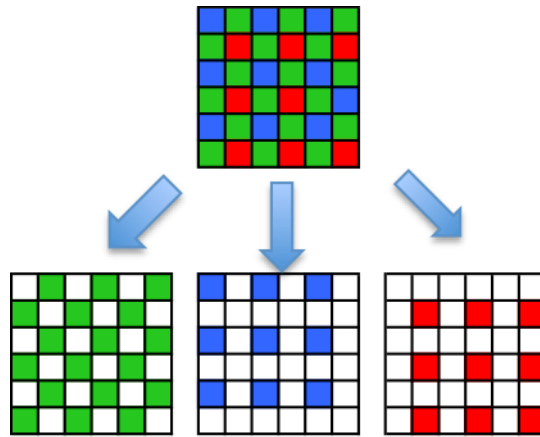


Figure 2-7: Bayer filter pattern

2.4 Camera Flight Software Processing

When sending data in JPEG mode, the flight software does five processing steps on the data: shutter subtraction, responsivity balancing, Bayer demosaicking, color correction, and companding.

2.4.1 Shutter Subtraction

A zero-second exposure is taken before the primary image. This is subtracted from the image to remove dark-current accumulation during image readout. It does not, however, account for dark current during the exposure itself (which is typically negligible at Mars temperatures).

2.4.2 Responsivity Balancing

Each pixel is multiplied by a factor, which varies depending on whether the pixel is Red, Green, or Blue (3x1 matrix). This serves to balance the responsivities of the three filters prior to demosaicking. The three factors used are tracked by the ground system and reported in the label. Because of the relatively low responsivity of the camera in the blue (Maki et al, 2018, [3]), this capability is used to boost the blue, in order to reduce problems with demosaicking noise prior to image compression.

2.4.3 Bayer demosaicking (de-Bayering)

The Bayer pattern is typically removed by the flight software using the Malvar algorithm (Malvar 2004, [14]) (in the case where a raw image is returned, the Malvar algorithm is applied on the ground). The Malvar algorithm is one of several demosaicking (or debayering) algorithms in widespread use. It is used on InSight because of its ease of implementation and generally good performance. Malvar was also used for the Mastcam, MAHLI, and MARDI instruments on MSL.

The Malvar algorithm produces three full size (1024x1024) images, one for each color band. The “missing” pixels are created using the assumption that chrominance (color) varies slowly, while luminance (intensity) varies quickly. The chrominance is determined by examining neighboring pixels; variations from this are assumed to be luminance changes. Thus the reconstructed image retains a full

1024x1024 spatial resolution. This is accomplished by applying convolution kernels to the raw Bayer-pattern image; see the Malvar paper for details.

Malvar can introduce artifacts into the image in areas where the chrominance varies rapidly. These generally manifest as a “zipper” pattern in the image. Experience with MSL shows that these artifacts do not generally impede operations or science on the images.

2.4.4 Color Correction Matrix

After demosaicking, the flight software applies a 3x3 color correction matrix. This original intent of this capability was to convert the iRGB image directly to sRGB prior to compression. However, it was decided pre-landing to not use this capability onboard due to the possibility of this step introducing noise into the red and green channels from the blue channel, which would preclude the recovery of the original noise levels in the individual R,G, and B values. Color correction will be performed on the ground instead. The onboard FSW matrix is reported in the label but is expected to be identity (1,0,0,0,1,0,0,0,1) throughout the mission unless the color correction approach changes.

2.4.5 Companding

Finally, the flight software “compands” the image, which converts it from 12 bits per pixel to 8 bits. This uses a square root encoding table, which preserves detail in the dark areas while allowing a high dynamic range in the bright areas. See Maki et al. (2018) [3] for more details on the companding function.

The flight software does not perform flat field correction or dark current removal (but see Section 2.4.1).

3. DATA PROCESSING OVERVIEW

The camera data, as with all telemetry data that is downlinked from the InSight lander, is acquired by the Deep Space Network (DSN) and staged on the Telemetry Delivery Subsystem (TDS) in SFDU packets. MIPL queries the TDS for specific packet identifiers (APID) and reconstructs the original image from the packets. The structure for all imaging APIDs is the same; only the downlink priorities differ. Imaging data may be in raw pixels, or JPEG-compressed. Imaging data may span multiple packets. Each packet contains identifiers so that packets for each image can be grouped together and processed. If a few packets are missed during transmission to Earth, it is sometimes possible to reconstruct a partial image if the camera image header is not missing. The detailed processing of missing packets and partial image generation are outside the scope of this document. In general, the MIPL telemetry processing attempts to make partial images when possible.

Table 3-1 lists the possible APIDs allocated to cameras. The content of the “unrecognized data” is outside the scope of MIPL requirements, and is not covered in this document. It is described in Lockheed-Martin’s Camera ICD.

Table 3-1: APIDs assigned to cameras

APID	APID Name	Purpose
126	ICC_1	ICC Image Data – priority 1
127	ICC_2	ICC Image Data – priority 2
128	ICC_3	ICC Image Data – priority 3
129	ICC_4	ICC Image Data – priority 4
130	ICC_5	ICC Image Data – priority 5
131	ICC_6	ICC Image Data – priority 6
132	ICC_UNRECOG	ICC Unrecognized Data (not processed by MIPL)
133	IDC_1	IDC Image Data – priority 1
134	IDC_2	IDC Image Data – priority 2
135	IDC_3	IDC Image Data – priority 3
136	IDC_4	IDC Image Data – priority 4
137	IDC_5	IDC Image Data – priority 5
138	IDC_6	IDC Image Data – priority 6
139	IDC_UNRECOG	IDC Unrecognized Data (not processed by MIPL)

3.1 Data Processing Level

The PDS4 terminology is described in the table below, and corresponding CODMAC and NASA product levels are also represented.

Due to historical reasons, and conventions used in past projects, the NASA product levels are commonly used within the InSight project. In this SIS, these terms are used interchangeably. It is expected that over time, the PDS4 convention will become the standard and will be used exclusively by both the science and engineering teams.

In addition, the terms “EDR” (Experiment Data Record) and “RDR” (Reduced Data Record) are used in a similar way. There is a large heritage of MIPL software and documentation in use by the InSight project, which references these terms, and products. Over time, their use may also be diminished in favor of the new convention.

Table 3-2: PDS and NASA Processing Levels

PDS4 processing level	PDS4 processing level description	NASA Level (used in PDS3)
Telemetry	Telemetry data with instrument data embedded. InSight does not archive telemetry data, with the exception of the extracted jpeg stream (contained in “EJP” products).	-
Raw	Original data from an instrument. If compression, reformatting, packetization, or other translation has been applied to facilitate data transmission or storage, those processes are reversed so that the archived data are in a PDS approved archive format. Often called EDRs (Experimental Data Records).	0
Partially Processed	Data that have been processed beyond the raw stage but which have not yet reached calibrated status. These and more highly processed products are often called RDRs (Reduced Data Records).	1A
Calibrated	Data converted to physical units, which makes values independent of the instrument. Often called RDRs.	1B
Derived	Results that have been distilled from one or more calibrated data products (for example, maps, XYZ point clouds, slopes). Supplementary data, such as calibration tables or tables of viewing geometry, used to interpret observational data should also be classified as ‘derived’ data if not easily matched to one of the other three categories. Often called RDRs.	2+

3.2 Product Label and Header Descriptions

InSight camera instrument EDRs and RDRs will be generated by JPL’s Multimission Image Processing Laboratory (MIPL). All imaging products will be comprised of two files: the image data file in VICAR format, and a detached PDS4 label file in XML format.

The image file is in VICAR format and it consists of two major parts: the VICAR label, which describes what the file is, and the image area, which contains the actual raster image pixels. So, each image has two labels: the attached VICAR label, which is embedded with the image data and is used during operations, and the detached PDS4 label, which is used for archiving the data at the PDS. Although the two labels' formats differ significantly, both labels contain the same semantic content and can be used interchangeably. During production, the PDS4 label is generated from the VICAR label to ensure consistency between them.

The VICAR label is used internally by the InSight project's GDS and also the MIPL software. The PDS4 label is added after each imaging product is generated so that it can be archived at the PDS.

See Section 4.1.1 for a complete description of the format.

3.3 EDR Data Product

As the fundamental image data product, the EDR will be generated as "raw" un-calibrated data within an automated pipeline process managed by the MIPL at JPL as part of the critical path in surface operations.

EDRs are versioned. When updated telemetry is received, a new version of the EDR will be generated. Updates are possible when a previously missing packet is either retransmitted from the lander, orbital assets, or the DSN.

3.4 Data Flow

The EDR processing begins with the MIPL's reconstruction of the original image from individual SFDU packets, which are delivered by the Deep Space Network. Each SFDU packet is in turn made from a CCSDS packet, which is the unit of data that is generated by the onboard flight system.

Packets may arrive at Earth out of order, more than once, or be missing. The MIPL telemetry processing pipeline will try to reconstruct the image from the available packets. It is not always possible to reconstruct an image, especially if key data from the camera header are missing. In such cases, the MIPL telemetry will hold the packets for a while in case additional packets from the same image are transmitted later. If possible, a newer and more complete version of an EDR will be generated.

Most of the SFDUs are processed by MIPL in near real time. There are built-in time gaps in the flow of the data, which make it impossible for all the packets to be processed in real time. Furthermore, partial images are only generated at the end of a downlink session. The details of packet processing are outside the scope of this document.

There may be multiple versions of any EDR due to retransmission of telemetry packets from the lander and/or upstream ground segments. Missing data will be identified within each image by fill (black pixel, i.e. 0) data. Note that the nature of the JPEG stream makes it impossible to reconstruct image data after a gap; the rest of the image will thus be fill data. In the event of retransmissions, new version of an

image will be stored with a new version number in the filename. The highest version always contains the most complete data available.

The EDR data products will be placed locally on the SPeX, and also published into FEI, as applicable, for distribution to the remote Instrument Operations Teams (IOT) as requested.

Note that since SPeX and FEI are used only during the life of the mission, the PDS archives are the official version of all EDRs.

3.5 RDR Data Product

RDR data products will be generated by MIPL using InSight-specific telemetry processing software and the multi-mission Mars Suite of VICAR image processing software at JPL (listed in Table 7-1). A high-level notional overview of the RDR interfaces in the VICAR processing flow are shown in Figure 3-1 (not all details are included).

The input for any given RDR will be one or more Camera EDRs or RDR data products and the output will be formatted according to this SIS. Additional metadata may also be added by each software to both VICAR and the PDS labels.

There may be multiple versions of InSight Camera RDRs, identified by the version number field in the filenames. All of the RDR products will be placed into SPeX and FEI for use and distribution during mission operations. The highest version is always the most complete RDR (version numbers do not in general match between EDR and RDR).

Note that since SPeX and FEI are used only during the life of the mission, the PDS archives are the official version of all RDRs.

For both EDRs and RDRs, the version number in PDS matches the version number of the identical product used in operations. Since generally only the highest version available at archive time is sent to PDS, this means not every version is in PDS, and there will be gaps in the version numbers in PDS – not all products will have a version 1, and available versions may not be sequential. This facilitates traceability and repeatability; if a scientist uses an operational product during the mission, the specific version used can either be unambiguously found in the archive, or it will be known that the specific version is unavailable in the archive (by definition, a higher-numbered version will be available in the archive).

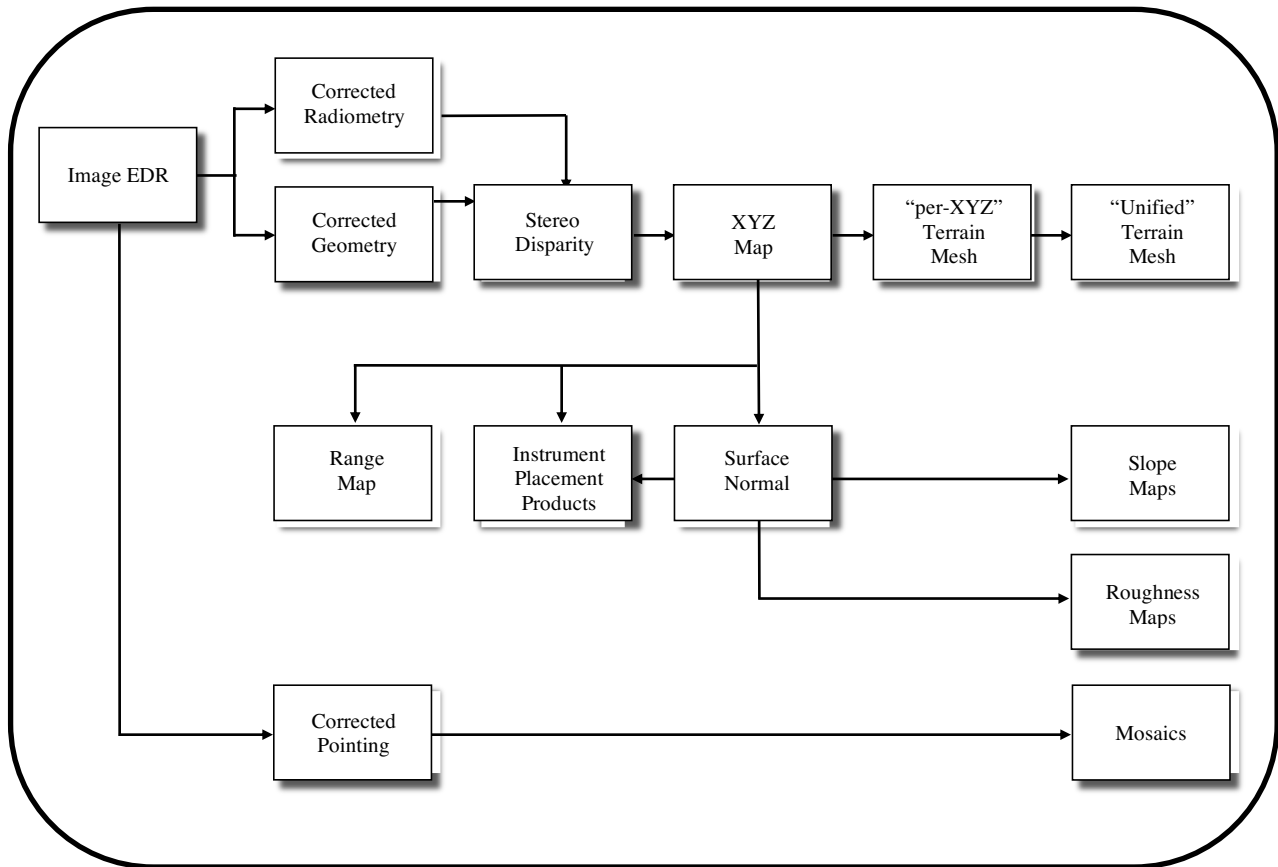


Figure 3-1: Image RDR Processing Flow (simplified)

3.6 Data Validation

Validation of the InSight EDRs will fall into two primary categories: automated and manual. Automated validation will be performed on every EDR/RDR product produced for the mission. Manual validation will only be performed on a subset of products. The Camera co-investigator is responsible for validation process, although he/she may assign that responsibility to MIPL.

Automated validation will be performed as a part of the MIPL product generation pipeline and at the time of archive generation. Validation operations performed will include such things as verification that the checksum in the label matches a calculated checksum for the data product (i.e., that the data product included in the archive is identical to that produced by the real-time process), a validation of the XML syntax of the label, a check of the label values against the PDS data dictionaries, schema, and semantic rules, and checks for internal consistency of the label items. As problems are discovered and/or new possibilities identified for automated verification, they will be added to the validation procedure.

Manual validation of the images will be performed both as spot-checking of data throughout the life of the mission, and comprehensive validation of a subset of the data (for example, a couple of days' worth of data).

These products will be viewed by a human being. Validation in this case will include inspection of the image or other data object for errors (like missing lines) not specified in the label parameters, verification that the target shown / apparent geometry matches that specified in the labels, verification that the product is viewable using the specified software tools, and a general check for any problems that might not have been anticipated in the automated validation procedure.

4. DATA PRODUCT OVERVIEW

4.1 Data Product Structure

The EDR and RDR image products described here consist of a VICAR file containing metadata and the image, and a separate (detached) PDS4 file containing metadata with the same semantic content in XML format.

There are two other formats used by a few RDR products: the original JPEG images received in telemetry are saved, and terrain meshes use a different format. Both are described in their respective sections in Section 5.

4.1.1 VICAR Image Format

The VICAR image file consists of an embedded ASCII VICAR label, followed a block of binary image data with the origin at the upper left pixel in line (row) 1, sample (column) 1.

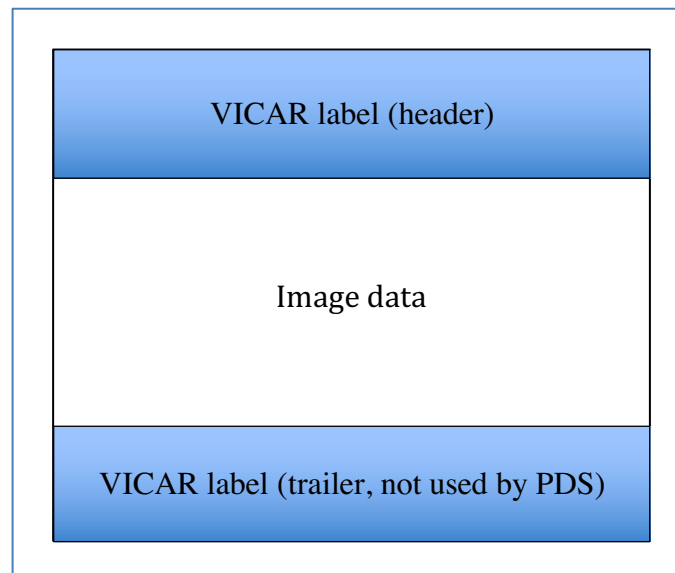


Figure 4-1: VICAR Image File Structure

The reader is referred to the VICAR File Format document [13] for complete details on the file format (see references in Section 1.3). However, the abbreviated description below should be sufficient to decode the file.

The VICAR label consists of a sequence of "keyword=value" pairs that describe the image, and is made up entirely of ASCII characters. Each keyword-value pair is separated by spaces. Keywords are strings, up to 32 characters in length, and consist of uppercase characters, underscores (" _"), and numbers (but

should start with a letter). Values may be integer, real, or strings, and may be multiple enclosed in parentheses (e.g. an array of 5 integers, but types cannot be mixed in a single value). Spaces may appear on either side of the equals character (=), but are not normally present. Single quotes enclose string values when needed, but not necessarily always (e.g. for values without spaces or special characters). A doubled single-quote inside a string translates to a single single-quote.

The first keyword is always `LBSIZE`, which specifies the size of the label area in bytes. `LBSIZE` is always a multiple of `RECSIZE`, even if the labels don't fill up the record. If the labels end before `LBSIZE` is reached (the normal case), then a 0 byte terminates the label string. If the labels are exactly `LBSIZE` bytes long, a null terminator is not necessarily present. The size of the label string is determined by the occurrence of the first 0 byte, or `LBSIZE` bytes, whichever is smaller. Note that VICAR allows for an EOL label section (End-Of-file Labels), which is a continuation area after the image, but this is not used for InSight PDS products. It is possible for EOL labels to occur on some operational products.

There are three sections of labels in the VICAR file. The first section is the system label, which describes the format and organization of the data. Most important are `NS`, `NL`, `NB` (number of samples, lines, and bands), and `FORMAT` (the data type). Following this, starting with `PROPERTY=`, is the “property” label section, which are groups of labels that roughly correspond to PDS3 groups or objects (there is a new `PROPERTY=` label for each group). Most of the image metadata is in these property labels. The final section, starting with `TASK=`, is the history label. This shows the name of each program that was run on the image during processing, and the parameters to that program. The parameter values may not mean much if you don't have access to the VICAR programs that processed the data, but they are included for completeness.

Following the labels is the image data itself. This data is an uncompressed raster of pixels. Multi-band images are in band-sequential (BSQ) format, meaning all of band 1, followed by all of band 2, etc. Although VICAR allows for binary headers and prefixes, PDS4 does not, so these are not used for InSight PDS or operational products.

Note that the VICAR image processing system is available open-source, and may be used to read the VICAR labels and images. The Mars-specific programs are not available as of this writing, but may be in the future.

4.1.2 PDS4 Label Format

The PDS4 label is a separate file with the same base name and an extension of “.xml”. It is in XML format whose content is controlled by the PDS Information Model and PDS, node, and mission data dictionaries. The PDS4 label contains the same semantic information as the VICAR label, although the format is quite different. For image files (ending in “.VIC” or “.IMG”), the label removes the VIC or IMG extension and replaces it with “.xml”. For other files, the “.xml” extension is added to the end of the complete filename, for example “.TXT.xml” or “.obj.xml”.

Full specifications for the PDS4 label are available in the PDS documentation. The InSight-specific contents are described in this document.

4.1.3 Mapping of PDS and VICAR Labels

The information contained in the PDS and VICAR labels are identical, by definition. For historic reasons, to address dependencies within the ground data system, and to facilitate potential future reprocessing, both labels are retained. MIPL uses the VICAR label exclusively for its processing; the PDS label is then generated from the VICAR label. This ensures consistency of label contents. In the event that additional VICAR labels are added during operations but after the PDS peer review, it is possible that those keywords may not appear in the PDS4 label.

It is important to note that these files are simultaneously valid PDS4 images, and valid VICAR images, and may be processed equally by tools of either system. It is critical for the integrity of the data that both labels be maintained, as described above.

The conversion from VICAR to PDS4 is accomplished using a Velocity template. Velocity is a transformation engine developed by the Apache Foundation that creates XML documents based on a template. Crucially, it has the ability to embed macros, which are used to extract VICAR label information.

4.1.4 Label References

Throughout this document, labels are referred to by their VICAR keyword names. This is for convenience; VICAR names are used in operations, and the names are more convenient to reference than PDS4 Xpath's. Appendix C provides the mapping between VICAR keywords and PDS4 Xpath descriptions, which specify where in the PDS4 XML label the value resides, as well as keyword definitions.

4.2 Binary Data Storage Conventions

EDR and RDR data products for the cameras are stored as binary data. For the image EDRs, the data formats include rescaled 8-bit integers stored in an unsigned byte, as well as 12-bit integers stored in signed 16-bit integers. RDRs may additionally be in 32-bit IEEE floating-point format, depending on the type. The VICAR labels are stored as ASCII text, and the PDS label is stored as a detached XML file.

4.2.1 Bit and Byte Ordering

The ordering of bits and bytes is only significant for pixel data; all other labeling information is in ASCII.

For non-byte data, which includes 16-bit signed shorts, 32-bit signed ints, and 32- and 64-bit IEEE floating-point numbers, the data may be stored in either Most Significant Byte first ("big-endian", as used by e.g. Sun computers and Java), or Least Significant Byte first ("little-endian", as used by e.g. Linux and Windows computers). In a single EDR/RDR product, the instrument data can have only one ordering, but it is dependent on the host platform where the data was processed. The endian-ness is

described in the PDS label, as well as the VICAR system label. This follows both the PDS and VICAR file format conventions.

Both file formats specify that bit 0 is the least significant bit of a byte. Although EDRs are constrained to use MSB only, RDRs are not; thus software should be flexible in order to handle either ordering.

Table 4-1: Camera EDR and RDR Byte Ordering

Address	MSB-first	LSB-first
n	most significant byte	least significant byte
n+1	next	next
n+2	next	next
n+3	least significant byte	most significant byte

4.3 File Naming

There are three file naming conventions used for InSight image data products. The first (Section 4.3.1) applies to the EDR data product and all Single-frame RDR data products. The second (Section 4.3.3) applies to all Mosaic data products. The third (Section 4.3.4) applies to Terrain Mesh products.

4.3.1 EDR and Single-Frame RDR Filename Convention

Single-frame images need only themselves and possibly their stereo partner to produce. This covers add EDRs and virtually all RDRs except for mosaics and unified (multi-image) meshes.

Each EDR can be uniquely identified by incorporating into the product filename the following fields: instrument identifier, mission source/epoch, starting spacecraft clock count (SCLK) of the camera event, data product type, and version. Each single-frame RDR requires additional fields to guarantee uniqueness: stereo ID, eye, linearization flag, filter, special processing flag, and creator. Additional fields (sol, mesh ID, mosaic ID, and sequence ID) are included to help with readability and to make pipeline processing easier.

The Single-frame RDR data products that share the naming scheme with the EDR data product are numerous. They are listed in Table 4-4 in Section 4.3.2.

Note that the “type” in column 2 of Table 4-2 is either “a” for ASCII, “i” for integer, or “n” for primarily integer (“number”) with some alpha exceptions.

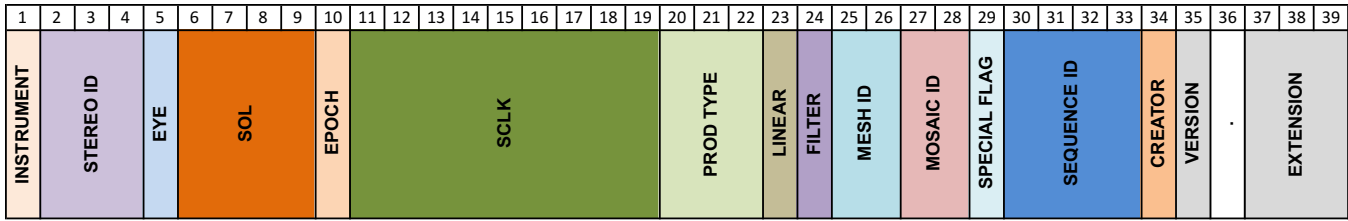


Figure 4-2: EDR/RDR File Naming Convention

Table 4-2: EDR/RDR Filename Fields

Field	Position (size, type)	Description
Instrument	01 (1,a)	Instrument Identifier: <ul style="list-style-type: none"> • C : InSight Context Camera (ICC) • D : InSight Deployment Camera (IDC)
Stereo ID	02 (3,n)	<p>Identifies which stereo pair this image belongs to, which allows stereo pairs to be easily matched.</p> <ul style="list-style-type: none"> • 000 : Mono image • 001-127 : Image is part of a stereo pair; stereo partner has the same ID • Any other characters (0-9, A-Z) : Image is part of a ground-matched stereo pair; stereo partner has the same ID <p>EDRs may only have numeric values, in the range 000-127. These values represent commanded stereo pairs and come from the IMAGE_ID field in the IDPH (Appendix A). Note that stereo ID values are only unique within a single (sol, sequence ID, mesh ID) tuple.</p> <p>RDRs may have the same numeric values. Additionally, they may have any combination of letters (A-Z) and numbers (0-9). If there are any letters present, the image is part of a stereo pair that was matched on the ground. It is quite possible for one original EDR to be matched with multiple partners as part of ground processing. This is accomplished by copying an RDR (typically ILT type) and renaming it to contain a different stereo sequence ID. These alternative matches act like independent images in the pipeline, but the SCLK can be used to trace them back to their original source image.</p> <p>Regardless of alpha vs. numeric, <i>all</i> of stereo ID, sol, sequence ID, and mesh ID have to match for the images to be considered a stereo pair (exception: stereo id of 000 means it is not a stereo image). Mosaic IDs need not match.</p>

Field	Position (size, type)	Description
Eye	05 (1,a)	<p>Identifies if the image should be used as left or right eye for stereo processing.</p> <ul style="list-style-type: none"> • A : Anaglyph • C: Colorglyph • X : Mixed, used for by products which have more than one image. Since InSight does not have a true stereo camera, it is sometimes necessary to mix single, mono images together to create a composite product. • L : Left eye • M : Mono • R : Right eye • S : Stereo 2-band <p>Note that “eye” is an artificial concept since there is only one IDC camera; the eye is assigned when the image is acquired to assist in stereo pipeline processing. Although this filename convention is for single-frame products, some (such as stereo anaglyphs) combine the image and its stereo partner into one product using this convention.</p>
Sol	06 (4,n)	<p>Solar day number, with landing day being zero. Zero-filled, decimal number.</p> <p>Testbed and Cruise data may instead treat this as Day of Year in the form YDDD where “Y” is the least significant digit of the year. Zero-filled decimal number.</p> <p>Note about “Y”: If the mission extends beyond year 2019, then “A” will be used for 2020, “B” for 2021, and so on. The “Y” field will never be less than “5” (2015). All YDDD data generated after landing must have an epoch of “T”, to distinguish them from surface data past sol 5000.</p>
Epoch	10 (1,a)	<p>Mission epoch identifier:</p> <ul style="list-style-type: none"> • _ : Surface operations, or Operational Readiness Tests. • T : Testbed or ATLO data • C : Flight model, cruise
SCLK	11 (9,i)	Space Craft Clock corresponding to the time the image was acquired
Product type	20 (3,a)	Product Identifier. See Table 4-4 for details.
Linear	23 (1,a)	<p>Linearization flag:</p> <ul style="list-style-type: none"> • L : Product has been linearized • _ : Non-linearized (raw geometry) <p>Note that linearization requires a stereo partner; this is indicated by the Stereo ID field. This corresponds to “Actual” linearization mode in MSL parlance (“Nominal” linearization is not possible for a single camera). If an image has multiple stereo partners, the linearized images will be different for each partner. For this reason, InSight uses non-linearized geometry for almost all products; linearized images are only intermediate products.</p> <p>All EDRs are Raw geometry (“_”).</p>

Field	Position (size, type)	Description																									
Filter / Color	24 (1,a)	<p>Color flag (see also Section 5.2.2.2.2):</p> <ul style="list-style-type: none"> • <code>_</code> : Raw Bayer pattern (or grayscale image if color is removed) <p>Otherwise, the flag is set based on the color space depending on whether the product is a 3-band product, or an individual band.</p> <table border="1"> <thead> <tr> <th>Color Type</th> <th>3-Band</th> <th>Band 1</th> <th>Band 2</th> <th>Band 3</th> </tr> </thead> <tbody> <tr> <td>RGB</td> <td>F</td> <td>R</td> <td>G</td> <td>B</td> </tr> <tr> <td>XYZ</td> <td>T</td> <td>X</td> <td>Y</td> <td>Z</td> </tr> <tr> <td>xyY</td> <td>C</td> <td>J</td> <td>K</td> <td>L</td> </tr> <tr> <td>HSI</td> <td>P</td> <td>H</td> <td>S</td> <td>I</td> </tr> </tbody> </table> <p>Other color flags may be defined in the future.</p> <p>Only “F” or “_” can appear in EDRs.</p>	Color Type	3-Band	Band 1	Band 2	Band 3	RGB	F	R	G	B	XYZ	T	X	Y	Z	xyY	C	J	K	L	HSI	P	H	S	I
Color Type	3-Band	Band 1	Band 2	Band 3																							
RGB	F	R	G	B																							
XYZ	T	X	Y	Z																							
xyY	C	J	K	L																							
HSI	P	H	S	I																							
Mesh ID	25 (2,n)	<p>Identifies which mesh this image is part of. Meshes are created by gathering images sharing the same mesh ID (see also Appendix A). EDRs may have the following values:</p> <ul style="list-style-type: none"> • <code>00</code> : Image is not part of any mesh/DEM • <code>01-09</code> : Reusable mesh ID that matches only within the sol • <code>10-63</code> : Mesh ID that matches across the entire mission <p>RDRs may have the above numeric values, or they may have alphanumeric characters, indicating a ground-assigned ID. Ground-assigned IDs should have at least one alpha (non-numeric) character.</p> <p>Images must share the same Eye, Epoch, Sequence ID, and Mesh ID to be combined in an automatically-generated mesh. The sol must also match if the mesh ID is 01-09.</p> <p>The Mesh ID is part of the <code>IMAGE_ID</code> field in the IDPH.</p>																									

Field	Position (size, type)	Description
Mosaic ID	27 (2, n)	<p>Identifies which mosaic this image is part of. Mosaics are created by gathering images sharing the same mosaic ID (see also Appendix A). EDRs may have the following values:</p> <ul style="list-style-type: none"> • 00 : Image is not part of any mosaic • 01-09 : Reusable mosaic ID that matches only within the sol • 10-63 : Mosaic ID that matches across the entire mission <p>RDRs may have the above numeric values, or they may have alphanumeric characters, indicating a ground-assigned ID. Ground-assigned IDs should have at least one alpha (non-numeric) character.</p> <p>Images must share the same Eye, Epoch, Sequence ID, and Mosaic ID to be combined in an automatically-generated mosaic. The sol must also match if the mosaic ID is 01-09.</p> <p>The Mosaic ID is part of the IMAGE_ID field in the IDPH.</p>
Special Flag	29 (1,a)	<p>Special Processing flag, applicable to RDRs only. EDRs always have "_".</p> <p>The special processing character is used to indicate off-nominal or special processing of the image. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than "_" which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges of sols.</p> <p>An ASCII text file will be maintained containing all special processing designators that are used, the sols they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.</p>
Sequence ID	30 (4, n)	<p>Context specific field for use by camera uplink team, generally used to indicate the command sequence the image was acquired from. Specific values will be assigned by the uplink team.</p> <p>EDRs must have numeric values in the range 0000-4095.</p> <p>RDRs may have the above numeric values, or they may have alphanumeric characters, indicating a ground-assigned ID. Ground-assigned IDs should have at least one alpha (non-numeric) character.</p> <p>The sequence ID must match for stereo pairs, and auto-generated meshes and mosaics. Ground-assigned IDs thus allow different pairings of data than anticipated when the images were acquired.</p>

Field	Position (size, type)	Description
Creator	34 (1,a)	ID for the team that created this product: <ul style="list-style-type: none"> • M : MIPL • _ : Undefined / Other Note: Other creator codes will likely be added in the future.
Version	35 (1,a)	Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists. The value can contain both digits and letters. <ul style="list-style-type: none"> • 1 – 9, then A-Z • “_” : indicates overflow in case all version numbers are used.
.	36 (1,a)	Always set to “.”
Extension	37 (3,a)	File extension: <ul style="list-style-type: none"> • VIC : Used for VICAR files • IMG : Same as .VIC; occasionally used for compatibility with some old software. • JPG : Used for JPEG compressed (non VICAR) image files. • PNG : Used for PNG-formatted image files (typically browse images). • TXT : used for ASCII files, such as TAU reports. • xml : PDS4 Label file (labels for files other than VIC or IMG have dual extensions, e.g. “.TXT.xml”) • iv : Inventor-format file, used for meshes • ht : Height-map file, used for meshes • rgb : Skin file in SGI RGB format, used for meshes • obj: Mesh file in Wavefront OBJ format • mtl: Materials file in Wavefront OBJ format

4.3.2 Product Identifiers

There are a large number of possible EDR and RDR Product Type values, which indicate what type of data the image contains. These are listed in Table 4-4. They are color coded as shown in Table 4-3 (and in Marsviewer) in order to help reduce confusion. Note that these indicate possible products, not actual products; many product types may never actually be generated and most are not generated systematically. The PDS Processing Level and Derived Image Type columns indicate values for the corresponding labels. The Sort Group is used by Marsviewer to put the RDRs in logical order according to the processing flow (more derived products generally have higher numbers). Section 5 discusses what the various product types are.

Note that the “TAU” type in the table is not an image and is thus not eligible for mosaics etc. Tau products are delivered separately to PDS and are not covered in this document. All the other types are images and may be made into mosaics (although not all types are typically made into mosaics).

Table 4-3: Product ID Color Codes

Primary to End-User: most important, most used, most popular products	
Secondary: intermediate products, or final product not commonly used	
Special: products generated outside of the pipeline as a special request	

Table 4-4: EDR/RDR Product Types

Data Product	Type	PDS Proc. Level	Derived Image Type	Sort Grp
First order product	EDR	Raw	IMAGE	0
Original JPEG as received from the lander.	EJP	Raw	N/A	5
Reference Pixel	ERP	Raw	N/A	5
CAHV-linearized (identical to “EDR” except Geometry type)	LIN	Derived	IMAGE	5
Bayer pattern (identical to “EDR” except Bayer pattern extract)	BAY	PP	IMAGE	7
Inverse lookup table (ILUT) with corrections such as de-spike	ILC	PP	IMAGE	10
Inverse lookup table (ILUT), Masked	ILM	PP	IMAGE	10
Inverse lookup table (ILUT) with pointing correction applied	ILP	PP	IMAGE	10
Inverse lookup table (ILUT)	ILT	PP	IMAGE	10
Image Mask File	MSK	Derived	MASK	15
Rad-corrected IOF radiance factor, float	IOF	Calibrated	IMAGE	20
Rad-corrected IOF radiance factor, integer	IOI	Calibrated	IMAGE	20
Rad-corrected absolute radiance units, float	RAF	Calibrated	IMAGE	20
Rad-corrected absolute radiance units, 15-bit integer static scale	RAD	Calibrated	IMAGE	20
Rad-corrected absolute radiance units, 12-bit integer static scale	RAS	Calibrated	IMAGE	20
Rad-corrected absolute radiance units, 15-bit int dynamic scale	RAY	Calibrated	IMAGE	20
Rad-corrected radiance (RAD), masked	RDM	Calibrated	IMAGE	20
Rad-corrected for Instrument Effects only, integer DN	RIE	Calibrated	IMAGE	20
Rad-corrected for Instrument Effects only, float	RIF	Calibrated	IMAGE	20
Rad-corrected scaled radiance (RAS), masked	RSM	Calibrated	IMAGE	20
Zenith-scaled radiance, float	RZF	Calibrated	IMAGE	20
Zenith-scaled radiance, 15-bit integer static scale	RZD	Calibrated	IMAGE	20
Zenith-scaled radiance, 12-bit integer static scale	RZS	Calibrated	IMAGE	20
Zenith-scaled radiance, 15-bit integer dynamic scale	RZY	Calibrated	IMAGE	20
Stereo Delta Disparity (2-band, true disparity offset)	DDD	Derived	DISP_DELTA_MAP	25
Stereo Delta Disparity Line (single-band)	DDL	Derived	DISP_DELTA_LINE_MAP	25
Stereo Delta Disparity Sample (single-band)	DDS	Derived	DISP_DELTA_SAMPLE_MAP	25
Stereo Disparity Error Metric	DSE	Derived	DISPARITY_ERROR_MAP	25
Stereo Disparity of Lines (single-band)	DSL	Derived	DISPARITY_LINE_MAP	25
Stereo Disparity Final	DSP	Derived	DISPARITY_MAP	25
Stereo Disparity Raw	DSR	Derived	DISPARITY_MAP	25
Stereo Disparity of Samples (single-band)	DSS	Derived	DISPARITY_SAMPLE_MAP	25
Stereo Disparity Mask File	MDS	Derived	MASK	25
Surface Roughness (general, not instrument-specific)	RUF	Derived	ROUGHNESS_MAP	25
Stereo First-stage Disparity Final	DFP	Derived	DISPARITY_MAP	30
Stereo First-stage Disparity Line	DFL	Derived	DISPARITY_LINE_MAP	30
Stereo First-stage Disparity Sample	DFS	Derived	DISPARITY_SAMPLE_MAP	30

Data Product	Type	PDS Proc. Level	Derived Image Type	Sort Grp
XYZ Mask File	MXY	Calibrated	MASK	35
XYZ Error Metric	XYE	Calibrated	XYZ_ERROR_MAP	35
XYZ expressed in Rover Nav frame (Lander frame for InSight)	XYR	Calibrated	XYZ_MAP	35
XYZ expressed in Site frame	XYZ	Calibrated	XYZ_MAP	35
XYZ expressed in Rover (Lander/IDA frame for InSight), Masked	XRM	Calibrated	XYZ_MAP	35
XYZ expressed in Site frame, Masked	XYM	Calibrated	XYZ_MAP	35
XYZ Filled	XYF	Derived	XYZ_MAP	40
XYZ with Overlay	XYO	Derived	IMAGE	40
Digital Elevation Model. Like ZZZ but positive up (Site frame for InSight)	DEM	Calibrated	ELEVATION_MAP	45
XYZ X-band	XXX	Calibrated	X_MAP	45
XYZ Y-band	YYY	Calibrated	Y_MAP	45
XYZ Z-band (Lander/IDA frame for InSight)	ZZZ	Calibrated	Z_MAP	45
XYZ X-band Filled	XXF	Derived	X_MAP	50
XYZ Y-band Filled	YYF	Derived	Y_MAP	50
XYZ Z-band Filled	ZZF	Derived	Z_MAP	50
Surface Normal (UVW) Filled	UVF	Derived	UVW_MAP	55
Surface Normal (UVW) with Overlay	UVO	Derived	IMAGE	55
Surface Normal (UVW) Projected onto Plane	UVP	Derived	UVW_MAP	55
Surface Normal (UVW) for Slope computations	UVS	Calibrated	UVW_MAP	55
Surface Normal (UVW) Angle ('T' for theta) between Normal and Plane	UVT	Derived	ANGLE_MAP	55
Surface Normal (UVW)	UVW	Calibrated	UVW_MAP	55
Surface Normal (UVW) U-band	UUU	Calibrated	U_MAP	60
Surface Normal (UVW) V-band	VVV	Calibrated	V_MAP	60
Surface Normal (UVW) W-band	WWW	Calibrated	W_MAP	60
Surface Normal (UVW) U-band Filled	UUF	Derived	U_MAP	65
Surface Normal (UVW) V-band Filled	VVF	Derived	V_MAP	65
Surface Normal (UVW) W-band Filled	WWF	Derived	W_MAP	65
Range Error Metric	RNE	Calibrated	RANGE_ERROR_MAP	70
Range Filled	RNF	Derived	RANGE_MAP	70
Range from Camera	RNG	Calibrated	RANGE_MAP	70
Range from Camera, Masked	RNM	Calibrated	RANGE_MAP	70
Range with Overlay	RNO	Derived	IMAGE	70
Range from Rover Nav frame origin (Lander for InSight)	RNR	Calibrated	RANGE_MAP	70
Arm Reachability, Masked	ARK	Derived	REACHABILITY_MAP	75
Arm Reachability	ARM	Derived	REACHABILITY_MAP	75
Arm Reachability with overlay	ARO	Derived	IMAGE	75
Arm Reachability Mask File	MAR	Derived	MASK	75
Solar Energy	SEN	Derived	SOLAR_ENERGY_MAP	80
Slope Heading	SHD	Derived	SLOPE_HEADING_MAP	80
Slope Heading with Overlay	SHO	Derived	IMAGE	80
Slope with Overlay	SLO	Derived	IMAGE	80
Slope	SLP	Derived	SLOPE_MAP	80
Slope Magnitude	SMG	Derived	SLOPE_MAGNITUDE_MAP	80
Slope Magnitude with Overlay	SMO	Derived	IMAGE	80
Slope Northerly Tilt with Overlay	SNO	Derived	IMAGE	80
Slope Northerly Tilt	SNT	Derived	NORTHERLY_TILT_MAP	80
Slope Radial Direction	SRD	Derived	RADIAL_SLOPE_MAP	80
Incidence, Emission, Phase angles Filled	IEF	Derived	IEP_MAP	85
Incidence, Emission, Phase angles	IEP	Derived	IEP_MAP	85
Overall goodness map for HP3	GUH	Derived	GOODNESS_MAP	90
Overall goodness map for SEIS/WTS	GUS	Derived	GOODNESS_MAP	90
Surface Roughness map for HP3	RUH	Derived	INST_ROUGHNESS_MAP	100

Data Product	Type				PDS Proc. Level	Derived Image Type	Sort Grp
Surface Roughness map for SEIS	RUS				Derived	INST_ROUGHNESS_MAP	100
Surface Roughness map for WTS	RUW				Derived	INST_ROUGHNESS_MAP	100
Surface Roughness map for WTS Skirt	RUK				Derived	INST_ROUGHNESS_MAP	101
Instrument Delta Tilt map for SEIS/WTS	TDS				Derived	INST_DELTA_TILT_MAP	105
Instrument Tilt map for HP3	TFH				Derived	INST_TILT_MAP	105
Instrument Tilt map for SEIS	TFS				Derived	INST_TILT_MAP	105
Instrument Tilt map for WTS	TFW				Derived	INST_TILT_MAP	105
Instrument Workspace boundary for HP3	WSH				Derived	INST_WORKSPACE_MAP	110
Instrument Workspace boundary for SEIS and WTS	WSS				Derived	INST_WORKSPACE_MAP	110
Instrument Normal in Site frame for HP3	UIH				Derived	INS_UVW_MAP	111
Instrument Normal in Site frame for SEIS	UIS				Derived	INS_UVW_MAP	111
Instrument Normal in Site frame for WTS	UIW				Derived	INS_UVW_MAP	111
Instrument Tilt map in Lander frame for HP3	UPH				Derived	INST_TILT_MAP	112
Instrument Tilt map in Lander frame for SEIS	UPS				Derived	INST_TILT_MAP	112
Instrument Tilt map in Lander frame for WTS	UPW				Derived	INST_TILT_MAP	112
Instrument Z Coordinate in Site frame for HP3	ZIH				Derived	INST_Z_MAP	113
Instrument Z Coordinate in Site frame for SEIS	ZIS				Derived	INST_Z_MAP	113
Instrument Z Coordinate in Site frame for WTS	ZIW				Derived	INST_Z_MAP	113
Instrument XYZ Coordinate in Lander frame for HP3	XPH				Derived	INST_XYZ_MAP	114
Instrument XYZ Coordinate in Lander frame for SEIS	XPS				Derived	INST_XYZ_MAP	114
Instrument XYZ Coordinate in Lander frame for WTS	XPW				Derived	INST_XYZ_MAP	114
Tau report	TAU				Derived	N/A	115
Terrain Class Confidence	TEN				Derived	TERRAIN_CLASSIFICATION_MAP	120
Terrain Class	TER				Derived	TERRAIN_CLASSIFICATION_MAP	120
Mosaic Input Index File	IDX				Derived	IDX_MAP	122
Mosaic Input Coregistration Map	ICM				Derived	ICM_MAP	122
Exposure Time Compensated Color	C**						
Raw DN Color		D**					
Zenith Scaled Color			Z**				
Masked Color				M**			
Color, No Illuminant, 15-bit scaled integer, linear space	CNR	DNR	ZNR	MNR	Derived	IMAGE	125
Color, No Illuminant, 12-bit scaled integer, linear space	CNS	DNS	ZNS	MNS	Derived	IMAGE	125
Color, No Illuminant, Float, linear space	CNF	DNF	ZNF	MNF	Derived	IMAGE	125
Color, No Illuminant, Byte, linear space	CNB	DNB	ZNB	MNB	Derived	IMAGE	125
Color, No Illuminant, Byte, Gamma-corrected	CNG	DNG	ZNG	MNG	Derived	IMAGE	125
Color, Standard Illuminant, 15-bit scaled integer, linear space	CSD	DSD	ZSD	MSD	Derived	IMAGE	125
Color, Standard Illuminant, 12-bit scaled integer, linear space	CSS	DSS	ZSS	MSS	Derived	IMAGE	125
Color, Standard Illuminant, Float, linear space	CSF	DSF	ZSF	MSF	Derived	IMAGE	125
Color, Standard Illuminant, Byte, linear space	CSB	DSB	ZSB	MSB	Derived	IMAGE	125
Color, Standard Illuminant, Byte, Gamma-corrected	CSG	DSG	ZSG	MSG	Derived	IMAGE	125
Color, White Bal, Mars Illuminant, 15-bit scaled int, linear space	CPD	DPD	ZPD	MPD	Derived	IMAGE	125
Color, White Bal, Mars Illuminant, 12-bit scaled int, linear space	CPS	DPS	ZPS	MPS	Derived	IMAGE	125
Color, White Balanced, Mars Illuminant, Float, linear space	CPF	DPF	ZPF	MPF	Derived	IMAGE	125
Color, White Balanced, Mars Illuminant, Byte, linear space	CPB	DPB	ZPB	MPB	Derived	IMAGE	125
Color, White Balanced, Mars Illuminant, Byte, Gamma-corrected	CPG	DPG	ZPG	MPG	Derived	IMAGE	125
Color, White Balanced, 15-bit scaled integer, linear space	CWD	DWD	ZWD	MWD	Derived	IMAGE	125
Color, White Balanced, 12-bit scaled integer, linear space	CWS	DWS	ZWS	MWS	Derived	IMAGE	125
Color, White Balanced, Float, linear space	CWF	DWF	ZWF	MWF	Derived	IMAGE	125
Color, White Balanced, Byte, linear space	CWB	DWB	ZWB	MWB	Derived	IMAGE	125
Color, White Balanced, Byte, Gamma-corrected	CWG	DWG	ZWG	MWG	Derived	IMAGE	125

4.3.3 Mosaic Filename Convention

The InSight camera Mosaic data products are usually derived from multiple EDR or RDR data products mosaicked together, although they can also be derived from single data products.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
INSTRUMENT 1	INSTRUMENT 2	EYE	FILTER	SPECIAL	SOL	MJLTISOL	PROD TYPE	MOSAIC ID	SEQUENCE ID	PROJECTION	GEOM CORR.	FRAME	BRT CORR.	EPOCH	DESCRIPTION	CREATOR	VERSION	.	EXTENSION																					

Figure 4-3: Mosaic File Naming Convention

Table 4-5: Mosaic Filename Fields

Field	Position (size, type)	Description
Instrument 1	1 (1,a)	First Instrument Identifier <ul style="list-style-type: none"> • C : InSight Context Camera • D : InSight Deployment Camera
Instrument 2	2 (1,a)	Second Instrument Identifier <ul style="list-style-type: none"> • C : InSight Context Camera • D : InSight Deployment Camera • X : Use when there are more than two instruments (unlikely for InSight) • _ : Use when there is only one instrument
Eye	3 (1,a)	Identifies the eye of the data used to make this mosaic. For stereo mosaics, indicates which eye should view this product (in the unlikely event these two definitions are in conflict, the second use prevails for stereo mosaics). <ul style="list-style-type: none"> • A : Anaglyph • X : Mixed, used for by products which have more than one image. Since InSight does not have a true stereo camera, it is sometimes necessary to mix single, mono images together to create a composite product. • L : Left eye • M : Mono • R : Right eye • S : Stereo 2-band <p>Note that “eye” is an artificial concept since there is only one IDC camera; the eye is assigned when the image is acquired to assist in stereo pipeline processing.</p>

Field	Position (size, type)	Description																									
Filter	4 (3,a)	<p>Indicates the filter or color used for each band of the mosaic. Single-band mosaics should repeat the same character 3 times. Three-band color spaces may either list the three one-band components (e.g. “RGB”) or repeat the 3-band code 3 times (e.g. “FFF”). Valid values:</p> <ul style="list-style-type: none"> • 0 or _: Filter not applicable <table border="1" data-bbox="634 562 1243 804"> <thead> <tr> <th>Color Type</th> <th>3-Band</th> <th>Band 1</th> <th>Band 2</th> <th>Band 3</th> </tr> </thead> <tbody> <tr> <td>RGB</td> <td>F</td> <td>R</td> <td>G</td> <td>B</td> </tr> <tr> <td>XYZ</td> <td>T</td> <td>X</td> <td>Y</td> <td>Z</td> </tr> <tr> <td>xyY</td> <td>C</td> <td>J</td> <td>K</td> <td>L</td> </tr> <tr> <td>HSI</td> <td>P</td> <td>H</td> <td>S</td> <td>I</td> </tr> </tbody> </table> <p>Other color flags may be defined in the future.</p>	Color Type	3-Band	Band 1	Band 2	Band 3	RGB	F	R	G	B	XYZ	T	X	Y	Z	xyY	C	J	K	L	HSI	P	H	S	I
Color Type	3-Band	Band 1	Band 2	Band 3																							
RGB	F	R	G	B																							
XYZ	T	X	Y	Z																							
xyY	C	J	K	L																							
HSI	P	H	S	I																							
Special	7 (1,a)	<p>Special Processing flag. The special processing character is used to indicate off-nominal or special processing of the image. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than “_” which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of Sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different Sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges of Sols.</p> <p>An ASCII text file will be maintained containing all special processing designators that are used, the Sols they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.</p>																									
Sol	8 (4,i)	<p>Primary Sol (or day of year) included in the mosaic. Nominally, this is the Sol/DOY of the last (in time order, or highest) SCLK input image included in the mosaic. This can change in off-nominal cases for special circumstances.</p> <p>Solar day number (Sol), with landing day being zero. Zero-filled, decimal number.</p> <p>Testbed and Cruise data may instead treat this as Day of Year in the form YDDD where “Y” is the least significant digit of the year. Zero-filled decimal number.</p> <p>Note about “Y”: If the mission extends beyond year 2019, then “A” will be used for 2020, “B” for 2021, and so on. The “Y” field will never be less than “5” (2015). All YDDD data generated after landing must have an epoch of “T”, to distinguish them from surface data past sol 5000</p>																									

Field	Position (size, type)	Description
Multisol	12 (1,a)	Flag indicating that a significant percentage of the data content was acquired on more than a single Sol, i.e. across multiple Sols. Specification of this flag is at the discretion of the mosaic product's producer, who determines what percentage is "significant". Valid values: <ul style="list-style-type: none"> • _ : Flags mosaic as single sol (insignificant multiple sol contribution) • X : Flags mosaic as containing significant content from multiple sols
Product Type	13 (3,a)	Product Identifier. See Table 4-4 for details.
Mosaic ID	16 (2,i)	Identifies which mosaic this is (or more precisely, which mosaic ID was used as input). Mosaics made from more than one ID should have an arbitrary ground-assigned ID that contains at least one alpha (non-numeric) character. Images must share the same Eye, Epoch, Sequence ID, and Mosaic ID to be combined in an automatically-generated mosaic. The sol must also match if the mosaic ID is 01-09.
Sequence ID	18 (4,a)	Context specific field for use by camera uplink team, generally used to indicate the command sequence used for the images the mosaic was generated from. Specific values will be assigned by the uplink team. Mosaics made from images with multiple sequence ID's may use the primary sequence ID, or a ground-assigned ID. Ground-assigned IDs should have at least one alpha (non-numeric) character.
Projection	22 (3,a)	Projection type. Valid values: <ul style="list-style-type: none"> • CYL - Cylindrical • CYP - Cylindrical-Perspective • ORT - Orthographic • ORR - Orthorectified • PER - Perspective • POL - Polar • VRT - Vertical (special case of Orthographic)
Geom Corr.	25 (1,a)	Geometric correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are: <ul style="list-style-type: none"> • _ : No correction (raw pointing) • A : Auto-correction via tiepointing • F : Auto-correction via tiepointing and auto-registration with fiducials • G : Auto-correction via tiepointing and manual registration with fiducials • T : Manual tiepointing • R : Manual tiepointing and auto-registration with fiducials • M : Manual tiepointing and manual registration with fiducials • O : Other correction not listed above

Field	Position (size, type)	Description
Frame	26 (1,a)	Coordinate system (frame) type used for the mosaic projection. Valid values: <ul style="list-style-type: none"> • S : Site frame • L : Local Level frame • R : Lander (IDA) frame (“R” is heritage from Rover) • U : Untilt (CYP only, is Lander frame with rotation) • O : Other
Brt Corr.	27 (1, a)	Brightness correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are: <ul style="list-style-type: none"> • _ : No correction • B : Automatic brightness adjustment (multiplicative and/or additive factor applied to each frame) • M : Manual brightness adjustment (same factors as “B”) • V : Anti-vignetting adjustment applied to some or all frames • G : General brightness correction • A : General brightness correction that can vary across the frame (automatic) • O : Other correction not listed above
Epoch	28 (1,a)	Mission epoch identifier <ul style="list-style-type: none"> • _ : Surface operations, or Operational Readiness Tests. • T : Testbed or ATLO data • C : Flight model, cruise
Description	29 (7,a)	General purpose mosaic identifier. Can be set to anything to help identify the mosaic, such as target name, panorama name, theme name, method of production, etc. Valid values include A-Z, 0-9, and underscore. Must always pad to 7 characters, using underscores as necessary.
Creator	36 (1,a)	ID for the team that created this product: <ul style="list-style-type: none"> • M : MIPL • _ : Undefined / Other Note: Other creator codes will likely be added in the future.
Version	37 (1,a)	Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists. The value can contain both digits and letters. <ul style="list-style-type: none"> • 1 – 9, then A-Z • “_” : indicates overflow in case all version numbers are used.
.	38 (1,a)	Always set to “.”

Field	Position (size, type)	Description
Extension	39 (3,a)	File extension: <ul style="list-style-type: none"> • VIC : Used for VICAR files • IMG : Same as .VIC; occasionally used for compatibility with some old software. • JPG : Used for JPEG compressed (non VICAR) image files. • PNG : Used for PNG-formatted image files (typically browse images). • TXT : used for ASCII files, such as TAU reports. • xml : PDS4 Label file

4.3.4 Terrain Mesh Filename Convention

Terrain Mesh products consist of two types, wedges and unified meshes. Per-image files, called “wedges”, are made from a single source image (stereo pair). They follow the single-frame RDR filename convention (Section 4.3.1), except they use extensions “.iv”, “.ht”, and “.rgb” for the Inventor mesh (used by RSVP) and “.obj”, “.mtl”, and “.png” for the OBJ-format mesh (used by PDS). Unified meshes combine multiple wedges into a single terrain mesh (in a manner analogous to mosaics) and are identified using the filename convention in this section. Unlike the other filenames, terrain mesh names are variable length, due to the leading Instrument field being 1, 2, or 3 characters. The position numbers below assume 3 characters for clarity of the table, but actual position could be one or two less (in reality, virtually all InSight meshes come from a single instrument). Unified meshes are created in Inventor format for use in operations. They can be created in OBJ format as well, although this is not expected for InSight. A Meshlab “project” file (Section 5.3.4), also uses the unified mesh filename.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
INSTRUMENT			-	EYE	SOL			MJLTISOL	PROD TYPE			LINEARIZATION	MESH ID	SEQUENCE ID			SPECIAL	EPOCH	DESCRIPTION					CREATOR	VERSION	.	EXTENSION							

Figure 4-4: Mesh/DEM File Naming Convention

Table 4-6: Mesh/DEM Filename Fields

Field	Position (size, type)	Description

Field	Position (size, type)	Description
Instrument	1 (3,a)	<p>Instrument Identifier. This is a variable-width field; include only as many characters (up to 3) as are needed. The “primary” (as determined by the data producer) instrument should be listed first.</p> <ul style="list-style-type: none"> • C : InSight Context Camera • D : InSight Deployment Camera • O : Orbital data (typically HiRISE from MRO but not necessarily) • X : Contains more than three instruments (must be 3rd character). Unlikely to be needed for InSight.
_	4 (1,a)	Always set to “_”
Eye	5 (1,a)	<p>Identifies the eye of the data used to create the mesh. Since meshes are created from stereo data, the “eye” indicates the reference eye, i.e. which eye matches the geometry of the XYZ file.</p> <ul style="list-style-type: none"> • A : Anaglyph • X : Mixed, used for by products which have more than one image. Since InSight does not have a true stereo camera, it is sometimes necessary to mix single, mono images together to create a composite product. • L : Left eye • M : Mono • R : Right eye • S : Stereo 2-band <p>Note that “eye” is an artificial concept since there is only one IDC camera; the eye is assigned when the image is acquired to assist in stereo pipeline processing.</p>
Sol	6 (4, i)	<p>Primary Sol (or day of year) included in the mesh. Nominally, this is the Sol/DOY of the last (in time order, or highest) SCLK input image included in the mesh. This can change in off-nominal cases for special circumstances.</p> <p>Solar day number (Sol), with landing day being zero. Zero-filled, decimal number.</p> <p>Testbed and Cruise data may instead treat this as Day of Year in the form YDDD where “Y” is the least significant digit of the year. Zero-filled decimal number.</p> <p>Note about “Y”: If the mission extends beyond year 2019, then “A” will be used for 2020, “B” for 2021, and so on. The “Y” field will never be less than “5” (2015). All YDDD data generated after landing must have an epoch of “T”, to distinguish them from surface data past sol 5000.</p>
Multisol	10 (1,a)	<p>Flag indicating data content from multiple Sols. Valid values are:</p> <ul style="list-style-type: none"> • _ - Data is from a single sol • X - Data is from multiple sols <p>Unlike mosaics, there is no discretion applied; multiple sols always get an X.</p>
Product Type	11 (3, a)	Product Identifier. See Table 4-4 for details.

Field	Position (size, type)	Description
Linearization	14 (1, a)	<p>Linearization flag:</p> <ul style="list-style-type: none"> • L : Mesh was derive from linearized images • _ : Mesh was derived from non-linearized (raw geometry) images • X : Mesh contains a mixture of linearized and non-linearized images <p>Note that linearization requires a stereo partner; this is indicated by the Stereo ID field. This corresponds to “Actual” linearization mode in MSL parlance (“Nominal” linearization is not possible for a single camera). If an image has multiple stereo partners, the linearized images will be different for each partner. For this reason, InSight uses non-linearized geometry for almost all products; linearized images are intermediate products and will not normally appear in meshes.</p>
Mesh ID	15 (2, i/a)	<p>Identifies which mesh this is (or more precisely, which mesh ID was used as input). Meshes made from more than one ID should have an arbitrary ground-assigned ID that contains at least one alpha (non-numeric) character.</p> <p>Images must share the same Eye, Epoch, Sequence ID, and Mesh ID to be combined in an automatically-generated mesh. The SOL must also match if the mesh ID is 01-09.</p>
Sequence ID	17 (4, i/a)	<p>Context specific field for use by camera uplink team, generally used to indicate the command sequence the image was acquired from. Specific values will be assigned by the uplink team.</p> <p>EDRs must have numeric values in the range 0000-4095.</p> <p>RDRs may have the above numeric values, or they may have alphanumeric characters, indicating a ground-assigned ID. Ground-assigned IDs should have at least one alpha (non-numeric) character.</p> <p>The sequence ID must match for stereo pairs, and auto-generated meshes and mosaics. Ground-assigned IDs thus allow different pairings of data than anticipated when the images were acquired.</p>

Field	Position (size, type)	Description
Special	21 (1,a)	<p>Special Processing flag. The special processing character is used to indicate off-nominal or special processing of the image. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc.</p> <p>The meaning of any individual character in this field (other than "_" which means nominal processing) will be defined on an ad-hoc basis as needed during the mission. Within one Sol or a range of Sols, the character will be used consistently. So, this field can be used to group together all derived products resulting from one kind of special processing. An attempt will be made to maintain consistency across different Sols as well, but this may not always be possible; thus the meaning of characters may change across different individual or ranges of Sols.</p> <p>An ASCII text file will be maintained containing all special processing designators that are used, the Sols they relate to, and a description of the special processing that was done. This file will be included in the PDS archive.</p>
Epoch	22 (1, a)	<p>Mission epoch identifier</p> <ul style="list-style-type: none"> • _ : Flight model, surface ops. • T : Testbed • C : Flight model, cruise
Description	23 (7,a)	<p>General purpose mesh identifier. Can be set to anything to help identify the mesh, such as target name, panorama name, theme name, method of production, etc. Valid values include A-Z, 0-9, and underscore. For Meshes, this field must ensure uniqueness in like-named mesh filenames of identical Site, Drive, Instrument, Mesh ID and Sequence ID. Must always pad to 7 characters, using underscores as necessary.</p>
Creator	30 (1,a)	<p>ID for the team that created this product:</p> <ul style="list-style-type: none"> • M : MIPL • _ : Undefined / Other <p>Note: Other creator codes will likely be added in the future.</p>
Version	31 (1,a)	<p>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists. The value can contain both digits and letters.</p> <ul style="list-style-type: none"> • 1 – 9, then A-Z • “_” : indicates overflow in case all version numbers are use.
.	32 (1,a)	Always set to “.”

Field	Position (size, type)	Description
Extension	33 (3,a)	File extension: <ul style="list-style-type: none">• iv - Unified Terrain Mesh RDR in Inventor ASCII format• mod - ASCII index list of mesh component Height map (.ht) and Vertex (.iv) files• obj - Mesh in standard OBJ format• mtl - Materials file for OBJ format• mlp – Meshlab project file

5. DETAILED DATA PRODUCT SPECIFICATIONS

This section describes each of the data product types in detail, including an overview of the algorithms used to produce them. Type codes in parentheses in the section headings refer back to the type codes in Table 4-4. The primary type codes are shown in the headings. For clarity, not all type codes are in the headings; for example overlay types (**O), masks (**M), and filled data (**F) are not typically listed. Table 4-4 has a complete listing of all type codes.

5.1 EDR Data Product Specification

The data packaged in the camera data files will be decoded, decompressed camera image data in single frame form as an Experiment Data Record (EDR). The Full Frame form of a standard image data file has the maximum dimensions of 1024 lines by 1024 samples. The other camera data files and their data sizes are listed in Table 5-1.

Table 5-1: Camera EDR Data Products

Image Type	Image Size	Pixel Size (bits)	Description
Image EDR, Raw	1024 lines x 1024 samples	16-bit signed integer, 1 band	Uncompressed full sized, full resolution data product, with Bayer pattern still encoded in it.
Image EDR, Color	1024 lines x 1024 samples	8-bit unsigned integer, 3 bands	3-band color image, with Bayer pattern removed and 12 to 8 bit scaling applied onboard. Converted to JPEG for transmission, but uncompressed into a raster image.
Original JPEG (EJP)	1024 lines x 1024 samples	JPEG-compressed 8-bit unsigned integer, 3 bands	While not technically an EDR, it is closely related and is included here for convenience. This file retains the original JPEG encoding as sent by the spacecraft. Note that the image is <i>not</i> rotated like EDRs are.
Reference pixels (ERP)	1024 lines x 32 samples	16-bit signed integer, 1 band	These are Dark pixels bookending (pre- and post-) image pixels during serial register readout. There are 17 “pre-” Reference and 14 “post-” Reference pixels, plus 1 for the camera hardware serial number (left-shifted by 4 bits if 12-bit data).

5.1.1 Image EDR (EDR)

Image EDRs are the fundamental image product from which all other RDRs are derived. EDRs are the closest representation possible to the data that came off of the CCD. This means that compression used for transport has been removed (i.e. the products are uncompressed raster images), and any irreversible operations done onboard (such as de-Bayering) are also part of the EDR. However, decompressing is *not* done as part of the EDR process; decompressed products are ILT RDRs.

There are two flavors of EDR for InSight, depending on whether or not the data was compressed onboard: Raw, and Color EDRs.

In either case, there may be multiple versions of any EDR due to retransmission of telemetry packets from the lander and/or upstream ground segments. Missing data will be identified within each image by

fill (black pixel) data. Because it is not possible to reconstruct a JPEG image with missing data (the JPEG stream does not include restart markers), any data after a gap is zeroed out in the EDR (this shows up as a black area on the right side of the image, due to image-to-detector rotation). In the event of retransmissions, new versions of an image will be stored with a higher version number in the filename. The highest version number is always the most complete.

Note that the cameras are mounted “sideways” (Maki et al 2018, [3]). This means that for the ICC and normal arm orientations for the IDC, the sky appears on the side of the CCD rather than the top. In order to ease interpretation and use of the images, the EDR is rotated to compensate for this, so the sky is up in the images. This rotation follows through into all RDRs as well. Only the EJP product (Section 5.1.2) is not rotated in this way. See Figure 5-1: EJP showing orientation of camera (left). EDR after rotation (right).

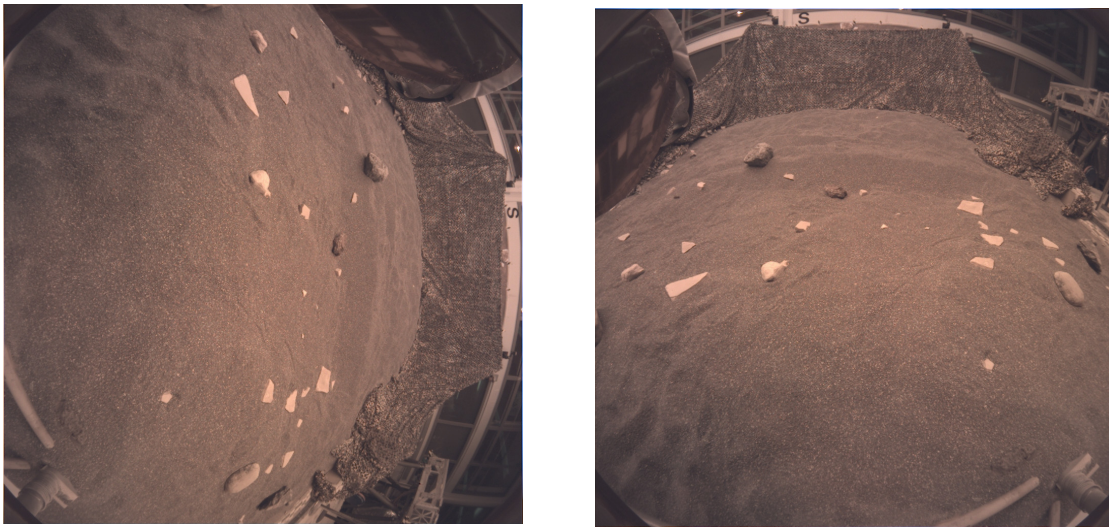


Figure 5-1: EJP showing orientation of camera (left). EDR after rotation (right).

5.1.1.1 Raw EDRs

For Raw EDRs, the original 12-bit data from the CCD is downlinked in 16-bit integers (with 4 bits of wasted space), uncompressed. This data is exactly what comes off the CCD; specifically, the Bayer pattern has not been decoded into color pixels. It is expected that data in this format will be somewhat rare after initial instrument checkout, although it is the only lossless data transmission mode and the only way to obtain reference pixels.

The raw EDR is thus a 16-bit signed integer, 1-band image in VICAR format, with a detached PDS4 label. Although the image is transmitted as 1056x1024, the 32 non-image pixels are stripped off into a separate Reference Pixels product (see Section 5.1.3), leaving the EDR at 1024x1024 pixels.

The Bayer pattern must be removed (de-Bayering or demosaicking) in order for the image to be usable. This is done in RDR processing. See Section 5.2.1.1.

5.1.1.2 Color EDRs

Color EDRs are created onboard the lander. The data from the CCD is transformed as detailed in Section 2.4. This color image is then compressed into a standard JPEG image for transmission, using a configurable quality setting (0-100). This data is decompressed on the ground.

The Color EDR is thus an 8-bit (unsigned byte), 3-band color image in VICAR format, with a detached PDS4 label. It is not possible to get Reference Pixels from a color EDR.

The color of the EDR is only an approximation; there has been no color correction applied other than the onboard blue boost (Section 2.4.2). The blue boost creates a more natural-looking EDR, but it is not a color-corrected product (see Section 5.2.2.2.2).

5.1.1.3 Subframe, Downsampling, and Thumbnails

InSight does not support subframing or downsampling of images, nor does it support thumbnails. These are mentioned here by way of contrast with the other Mars *in-situ* missions (MER, Phoenix, MSL, Mars 2020), all of which do support these concepts.

5.1.2 Original JPEG (EJP)

The original JPEG file as downlinked in telemetry is saved in the EJP type. While not technically an EDR, it is described here because it is very closely related to the EDR. The EJP is saved in order to preserve the originally downlinked image, in case questions about image artifacts arise that can be answered by examining the original JPEG stream. No processing is done on EJP files. Both the ICC and IDC camera CCDs are mounted “sideways”, thus “up” in these images is not the same as the EDR (which is rotated to make it “upright” for human viewing).

The EJP file should *not* be considered a browse image and should not be used in lieu of the EDR or browse image for any purpose whatsoever – its *only* purpose is to preserve the original JPEG stream in case that is ever needed.

Note that EJP’s do not have their own PDS4 label; they are referred to as a supplemental file in the primary EDR label.

5.1.3 Reference Pixels (ERP)

The onboard CCD array has 17 "pre-Reference" dark (non-photo-sensitive) pixels (12-bits) located at the beginning and 14 "post-Reference" dark pixels (12-bits) located at the end of each row. Following the last "post-Reference" dark pixel, at the very end of each row, is an additional pixel containing the camera hardware serial number (left-shifted by 4 bits if 12-bit data).

These reference pixels are used for camera calibration. They are only downlinked as part of Raw (uncompressed) EDRs; reference pixels are unavailable for JPEG EDRs. Although they are downlinked as part of the image area (as a 1024 x 1056 array), the telemetry processor strips off the reference pixels and puts them in a separate ERP file. Thus the actual raw image EDR contains only the 1024 x 1024 photosensitive pixels.

The ERP product is a single-band, 1024-line by 32-sample, 16-bit signed integer image in VICAR format, with a detached PDS label.

5.2 Single-Image RDR Data Product Specification

RDR data products described in this document will be generated by MIPL personnel using the Mars Suite of VICAR image processing software (see Table 1-1). The RDRs produced correspond to the “partially processed”, “calibrated”, and “derived” PDS processing levels (see Table 3-2). The input will be one or more image EDR or RDR data products and the output will be formatted according to this SIS. Additional metadata may be added by the software to the product label.

There may be multiple versions of image RDRs; the highest numbered version is always the most recent (and by definition, “best” – meaning created with the most complete EDR and including any reprocessing that may have been needed). The RDR data products will be placed into FEI for distribution, saved on the SPeX, and archived in the PDS.

Processing is different for each image RDR type, as described in this section. In general, each RDR process inherits the metadata from its source EDR/RDR, modifying a portion of the metadata as necessary to reflect the subsequent output RDR. Thus, the bulk of the labels are the same throughout the EDR->RDR processing chain.

5.2.1 Common Processing

Although the RDR types vary widely, certain types of processing are common to multiple types of RDR files. These common areas are described below.

5.2.1.1 Bayer color processing (BAY)

The IDC and ICC cameras generate images using a Bayer-pattern filter over the CCD (see Section 2.3). This Bayer pattern must be removed and the images converted to RGB color before they can be used.

The InSight project uses the Malvar algorithm (Malvar 2004, [14]), which is the same as that used by the Mastcam, MAHLI and MARDI cameras on MSL. This is a very simple algorithm to implement (consisting of a set of simple convolution kernels) yet delivers very good results in most cases.

The Malvar algorithm is implemented both onboard and in ground software. The onboard algorithm is used to create Color EDRs (see Section 2.4.3), while the ground algorithm is used to create color images out of Raw EDRs (creating the BAY type of RDR). Note that the BAY type of RDR is not created for color EDRs, as it is not necessary.

The Malvar algorithm is certainly not the only possible de-Bayering (demosaicking) algorithm; the literature contains many algorithms. For any raw EDRs, users may experiment with these algorithms as desired. However, it is not possible to recover the original Bayer-pattern image from Color EDRs, meaning there is no choice for these but the onboard Malvar. Note that raw images will be exceedingly rare; we might not get any during the course of the mission.

5.2.1.2 Color RDRs

Regardless of whether de-Bayering is done onboard or on the ground, RDRs at the ILT (inverse LUT)

stage are identical. RDRs that contain image data, such as ILT, RAD, RAS, etc., are produced by processing all three color bands simultaneously. This is represented as F (for Full Color) in the Filter field of the filename. Note that for raw images, these products are produced both on the raw Bayer-pattern images (filter code “_”) and the de-Bayered images (filter code “F”).

It is important to realize that these images are only approximate color. Color-corrected products are described in Section 5.2.2.2.2.

RDRs related to image geometry (e.g. disparity, XYZ, all downstream products) are created using only the Green band extracted from the color image. The Green band is used because there are two green pixels per Bayer cell, as opposed to one for red and blue, leading to a slightly higher resolution image. These are indicated by a “G” in the Filter field of the filename.

5.2.1.3 Geometrically Corrected Images (Linearization)

EDRs and single-frame RDRs are described by a camera model. This model, represented by a set of vectors and numbers, permits a point in space to be traced into the image plane, and vice-versa.

EDR camera models are derived by acquiring images of calibration targets with known geometry at a fixed azimuth/elevation. The vectors representing the model are derived from analysis of these images. These vectors are then translated and rotated based on the actual pointing of the camera to represent the conditions of each specific image. The resulting vectors make up the “camera model” for the EDR. (see [15] through [23]).

The IDC uses a CAHVOR model, while the ICC uses a more general CAHVORE model. Both model types are nonlinear and involve some complex calculations to transform line/sample points in the image plane to XYZ positions in the scene. To simplify this, the images are “warped”, or reprojected, in a process often called “linearization”, such that they can be described by a linear CAHV model. See Figure 5-2 for a visual comparison between a ICC image EDR (left) and the ICC “linearized” image RDR (right).

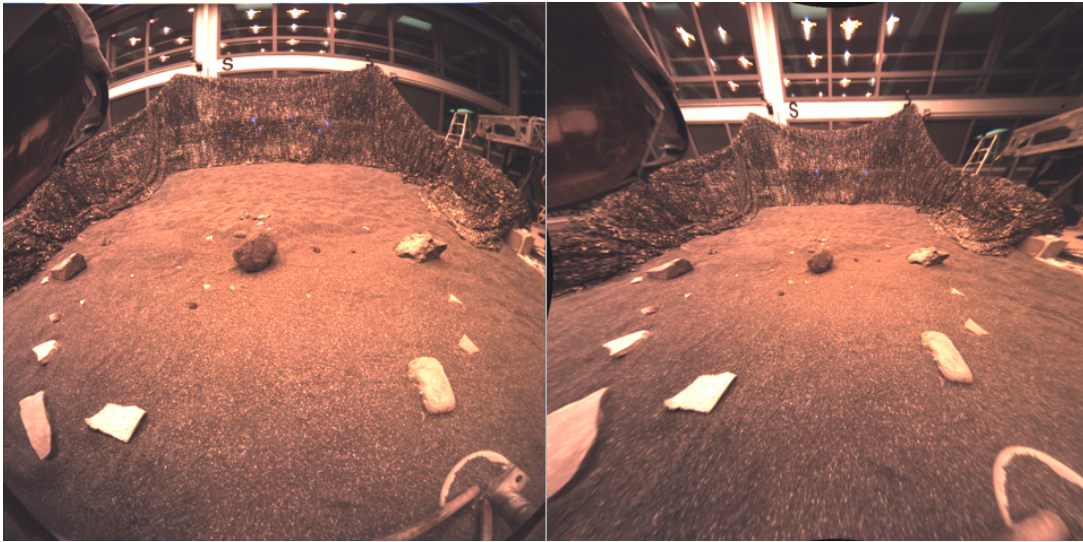


Figure 5-2: ICC Image EDR (left) vs “Linearized” ICC Image RDR (right)

This linearization process has several benefits:

- 1) It removes geometric distortions inherent in the camera instruments, with the result that straight lines in the scene are straight in the image.
- 2) It aligns the images for stereo viewing (applies only to IDC), known as epipolar alignment. Matching points are on the same image line in both left and right images, and both left and right models point in the same direction.
- 3) It facilitates correlation, allowing the use of 1-D correlators in some cases.
- 4) It simplifies the math involved in using the camera model.

However, it also introduces some artifacts in terms of scale change and/or omitted data [24]. In the case of InSight, these distortions can become severe due to the arm geometry, so linearized images are used only for intermediate products (more on this below).

The linearized CAHV camera model is derived from the EDRs camera model by considering both the left and right eye models and constructing a pair of matched linear CAHV models that conform to the above criteria. For details on this algorithm see references [15] through [23].

There are two types of linearization. When stereo is implemented via two cameras at a fixed distance from each other, as in most other Mars surface missions, “nominal” linearization can be used. The stereo partner can be predicted based on the fixed camera geometry, and this model can be used for linearization. This allows each image to be processed independently, without the need to find the stereo partner (or even acquire the image), yet provides the same results as if the match had been performed.

However, that is not available for InSight, because its only “stereo camera” is the arm mounted IDC. There is no virtual partner to linearize to. Thus, all IDC stereo images are linearized with the *actual* stereo partner. This means both images must be available before linearization can be done.

Regardless of linearization type, the image is then projected, or warped, from the CAHVOR/CAHVORE model to the CAHV model. This involves projecting each pixel through the EDR camera model into space, intersecting it with a surface (which matters only for the CAHVORE-based ICC and is a 1m radius sphere centered on the camera), and projecting the pixel back through the CAHV model into the output image. The CAHV camera model has the following parameters (all tuples of 3 numbers):

- C – The 3D position of the entrance pupil
- A – A unit vector normal to the image plane pointing outward (towards C)
- H – A vector pointing roughly rightward in the image; it is a composite of the orientation of the CCD rows, the horizontal scale, and the horizontal center
- V – A vector pointing roughly downward in the image; it is a composite of the orientation of the CCD columns, the vertical scale, and the vertical center

If P is a point in the scene then the corresponding image locations x and y can be computed from:

$$x = \frac{(P - C) H}{(P - C) A}$$

$$y = \frac{(P - C) V}{(P - C) A}$$

For details on the camera model math and calibration and more description of the CAHV-model family, see references ([15] through [23]). Note that the X and Y positions above are 0-based coordinates; i.e., the coordinate (0,0) is the center of the upper-left pixel. This is different than the common PDS convention of 1-based coordinates, where (1,1) is the center of the upper-left pixel, used elsewhere in this document.

The requirement to use actual linearization has important implications for InSight. First, the geometry of the arm is such that the images sometimes need to be severely distorted to achieve epipolar alignment. More importantly, it means that the linearized geometry changes depending on what image is used as a partner. Not only that, it changes due to pointing corrections of either image of the stereo pair. Pointing correction (Section 5.4.2) is very important to InSight processing in order to correct inaccuracies in arm position knowledge. If linearized images were used, it means pointing correction would change those linearized images, meaning the entire stereo processing chain (including the time-consuming correlation) would have to be re-done. This is not practical operationally.

Therefore, InSight uses non-linearized images almost exclusively. Linearized images are still created, because the correlation process needs them – it correlates on linearized images before “unlinearizing” the result and re-correlating – but the linearized images are not used otherwise.

The exception to this is for human viewing of stereo images (Figure 5-3), where epipolar alignment is critical and thus linearized images must be used.

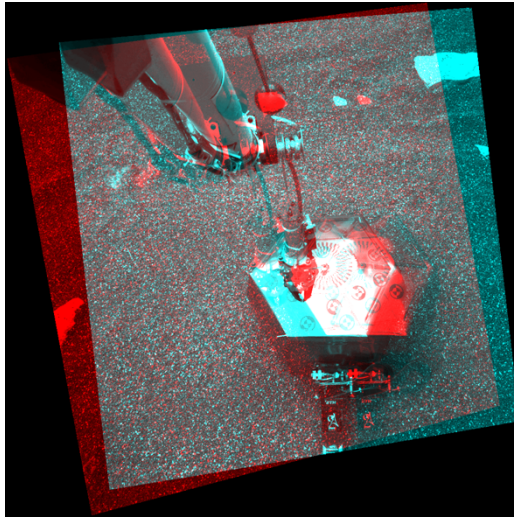


Figure 5-3: Left and Right Eye IDC Images Linearized to each other for stereo viewing

5.2.1.4 Image Overlays (O)**

Many image RDR types represent some quantity other than intensity of light, such as XYZ or slope. The value at each pixel indicates the measurement of the quantity at the corresponding point in the original image. These types can be visualized in various ways, such as a color rainbow or by contours, and then overlaid on a background of the EDR or other intensity image. This is the same approach used by the marsviewer operations and archive viewing tool [26]. The visualization created by Marsviewer can be saved as a color image, called an overlay image. Many examples of these overlays are shown in the figures in this section (for example Figure 5-5, XYZ overlay, or Figure 5-6, Range overlay). Overlays are generally indicated by a product type ending with “O” in the filename, and are not further discussed in this document.

5.2.1.5 Filled Images (F)**

Many RDR types do not achieve full coverage, e.g. the correlator is unable to find a solution at every point or the XYZ point failed the various filters. These “holes” are preserved in the RDRs using some value to indicate no solution (specified by the MISSING_CONSTANT labels). For ops work, it is critical to know where the holes are, so they are preserved in all nominal RDRs. Some RDRs created for purposes other than operations (such as photometry processing, or public engagement) have these holes filled in using an interpolation mechanism. These are referred to as “filled” RDRs. Filled RDRs are generally indicated by a product type ending with “F” in the filename (but not all trailing “F”s mean Filled). Use of Filled products is not expected for InSight, but the file types exist in case they are needed.

5.2.1.6 Lander Volume Exclusion Image Masks (MXY, M)**

MIPL creates “Lander Volume Exclusion Mask” files that can be applied to several types of RDR. In general, they are used to filter out lander features from generated terrain products, as well as the horizon and other undesirable features. For InSight, they are used only to mask out the arm and (stowed) grapple for IDC images. They are single-band, byte files corresponding to the source image, where 255

indicates the corresponding pixel should be removed, or 0 indicates the pixel should remain in the output. This product is specifically the MXY file; other masks may exist for other purposes and generally start with M (e.g. MSK).

5.2.1.7 Masked Images (M)**

The exclusion masks, or other masks, can be combined with any RDR to create a Masked image. The contents are identical to the source RDR except that where the mask is 255, the value is set to the value specified by MISSING_CONSTANT (usually 0). Masked files typically have an “M” as the third character of the product type (exception: masked ARM is called ARK). Masked files are important for InSight because of the constant presence of the arm in the IDC images. The most commonly used masked products are ILM (for mosaics) and XYM (for meshes).

5.2.1.8 EDR-like RDRs

The type code “EDR” can only be used for actual EDRs. Most of the time, linearization is indicated by setting the “linear” field in the filename to L. However, if linearization is applied directly to an EDR, that would end up creating an “EDRL” type of RDR, which is illegal – once the product is linearized, it is by definition no longer an EDR. To avoid confusion between these and actual EDRs, the type code is changed to LIN (plus “linear” is set to L). Similarly, the BAY code is used to indicate an EDR that has been de-bayered and would otherwise be distinguished only by the filter field. Both of these avoid having any RDRs that use the “EDR” type code.

5.2.2 Image RDRs

This section describes the RDRs that are traditional images, where pixel values relate to the intensity of light recorded by the camera. This is distinguished from geometry RDRs such as XYZ and surface normals, and instrument placement products.

5.2.2.1 Inverse LUT RDR (ILT, ILM, ILC, ILP)

The ILT RDR is produced by MIPL to reconstruct the original 12-bit pixels generated by the camera sensors. If the EDR is already in 12-bit format (Raw EDR), ILT is simply a copy. However, Color EDRs (Section 5.1.1.2), which are de-Bayered on-board, have been scaled to 8 bits using a Lookup Table (LUT) before transmission (this is often called “companding”). In this case, an Inverse LUT (ILUT) is used to recover (close to) the original 12-bit data from the 8-bit EDR. The LUT is a square-root encoding table that compresses the high end (bright) while preserving low-end (dark) detail. The ILUT recovers the original linear intensity space, although with loss of precision due to the reduction in bits.

All downstream processing is based on 12-bit data. The ILT product can thus be considered the “base” product, as it is the same regardless of whether the EDR was Raw or Color.

The ILM is the ILT with the arm mask applied (Section 5.2.1.6). It is used as the basis for most mosaics, because the masked-out arm appears transparent to the mosaic programs.

The ILC type has had additional processing to remove certain instrument artifacts, such as noise spikes or dead pixels. Due to the short duration of imaging operations for InSight, it is not expected this will be needed, but it is retained as a placeholder.

The ILP type represents an ILT or ILC that has been pointing-corrected (see Section 5.4.2) or re-localized based on ground analysis. It is expected that a special processing flag will be set for any products employing this type.

5.2.2.2 Radiometrically Corrected RDR (RA*, IO*, RI*, RZ*)

There are 4 different kinds of radiometrically corrected products. “RA*” (including RDM,RSM) have been corrected to absolute radiance units of W/m²/nm/steradian. “RI*” products have been corrected for instrument effects only, and are in units of DN. “IO*” products are radiance factor (I/F) and are dimensionless. “RZ*” products are based on “RA*” but have been further scaled based on solar elevation and tau. Within each kind, the values may be represented as floating-point values, or scaled to integers for ease of manipulation.

Independent of type, there are potentially multiple methods of performing radiometric correction, distinguished by the RADIOMETRIC_CORRECTION_TYPE keyword. However, as of this writing, only MIPLRAD has been implemented.

5.2.2.2.1 MIPLRAD Method

This refers to radiometric correction of any camera instrument data systematically performed by MIPL to meet tactical time constraints. The resulting rad-corrected RDRs are integrated into terrain mesh products used for traverse planning. This method is typically less precise than the methods used by the science teams.

MIPLRAD is a first-order correction only and should be considered approximate. MIPLRAD first backs out any onboard flat field that was performed (not applicable to InSight), and any onboard responsivity balancing (Section 2.4.2). Backing out the onboard responsivity correction allows us to treat both raw and JPEG images identically. MIPLRAD then applies the following corrections: reversal of onboard responsivity balancing, flat field, exposure time, and temperature-compensated responsivity. The result is calibrated to physical units of W/m²/nm/sr. The actual algorithm and equations used for MIPLRAD are shown below. Each correction is applied in sequence, to every pixel:

1. If responsivity balancing was applied onboard (Section 2.4.2), it is reversed. This is accomplished by dividing each band by the corresponding value in ONBOARD_RESPONSIVITY.

$$\text{output}(x,y) = \text{input}(x,y) / \text{ONBOARD_RESPONSIVITY}$$

2. For the flat-field adjustment, the x and y coordinates are used to find the corresponding pixel in the flat field, then the DN is divided by the flat field value (obtained by pre-launch calibration):

$$\text{output}(x,y) = \text{input}(x,y) / \text{flat_field}(x,y)$$

3. Exposure time is then removed. Exposure time comes from EXPOSURE_DURATION, converted to seconds:

$$\text{output}(x,y) = \text{input}(x,y) / \text{exposure_time}$$

4. The temperature responsivity is removed next. The temperature of the CCD is determined from INSTRUMENT_TEMPERATURE (first element only). The temperature is combined with parameters R0, R1, and R2, which were derived from ground calibration and come from the flat field parameter file (see Appendix B) according to the following formula:

$$\text{output}(x,y) = \text{input}(x,y) * (\text{R0} + \text{R1} * \text{temp} + \text{R2} * \text{temp} * \text{temp})$$

5. Finally, the result is (optionally) converted to integers using the RADIANCE_OFFSET and RADIANCE_SCALING_FACTOR keywords (but see Section 5.2.2.2.3):

$$\text{output}(x,y) = (\text{input}(x,y) - \text{RADIANCE_OFFSET}) / \text{RADIANCE_SCALING_FACTOR} + 0.5$$

Note that the InSight cameras were not well calibrated radiometrically. Specifically, only flat fields were obtained, not responsivity coefficients. The responsivity coefficients were derived theoretically from first principles using the known filter transmission curves, CCD characteristics, and Martian lighting conditions (see Maki 2018 [3]). There is no temperature compensation applied to these values.

5.2.2.2.2 Zenith Scaling

Radiometric correction results in a product that describes how much light entered the camera. However, lighting varies due to solar elevation and atmospheric opacity, among other things. These photometric effects mean that images taken at different times of day have different intrinsic intensities, making them difficult to combine into a mosaic or mesh, even after radiometric correction. In order to help with this, a limited form of photometric correction, known as zenith scaling, has been implemented.

Zenith scaling attempts to reconstruct what the radiometric product would look like if the sun were at the zenith on a clear day. It takes into account the solar elevation and “tau”, which is a measure of atmospheric opacity. The formula is:

$$\mu = \sin(\phi)$$

$$f = \mu e^{\left(\frac{-(\tau - \tau_{ref})}{6\mu}\right)}$$

where f is the correction factor, ϕ is the solar elevation, τ is the solar opacity (tau), and τ_{ref} is the reference tau, what is defined to be a “clear day”. For InSight the reference tau defaults to 0.3. Note that if the solar elevation is less than a minimum (a tunable parameter which defaults to 5 degrees), it is set to that minimum in the formula above.

The radiometric product is divided by this correction factor in order to create the zenith-scaled product. This has the effect of boosting the brightness of images taken later in the day.

The MIPL pipeline currently uses a constant tau (of 0.3), meaning the tau factor above has no effect on the outcome. However, a different tau could be used for special products, and the label indicates what was used. Since tau typically varies slowly, a constant tau is a reasonable assumption for most mosaics, although a variable one would compensate for darkening during dust storms. This is an item for future development.

5.2.2.2.3 Float to Integer Scaling

The various kinds of radiometrically corrected products are all based on physical units. They are therefore naturally floating-point values. However, floating-point images are more difficult to deal with and many image processing or display systems work only with integer images. For this reason, the floating point files are often scaled to integers (the original floating-point images are also generally preserved).

There are two types of scaling: static and dynamic. Both use a scaling factor and offset (which are stored in the label) to convert between integer and float:

$$\text{float} = \text{int} * \text{scale} + \text{offset}$$

The difference is in how the scale is determined (the offset is 0 in current software).

Static scaling uses single predefined scaling factors, which have been selected to best represent the usual range of intensities seen across all images. Static scaling has the very useful property of being constant from image to image, thus making mosaics or meshes much easier to create. However, in exceptionally bright (specular reflection) or low-light cases, static scaling can saturate or be too dark to see.

Dynamic scaling determines the scale factors independently for each image. This optimizes the dynamic range for each image, but makes it unsuitable for direct use in mosaics (without reconstituting the float value first) because the scaling varies from image to image. Dynamic scaling is thus better for single images, with static scaling better for mosaics. The dynamic scale factor incorporates the exposure time, maximum responsivity across bands, and average flat field (called “flat_darkening_factor” in the calibration files). The result is that the integer image is very similar to the ILT from before radiometric correction in dynamic range, with corrections only for *average* flat field, and the ratio of responsivities across bands. However, the floating-point value can still be reconstructed using the scale factor. The scale factor is computed as:

$$\text{dynamic_scale} = \text{exposure_time} * \text{flat_darkening_factor} / \text{max_responsivity}$$

5.2.2.3 Color-Corrected Products (C**/D**/Z**/M**)

The InSight cameras are MSL flight spares that have been converted to color by replacing the original grayscale detectors with Bayer-pattern color filter array (CFA) versions of the same detector type (Maki et al, (2018) [3]). In order to display color images from the cameras correctly on a computer monitor, raw color images are converted from the instrument color space (iRGB) to other color spaces by

multiplying the R, G, and B bands of the iRGB image by a 3x3 color conversion matrix. The color transform step is performed on the ground as part of the automated MIPL pipeline processing.

During the InSight operational mission, raw iRGB images will be converted to two main color spaces: the device-independent CIE (Commission internationale de l'Eclairage) XYZ color space (CIE, (1932) [31]) and the IEC sRGB color space (ITU, (2015) [32]), for use on display devices. A third colorspace, CIE xyY, is related to CIE XYZ, but it separates the luminance component (Y) from the color, leaving a 2-dimensional chromaticity component (xy). For more information on the CIE spaces the reader is referred to Wyszecki & Stiles (1982) [33]. A third class of colorspace are white-balanced color spaces. Images are white balanced by adjusting the primary (R, G, B) colors such that white objects in the scene are represented as white in the image. This is typically done by applying a 3x1 matrix multiplication (or in practice, a 3x3 matrix with non-zero coefficients along the diagonal, and zeroes elsewhere). During the surface mission white-balanced iRGB and sRGB images were generated for the operations team.

The RGB images discussed above are created and converted in linear brightness space. However, most display systems (including the sRGB standard) expect images to be gamma-encoded. Therefore, the sRGB and white-balanced images are gamma-encoded for display (which also converts them from 12 bits per pixel to 8 bits per pixel) using the function described in Maki et al (2018) [3]. It is these gamma-encoded products that should be used on most display systems. The CIE XYZ and CIE xyY products are both in floating-point format, so no gamma encoding is done on those.

Independently of all the above, there are three types of scaling that are performed on color products, which is indicated by the first letter of the RDR type code.

Those that start with C are exposure-time compensated (or normalized) color products. These are derived directly from the radiometrically corrected product (RAF), with a static scaling factor to bring the DN values of the color products into the 0-4095 range. This means the C** products are similar to the “static scaling” rad products defined in Section 5.2.2.2.3, in that the color is consistent from image to image in a mosaic, but it’s possible the dynamic range of an individual image may be too bright (saturated in places) or dark to see well.

Those that start with D are raw DN color products. These are created in the same way as the C** products, except that the scaling factor is the exposure time. This has the effect of removing exposure time compensation (it was included in the RAF source product), so the DN values are based on the raw DN’s that came from the spacecraft (modified by the average flat field). In this way they are similar to the “dynamic scaling” rad products from Section 5.2.2.2.3. The dynamic range of each image is optimized, but the overall intensity will vary from image to image in a mosaic.

Those that start with Z are zenith-scaled color products. They are just like C** except they are derived from the zenith-scaled radiometry (RZF, Section 5.2.2.2.2) rather than RAF. They are thus the best products to use for mosaics or meshes (better even than C**) because they are compensated (in an approximate way) for lighting conditions.

Finally, those that start with M are masked color products. These are one of the other types that has had a mask applied to it to remove the arm from IDC images. The M** type can be based on any of the

other (C/D/Z) color types; see the label for which it is. As of this writing, the pipeline makes MPS and MPG derived from ZPS and ZPG.

The three-letter product codes for color products follow a standard pattern:

Character 1: Color Indicator

- C = Exposure Time Compensated (or Normalized) Color
- D = Raw DN Color
- Z = Zenith-Scaled Color
- M = Masked Color

Character 2: sRGB or White Balance Indicator

- N = Neither (N/A, or XYZ/xyY color spaces)
- S = sRGB, as measured by the camera
- P = White-balanced sRGB, Mars illuminant
- W = White-balanced iRGB, Mars illuminant

Character 3: Data Type

- D = 15-bit integer, linear space
- S = 12-bit integer, linear space
- F = Float, linear space
- B = Byte, linear space
- G = Byte, gamma correction applied

Only a few of the possible product types are generated systematically by the pipeline (marked by green or yellow in Table 4-4). Any type may be created as a special product, however.

Figure 5-4 shows the pipeline processing flow for radiometric and color products. Note that floating-point versions of the products are used at all intermediate steps (except for the linear to gamma conversion) to avoid clipping due to saturation. Additional clipping in the linear to gamma conversion is not possible given how gamma conversion is defined. The same flow is used for C** and D** products; the only difference is the parameter given to marscolor when creating the CNF/DNF products.

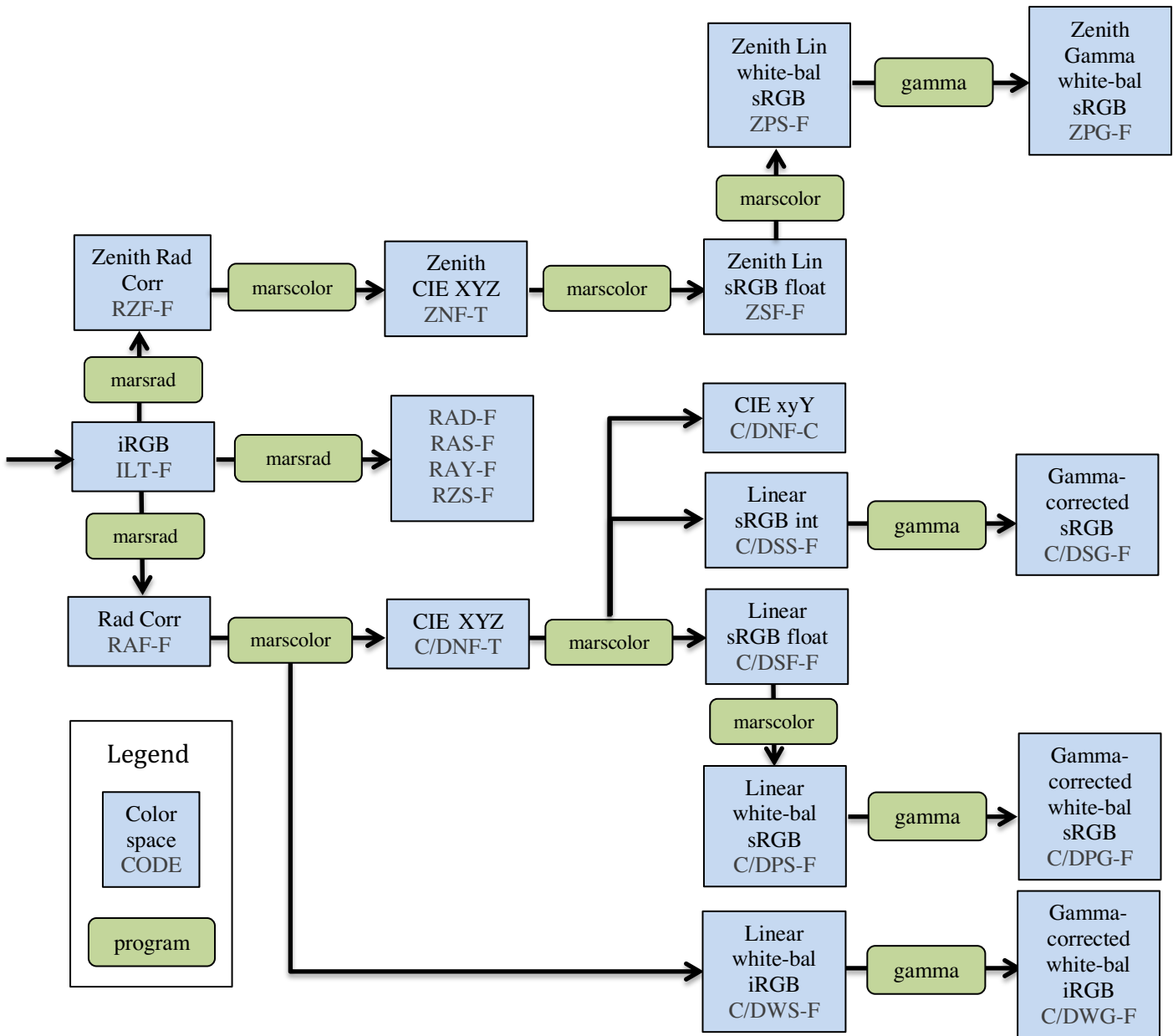


Figure 5-4: Color and Radiometric Processing Flow for InSight. The code contains the 3-letter product type and the Filter/Color code (indicating the color type)

5.2.2.4 Anaglyph/Colorglyph RDR

A stereo anaglyph is a method of displaying stereo images quickly and conveniently using conventional display technology (no special hardware) and red/blue glasses. This is done by displaying one band of the left eye of the stereo pair in the red channel, and displaying the same band from the right eye in the green and blue channels. An anaglyph data product simply captures that into a single 3-band color image, which can be displayed using any standard image display program with no knowledge that it is a stereo image. The red (first) band contains the left eye image, while the green and blue (second and third) bands each contain the right eye image (so the right image is duplicated in the file).

A closely related product, dubbed “colorglyph”, uses the red channel from the left eye, and the green and blue channels from the right eye (as opposed to the same source band in all channels), in order to create an impression of color with red/blue glasses while still retaining stereo.

Anaglyphs and colorglyphs can be created for both single-frame stereo pairs, and for mosaic products. For single-frame pairs, they generally should be linearized for proper epipolar alignment.

5.2.3 Geometry RDRs

This section describes the RDRs that are in image format (a raster of values coregistered with the original image), but whose values are not intensity of light. Rather, they represent geometry, such as XYZs or surface normals, that correspond to the given pixel.

5.2.3.1 Disparity RDR (DSR, DSP, DFF, DDD, MDS)

Stereo correlation works by matching features in one image with the same features in another image. The parallax, or spatial difference measured in pixels, between a feature location in two individual images (typically the left and right images of a stereo pair) is called the “disparity”. Disparity files contain these disparity values in both the line and sample dimension for each pixel in the reference image. This reference image is traditionally the left image of a stereo pair, but could also be the right image (and on InSight, we process both directions in all cases). The geometry of the Disparity image is the same as the geometry of the reference image. This means that for any pixel in the reference image, the disparity of the viewed point can be obtained from the same pixel location in the Disparity image.

There are three types of disparity. In the primary type (DS*), the values in the Disparity image are the 1-based *coordinates* of the corresponding point in the non-reference image. Thus for each pixel, the line/sample coordinates of the matching pixel in the other image are read directly from the disparity file. Disparity values of 0.0 indicate no valid disparity exists, for example due to lack of overlap, correlation failure, or parallax occlusion. This value is reflected in the MISSING_CONSTANT keyword. Such holes are rather common, depending on the scene.

Disparity images can also be “delta” disparity (DD*), which measures the relative offset between coordinates in the two images. This is what most imaging scientists mean by disparity. Thus you take the value of the pixel in the disparity file, and add it to the line/sample coordinates of the pixel, to get the coordinates in the other image. These delta disparity products are not produced in the nominal pipeline,

but can be produced as special products. Missing values are flagged by the value specified in `MISSING_CONSTANT`, typically 0.0.

The third type is a “first-stage” disparity (DF^*). This is a by-product of the two-stage MIPL correlation procedure (Deen & Lorre, 2005 [25]) and represents the intermediate step between stages. It contains coordinate values, as in the primary disparity. Thus, format-wise, they are the same as primary disparity (`DSR,DSP`), but they are often at a reduced resolution due to pyramiding, and are not the final results.

For each of the three primary types, several kinds of file can be produced: normal, line, sample, raw, error metric, and grid. Not all kinds apply to all types.

Normal disparity files (`DSP,DFE,DDD`) contain 2 bands of floating-point numbers in (line, sample) order using the Band Sequential format. The line and sample components may alternatively be stored in separate single-band files (`DSL,DSS,DDL,DDS,DFL,DFS`).

Raw files (`DSR`) contain the results before doing left->right and right->left reconciliation. They should be considered an intermediate result. Disparity masks (`MDS`) are used as part of the left<->right reconciliation, to mark pixels which pass or fail.

Disparity error metric files (`DSE`) contain information about the quality of the correlation match. The tools to create them are in their infancy and are not yet included in the InSight pipeline; they are listed here as a placeholder for future expansion. Thus the format has not yet been determined as of this writing.

Grid overlays (`DSG`) are an aid to visualization that may be created on occasion. These files are single-band byte images showing how a regular grid is distorted by the disparity matches (which is itself an indication of the terrain).

Note that `DSP` is the final output disparity product and should be used in almost all cases.

5.2.3.1.1 Stereo Pair Matching Method

Inherent in the designed operation of the IDC instrument are time offsets between the acquisition of left and right images intended for stereo processing. So, the `SCLK` timestamp cannot be used to automatically identify a stereo image pair as it can on other in-situ missions such as `MER`, `Phoenix`, and `MSL`.

For the InSight mission, MIPL will instead rely on the value of the `image_id` that will be included in the image header (`IDPH`). The `image_id` contains several subfields, all of which show up in the filename, as well as in the label: stereo ID, eye, mosaic ID, mesh ID, and sequence ID. A pair of images is considered a stereo pair if the stereo ID, sequence ID, and mesh ID fields, along with `Sol`, all match, in which case the `Eye` field identifies left and right. These values are used by MIPL’s pipeline for automatic generation of stereo products. Technically it is possible to use other pairs of images that are *not* marked as stereo pairs, but it requires manual processing. MIPL may do that on occasion for special requests. Appendix A shows the layout of the `image_id` field and details of the field meanings.

5.2.3.2 XYZ RDR (XYZ, XYR, XYE, DEM)

An XYZ file contains 3 bands of 32-bit floating point numbers in the Band Sequential order. Alternatively, X, Y and Z may be stored in separate single-band files.

The pixels in an XYZ image are coordinates in 3-D space of the corresponding pixel in the reference image (note that all locations are in meters unless otherwise noted). This reference image is traditionally the left image of a stereo pair, but could be the right image as well. The geometry of the XYZ image is the same as the geometry of the reference image. This means that for any pixel in the reference image the 3-D position of the viewed point can be obtained from the same pixel location in the XYZ image.

XYZ files are also sometimes called point clouds. The difference between an XYZ file and a traditional point cloud is that point clouds are typically just a list of coordinates (perhaps with intensity values attached), whereas XYZ files retain the spatial relationships between the coordinates in image space.

The coordinate system for XYZ values is important for InSight. 3-D points can be referenced to any of the InSight coordinate systems (specified by DERIVED_IMAGE_PARAMS Group in the VICAR label, and described in Section 6.3). However, by convention, the product named “XYZ” is in Site frame and the product named “XYR” is in Lander (IDA) frame (the name comes from R=Rover, and is heritage from other missions).

The distinction matters for the instrument placement products (Section 5.2.4) and orthorectified mosaics (Section 5.4.13). The operations team decided to use Lander (IDA) frame for operations, thus the “XYR” images will be used as the geometry inputs for orthorectified mosaics. However, the instrument placement tools need to look at absolute slope and elevation, independent of lander tilt, meaning XYZ values need to be in Site frame. Thus the XYZ orthorectified mosaic that is the basis of the primary instrument placement products is generated using Lander (IDA) frame geometry, while the *values* in the mosaic are Site frame XYZ values. This is indicated by “XYZ” as the product type code in the mosaic filename, and “R” as the “frame” character in the filename (Section 4.3.3).

Most XYZ images will contain "holes", or pixels for which no XYZ value exists. These are caused by many factors such as differences in overlap, correlation failures, and the failure of a result to meet quality checks in the XYZ program (this list is only representative, not exhaustive). Holes are indicated by X, Y, and Z all having the same specific value. This value is defined by the MISSING_CONSTANT keyword in the label. For the XYZ RDR, this value is (0.0,0.0,0.0), meaning that all three bands must be zero (if only one or two bands are zero, that does not indicate missing data). The use of zero as a convenient sentinel value is based on legacy software from previous missions and is compatible with current mission software. Note that “0.0,0.0,0.0” could in principle occur as a result of stereo processing, but this is unlikely for InSight; the coordinate system origin is inside the shoulder joint of the arm, which cannot be seen by the cameras. Even if the value were to be returned as actual data, losing a single pixel in the image is not problematic.

An XYZ Error metric (XYE) is available, which gives the estimated error for each pixel. It is a 3-band float product, with the three bands indicating the estimated range error (in meters) along each of the X, Y and Z axes (by convention this is only done in Site frame). These values together define the error ellipsoid. Note that these values are axis-aligned, while the error is naturally range-aligned. Therefore,

the Range Error (RNE, Section 5.2.3.3) product will generally be more accurate. XYE is provided as a convenience and its ellipsoid will always completely contain the RNE ellipsoid.

XYZ files (and individual X,Y,Z files) can be filled (XYF, XXF, YYF, ZZF) (see Section 5.2.1.5). They can have associated lander mask files (MXY) as well as becoming masked (XYM) and saved as an overlay (XYO), as shown in Figure 5-5 below. The lander mask files are discussed in more detail in Section 5.2.1.7.

Finally, DEM is a special case of ZZZ (Z-band only). ZZZ is the Z-component of the XYR product (on InSight, by convention ZZZ is measured in Lander/IDA frame). DEM is the same as ZZZ except the sign has been reversed (so positive is up). By convention, on InSight the DEM is measured in Site frame. Thus, ZZZ shows the elevation model from the lander's perspective (using the engineering +Z down coordinate system), while DEM shows the elevation model according to gravity (using the science-friendly +Z up coordinate system). The DEM type is generally used only in orthorectified mosaics.

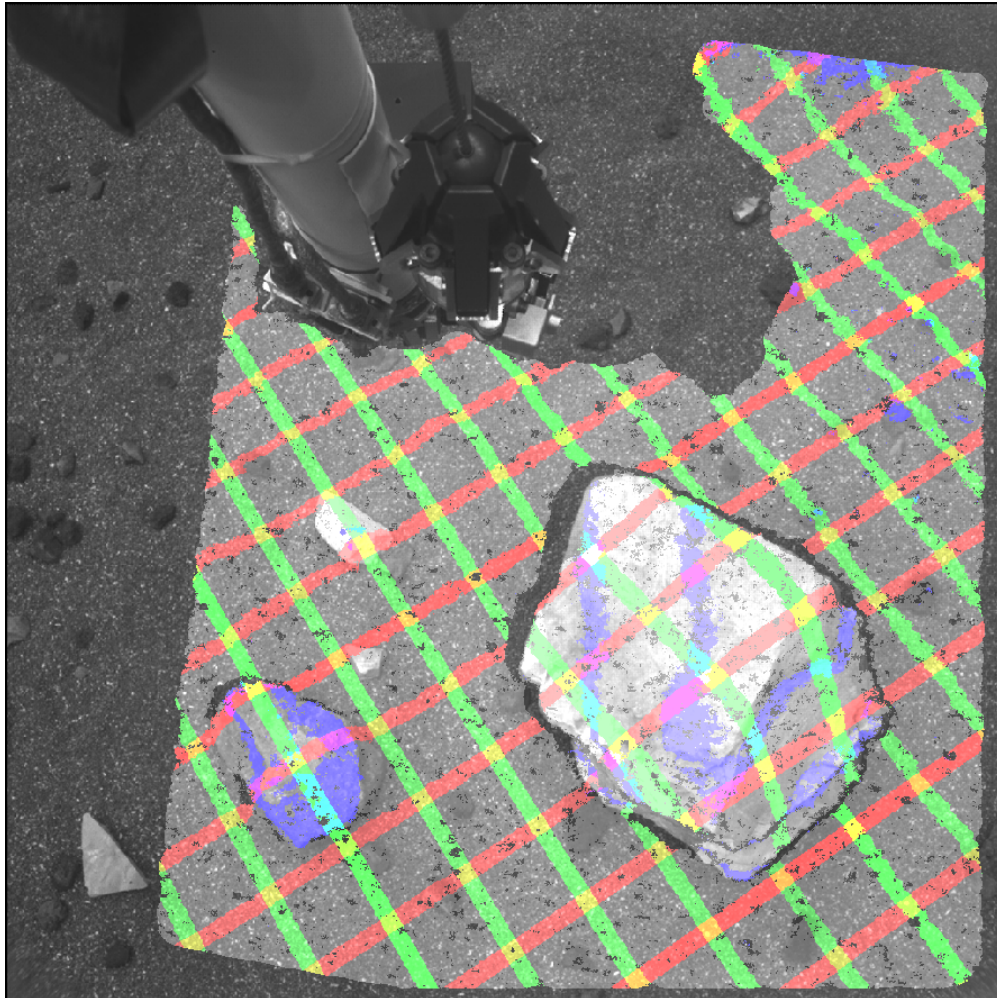


Figure 5-5: XYZ Data Masked and Overlaid onto Image EDR. Red, green, blue indicate lines of constant X, Y, Z respectively at 0.1m spacing (blue lines are thus traditional elevation contours).

5.2.3.3 Range RDR (RNG, RNR, RNE)

The pixels in a Range image represent Cartesian distances from a reference point (defined by the RANGE_ORIGIN_VECTOR keyword in the VICAR label) to the XYZ position of each pixel (see XYZ RDR). This reference point is normally the camera position as defined by the C point of the camera model for RNG, and the origin of the Lander coordinate system for RNR (the frame used is defined by the reference coordinate system labels in the DERIVED_IMAGE group). A Range image is derived from an XYZ image and shares the same pixel geometry and coordinate system. As with XYZ images, range images can contain holes, defined by MISSING_CONSTANT. For InSight, this value is 0.0. Range files are single band, 32-bit floating-point images.

The Range Error metric (RNE) gives the estimated error of the XYZ point in meters for each pixel. Like XYE, it is a 3-band float product, whose three bands define the error ellipsoid (in meters). However, for RNE the bands are interpreted differently, in a way that more naturally represents the underlying error

mechanism. The first band is the error in the downrange direction - radially away from the camera. This is the primary error for any stereo ranging system. The other two bands contain the cross-range error, orthogonal to each other and to the downrange vector direction. Band 2 (first crossrange) is coplanar with the camera stereo baseline and as such can generally be thought of as the "horizontal" crossrange direction, with band 3 the vertical, for common stereo geometries.

Range products can become masked (RNM), as well as filled (RNF) and overlaid (RNO), as shown in Figure 5-6 below.

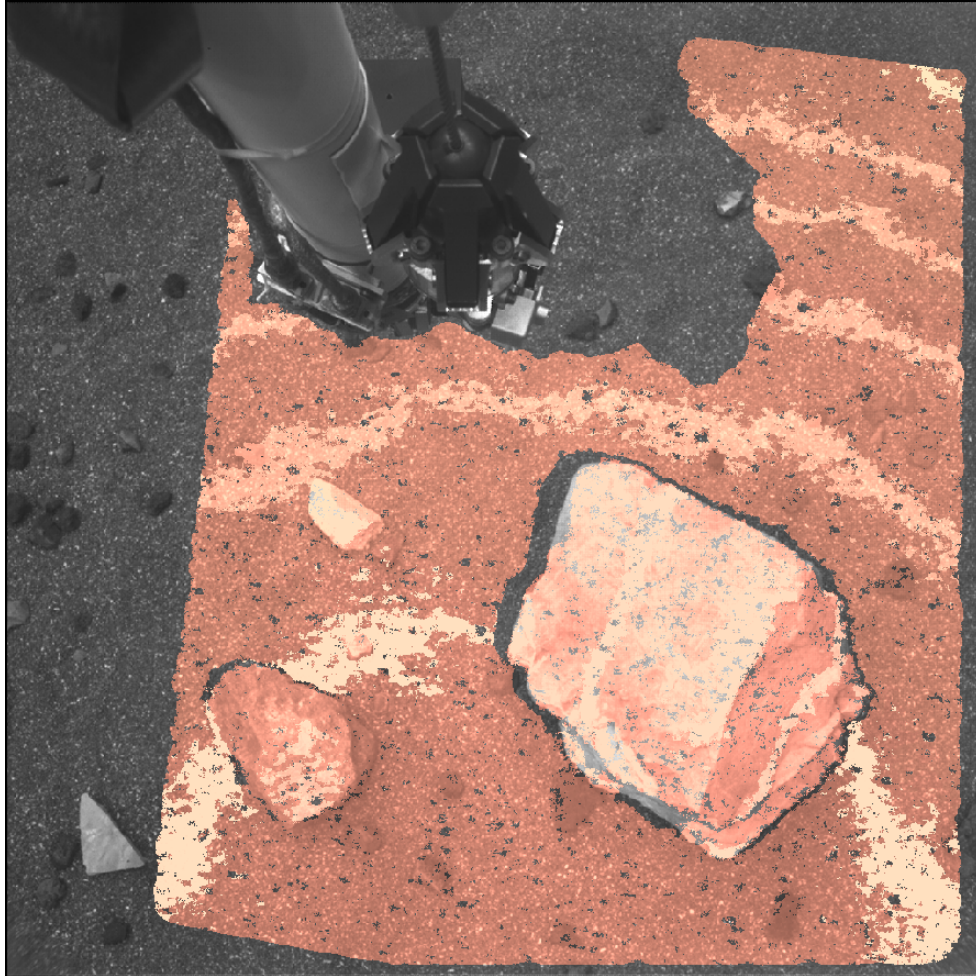


Figure 5-6: Range (Distance) data overlay on image EDR

5.2.3.4 Surface Normal RDR (UVW, UVS)

The pixels in a surface normal (UVW) image correspond to the pixels in an XYZ file, with the same image geometry. However, the pixels are interpreted as a unit vector representing the normal to the surface at the point represented by the pixel. U contains the X component of the vector, V the Y component, and W the Z component. The vector is defined to point out of the surface (e.g. upwards for a flat ground), thus W is normally negative (since +Z is down). The unit vector can be referenced to any of the Insight coordinate systems (specified by the DERIVED_IMAGE_PARAMS Group in the VICAR

label, see Section 6.3) but are normally expressed in the Lander frame for UVW and Site frame for UVS.

Most UVW images will contain "holes", or pixels for which no UVW value exists. If the corresponding XYZ point doesn't exist, neither will the UVW, but the UVW will have additional holes due to failures in the surface normal computation (such as insufficient coverage or excessive variance). Holes are indicated by U, V, and W all having the same specific value, defined by MISSING_CONSTANT as (0.0,0.0,0.0). Unlike with XYZ files, zero is not a valid surface normal value, since the magnitude of the surface normal must be 1.0.

A Surface Normal (UVW) file contains 3 bands of 32-bit floating point numbers in the Band Sequential order. Alternatively, U, V and W may be stored in separate single-band files.

Two special kinds of surface normal products are defined based on MER experience. These are not expected to be used during nominal InSight ops, but are included in case they become needed. The UVP type projects the surface normal onto an arbitrary plane in space, so the unit vectors always lie parallel to the plane. The UVT type provides the angle between the surface normal and the same arbitrary plane in a single band.

On other missions, the UVW product contains the surface normal resulting from analyzing a small patch of pixels, which is appropriate for arm work. In contrast, the UVS product contains the surface normal resulting from analyzing a much larger patch of pixels (comparable to the size of the lander), which is appropriate for slope determination for rovers. For InSight, the slope will be computed over a much smaller scale, comparable to the size of the instruments. For both UVW and UVS products, the actual patch size can be determined from the history label. For the program "marsuvw", "SEPARATION" specifies the maximum radius in meters, and "RADIUS" specifies the maximum distance in pixels (a square area, e.g. 4 would be a 9x9 window). The actual patch size is the minimum of those, which can vary depending on range.

Surface normals can be filled (UVF) and overlaid (UVO). See Figure 5-7. Separate U,V,W files can also be filled.

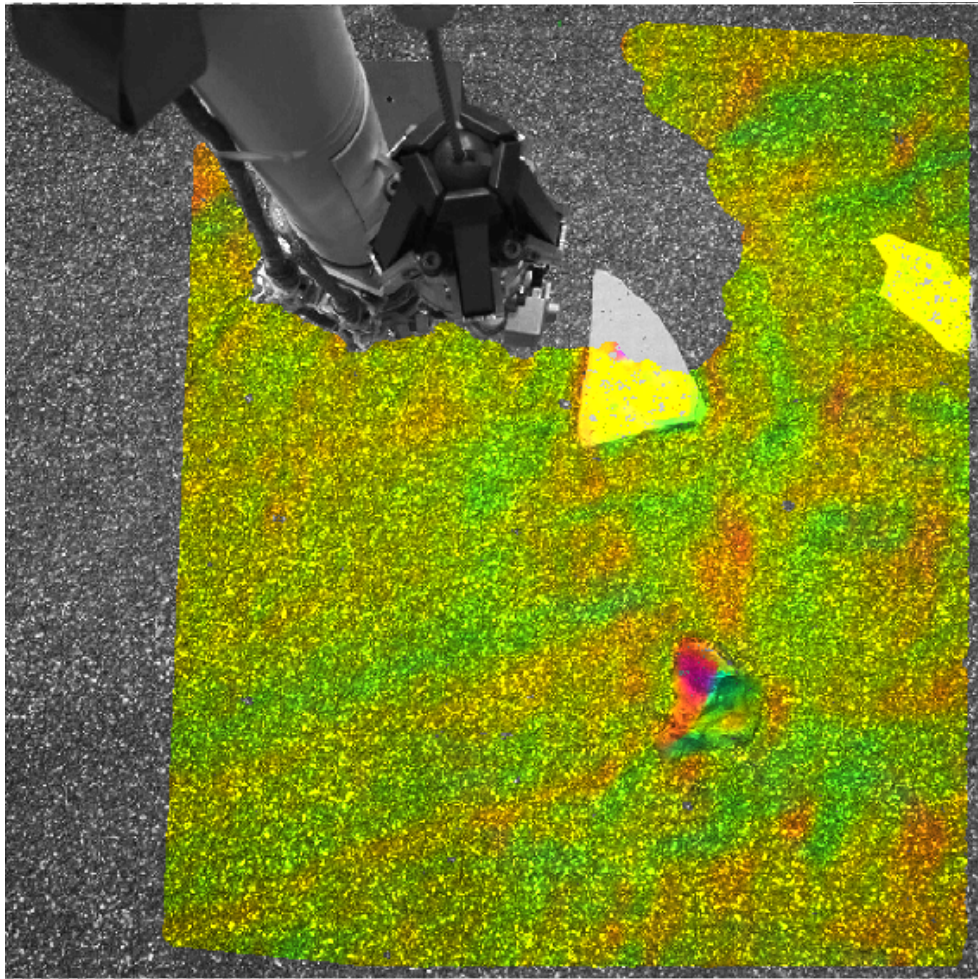


Figure 5-7: Surface Normal Data Overlaid onto Image EDR. RGB colors map directly to XYZ components of the surface normal

5.2.3.5 “Surface Roughness” RDR (RUF)

The roughness map (RUF) contains surface roughness estimates at each pixel in the image, along with a "goodness" flag indicating whether the roughness meets certain criteria. This product is not expected to be used for InSight but is being retained as a placeholder. Instrument-specific roughness products are generated instead, see Section 5.2.4.6.

5.2.3.6 Slope RDRs (SLP, SRD, SHD, SMG, SNT, SEN)

The Slope-related RDRs represent aspects of the slope of the terrain as determined by stereo imaging. The Slope Maps are derived from the UVS product. There are several slope types, each of which can additionally be overlaid on a background.

In the equations below, u , v , and w are values from the UVS file, while x , y , and z are values from the XYZ file (converted to the same coordinate system as necessary)

5.2.3.6.1 Slope (nominal) RDR (SLP)

The SLP (SLO for overlay) type contains the slope in degrees for each pixel. See Figure 5-8.

$$\text{slope} = \frac{180}{\pi} \left(\frac{\pi}{2} + \tan^{-1} \left(\frac{w}{\sqrt{u^2 + v^2}} \right) \right)$$

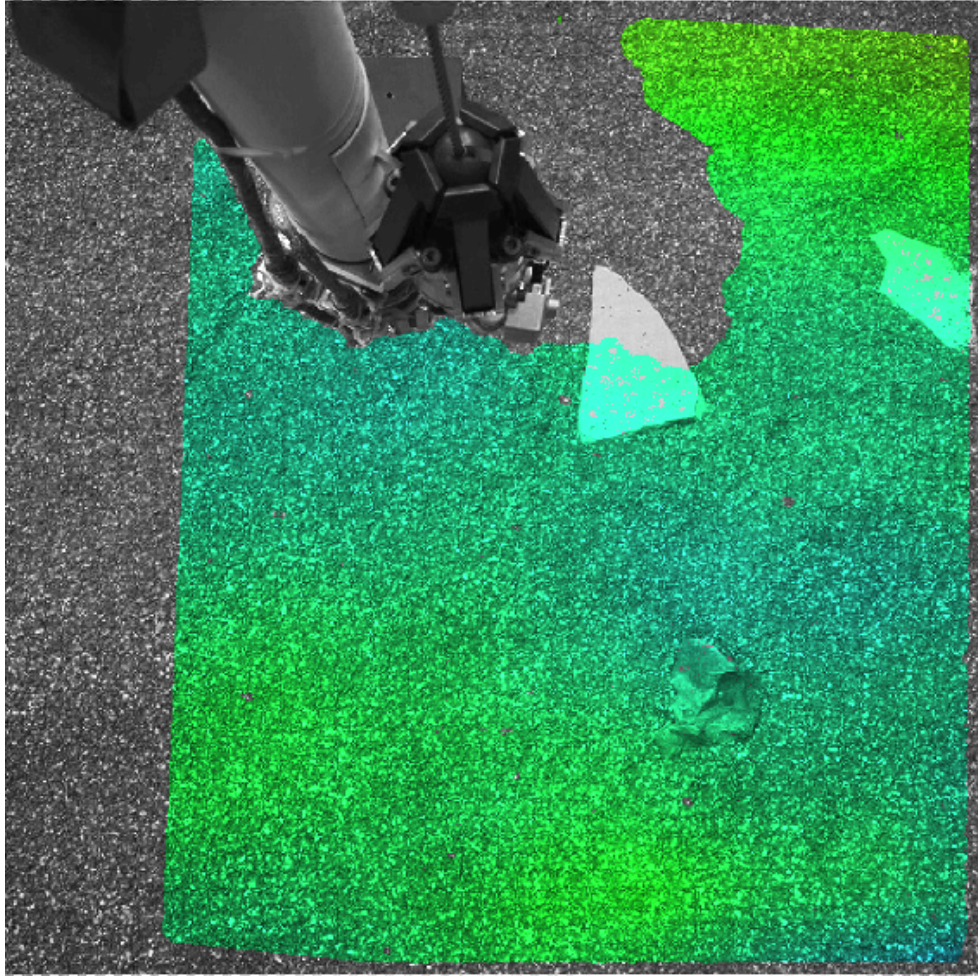


Figure 5-8: Slope Data Overlaid onto Image. Blue is less slope, green into red is more slope.

5.2.3.6.2 Slope Radial Direction (SRD)

The SRD (SRO for overlay) type contains the component of the slope (in degrees) that was facing the lander, i.e. the component of slope radially outward from the coordinate system origin. In the formula below, R is the coordinate system origin (always 0,0,0 for InSight). See Figure 5-9.

$$\mathbf{V} = \frac{[x - \mathbf{R}_x, y - \mathbf{R}_y]}{\sqrt{(x - \mathbf{R}_x)^2 + (y - \mathbf{R}_y)^2}}$$

$$\text{SRD} = -\frac{180}{\pi} \tan^{-1} \left(\frac{\mathbf{V}_x u + \mathbf{V}_y v}{-w} \right)$$

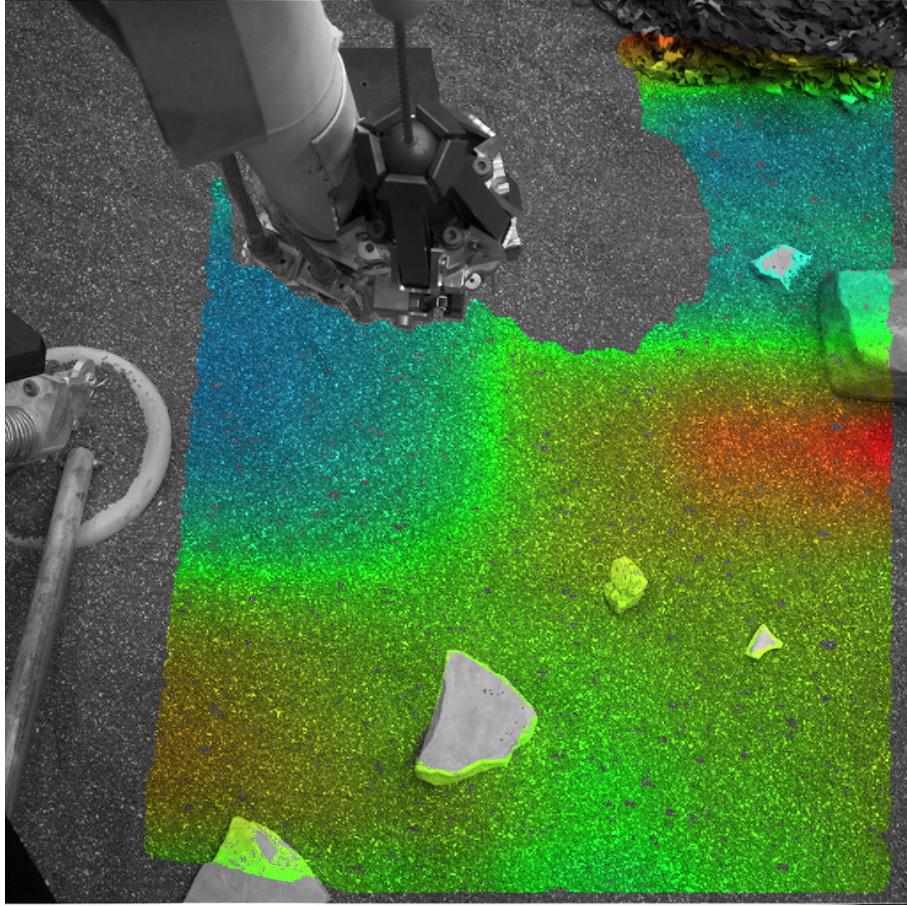


Figure 5-9: Slope Radial Direction (SRD). Data ranges from -7 degrees (blue) to +8 degrees (red)

5.2.3.6.3 Slope Heading RDR (SHD)

The SHD (SHO for overlay) type contains the direction of the slope as a clockwise angle from north, in degrees. Use the 4-quadrant form of arctangent to get a full 360-degree range. See Figure 5-10.

$$\text{slope_heading} = \frac{180}{\pi} \tan^{-1} \left(\frac{v}{u} \right)$$

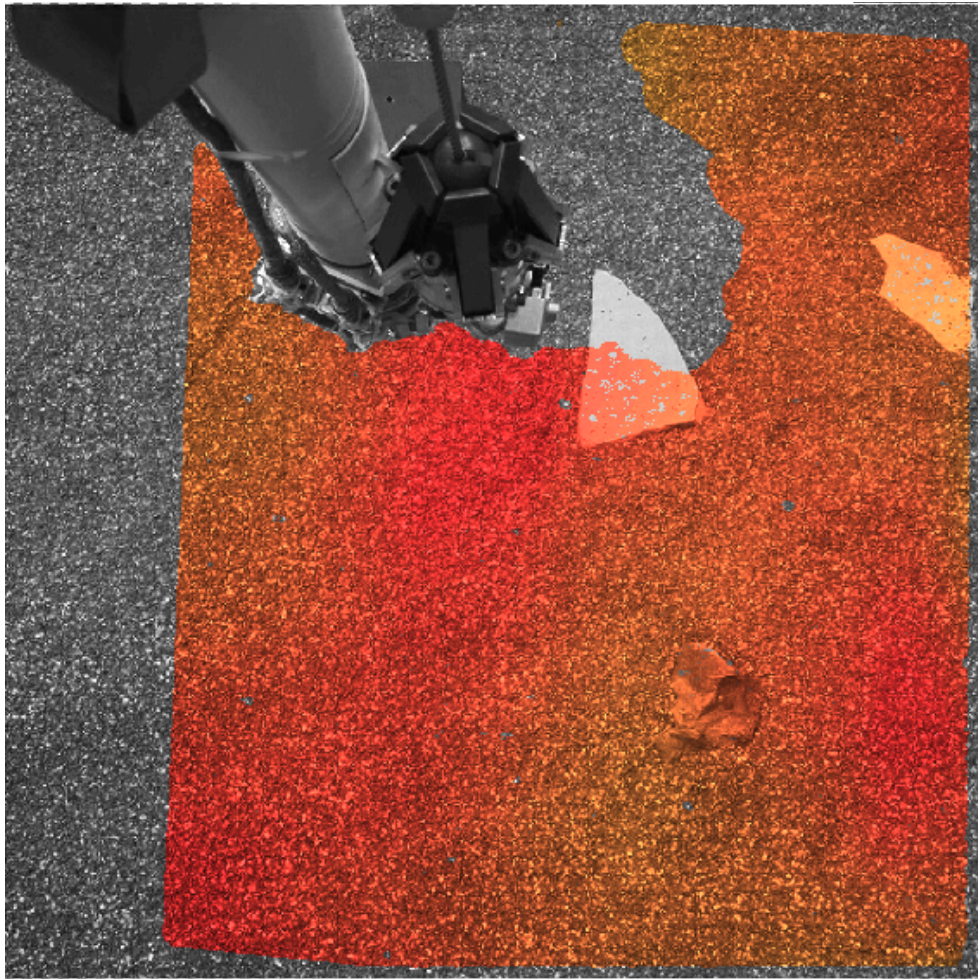


Figure 5-10: Slope Heading Overlayed onto Image EDR. Heading is arbitrarily mapped to hue.

5.2.3.6.4 Slope Magnitude RDR (SMG)

The SMG (SMO for overlay) type contains the magnitude of the normal unit vector projected onto the horizontal plane. It is directly related to $\sin(\text{slope})$.

$$\text{slope_mag} = \sqrt{u^2 + v^2}$$

5.2.3.6.5 Slope Northerly Tilt RDR (SNT)

The SNT (SNO) type contains the component of the slope in degrees that points north.

$$\text{northerly_tilt} = \frac{180}{\pi} \sin^{-1}(u)$$

5.2.3.6.6 Solar Energy RDR (SEP)

The SEP (SEO for overlay) type is included as a placeholder. It is not used for Insight.

5.2.3.7 *Photometry RDR (IEP)*

The IEP type contains incidence, emission, and phase angles for each pixel for use in photometry work. It is a 3-band float product derived from UVW in the order (incidence, emission, phase). It can also be filled (IEP). This is not generated by the MIPL pipeline but may be created in collaboration with the science team.

5.2.4 Instrument Placement RDRs

The Instrument Placement maps contain information about whether or not, and where, the instruments to be deployed by the IDA (arm) can be deployed safely and effectively. Each product has a “goodness” status band associated with it, based on the deployment requirements set by the InSight deployment and instrument teams, as well as numeric results of the analysis. The placement products assume the center of each computed pixel is where the grapple point of the instrument would be if the instrument was placed at that spot. Thus the XY location of a given point is what should be communicated to the IDA (arm) planners as the location for the grapple.

All of the instrument placement products are derivatives of the XYZ product. The geometry of the reachability maps match the reference XYZ, in that each pixel in the file directly corresponds to the pixel at the same location in the other products.

For the placement products, pixels with a DN of 0 denote no data for that pixel. To avoid ambiguity in the operational data store and archive, all placement products are generated and archived, regardless of whether any reachability values are non-0. Consequently, there are placement products comprised entirely of pixels with DN 0 - that is, with no successfully computed pixels.

The placement products help document the decision process employed by the InSight project when identifying deployment locations. For more information on how these products are used in operations, see Golombek et al (2018) [27].

Note that placement RDRs calculated from single frames are not particularly useful, as they consider only that one frame (and the WTS, in particular, barely fits in a single frame). Much more important are the placement mosaics. Unlike most mosaics, these are created by first mosaicking the XYZs (usually in an orthorectified projection), and then running the placement programs on the mosaic. Thus, they consider the entire terrain covered by the mosaic, not just one frame.

5.2.4.1 *Common Concepts*

Concepts that are common to multiple instrument placement products are discussed in this section.

5.2.4.1.1 *Instrument Placement Modeling*

The programs represent the feet as a set of circles - 3 each for SEIS and WTS, 4 for HP3. The size of these circles matches the size of the feet as they interact with the ground. The circle radii are 3cm for SEIS, 2.8cm for WTS, and 5cm for HP3.

The bodies of the instruments are also modeled by circles. This is natural for SEIS and WTS since they are circular (19.75cm and 36.15cm, respectively), but HP3 is rectangular. Thus the body model consists

of a set of 40 circles of 50cm radius each, which fill the space of the body.

5.2.4.1.2 Tether Effects

The feet are initially set assuming the tether points directly down the -X axis. However, the actual instruments while being deployed will tend to rotate so the tether points back toward the spacecraft. This is accounted for by rotating the instrument (i.e. feet position) based on the tether pivot points.

For SEIS, the tether emanates from a single tether box position. This position is hardcoded at $x=0.280$, $y=-0.1015$ (meters) in the lander frame. The tether points radially from this point to the instrument deployment location, and the SEIS feet are rotated to match.

For HP3, the algorithm is more complicated, because the tether can be routed different ways based on where the instrument is being deployed. HP3 has two "virtual" tether anchor points (in meters):

- HP3 tether guard post1: $P1 = (X1, Y1) = (0.083, -0.435)$
- HP3 mount: $P2 = (X2, Y2) = (-1.266, 0.147)$

The software determines which anchor point the tether will wrap around for the given location, and updates the rotation angle to match.

For WTS, there is no tether. However, the arm moves slowly enough that the WTS is likely to rotate with the arm, meaning that there is a most-likely rotation value that acts as if there was a tether.

Additionally, there is an offset between the WTS and SEIS, because the grapples are not coaxial in the nominal placement. The WTS position is thus offset by an amount determined in the SEIS_OFF_X and SEIS_OFF_Y parameters to the programs (this can be found in the VICAR history label). This offset is applied before any rotation is done (so SEIS_OFF_Y should generally be 0). All WTS rotations (such as for clocking) are done centered around the nominal (SEIS) location. This means that WTS clocking rotation (see below) also introduces a small translation in the WTS.

5.2.4.1.3 Instrument Clocking

Although the tether controls the rotation of the instrument while hanging on the arm to some extent, the instrument can still rotate within a certain range. This rotation is called "clocking". Zero clocking is defined as the rotation resulting from the tether calculation.

In order to assess the safety of placing an instrument at any given point, the programs analyze the instrument placement throughout this range of clocking angles. Because it is computationally intractable to compute every possible angle, we settle for checking a discrete set of clock angles. The set of angles checked are controlled by the CLOCK_RANGE and CLOCK_STEP parameters. Generally these will be set to be centered around 0 clock angle (and to include 0 clock angle), but this is not mandatory. As with the WTS offset, these values should appear in the VICAR and PDS4 history label (unless they are the default, -15 to +15 degrees at 3 degree steps).

Note that the bodies of the SEIS and WTS are simple circles and are rotationally symmetric. Thus, clocking is not done for them (but it is for their feet). The WTS is assumed to rotate around the position

after taking into account the SEIS_OFF_X/Y parameters, which again makes it rotationally symmetric.

For the HP3 body, the density of circles representing the body is such that only a few clock angles need to be checked in most cases. The default is to check just three - both limits of clock_range, and the center. This is sufficient for the standard +/- 15 degree clock range.

5.2.4.1.4 Instrument Footpad Settling

The Z location of each foot is defined as the highest point found within the foot circle, after removing outliers. The theory behind this is that the foot will be perched on the highest point in the area. However, if the ground is sandy, the feet are likely to "settle" into the sand by some amount. In order to account for this, the programs "settle" the feet by a specified amount. This amount is added to each foot location (because +Z is down, adding the value lowers the foot location). Analysis is then done using all combinations of settled and unsettled feet. This should pick up the worst-case scenario on both ends of the spectrum - no settling and maximum settling.

5.2.4.1.5 Holes in the Data

Holes or gaps (missing data, typically from correlation failures) in the XYZ image are generally ignored. The central position of each circle has to have a valid XYZ location, so a hole in the middle of a foot, if big enough, could cause the location to be rejected. But no other analysis is done of holes in the area the foot "should" cover. Holes represent unknowns, so normally the presence of holes would be a red flag for placement. However, the feet are generally small enough that significant unknown terrain within one foot patch is unlikely. Also, we look for the maximum within the foot patch, so holes would have to also be the highest terrain in the patch to cause a problem. However, it does open the possibility of single pixels representing an entire foot patch.

For body roughness, holes are more of a concern, because of the much larger area covered by the body. Analysts must make sure there is enough coverage around the instrument to be believable.

5.2.4.1.6 Grapple Deflection

If the lander is tilted, the grapple will be deflected so it is no longer aligned with the lander. This (rather small) effect is not accounted for in any of the instrument placement products, which are intended to screen locations. The instrument placement team takes this effect into account when using the RSVP program for detailed analysis of the chosen locations.

5.2.4.1.7 WTS Offset

The WTS is nominally supposed to be deployed 5cm closer to the lander than the SEIS. In order to be able to bookkeep a single SEIS/WTS location, this offset is incorporated into the placement programs. This means that the WTS placement products are reported as if they were deployed concentrically, but they are calculated as if the WTS was placed 5cm closer (along the vector to the Lander coordinate system origin). The 5cm value is a parameter; if the value is changed it will appear in the history labels. The bottom line is that the SEIS and WTS placements can be evaluated as if they were concentric, even though there is an offset applied during deployment.

5.2.4.2 Workspace Reachability RDR (WSH, WSS)

The Workspace reachability product (WSS, WSH) is a mask product showing the boundaries of the workspace where the InSight instruments can be deployed. The boundaries are computed by comparing the XYZ coordinates with precomputed polygons indicating the edges of the workspace. It is important to note that workspace reachability is purely a function of XY coordinate – the actual terrain (and thus the Z coordinate) does not play a role. Specifically, if the lander is tilted, the grapple may hang at a different angle relative to the lander, which can either increase or decrease the actual instrument range. This effect is not taken into account in these products.

Workspace reachability products are single-band BYTE images, containing a code 0-7 indicating which polygons the point is inside. See Table 5-2 for values.

The files are different based on instrument: WSH contains the HP3 workspace and WSS contains the combined SEIS/WTS workspace. See Figure 5-11 for an example.

Table 5-2: Workspace Reachability Values

Value	Deployable Instrument		Marsviewer Color
	HP3 (WSH)	SEIS/WTS (WSS)	
0	no data	no data	Black
1	outside HP3 boundary, or no solution	outside all SEIS/WTS areas, or no solution	Red
2	HP3 footplane (footprint) boundary	WTS footplane (footprint) boundary	Blue
3	n/a	SEIS footplane (footprint) boundary	Purple
4	IDA grapple workspace --- IDA grapple kinematically reachable zone for HP3 deployment	IDA grapple workspace --- IDA grapple kinematically reachable zone for SEIS/WTS deployment	Orange
5	nominal (IDA grapple workspace) zone for HP3 deployment	nominal (IDA grapple workspace) zone for SEIS/WTS deployment	Green
6	inside HP3 grapple point boundary, but ICC view partially occluded	inside SEIS/WTS grapple point boundary, but ICC view partially occluded	Yellow
7	inside HP3 grapple point boundary, but more maneuvering required for collision prevention	inside SEIS/WTS grapple point boundary, but more maneuvering required for collision prevention	Magenta (“Pink”)

The assigned numbers use 5 to indicate the best workspace, in order to maintain consistency across all InSight goodness masks. 6 and 7 are within the workspace, but are less desirable due to occlusions or arm constraints. For the purposes of the overall goodness product (Section 5.2.4.7), 6 and 7 are considered good. As WSH only represents one instrument, a value of 3 is not used for that product.

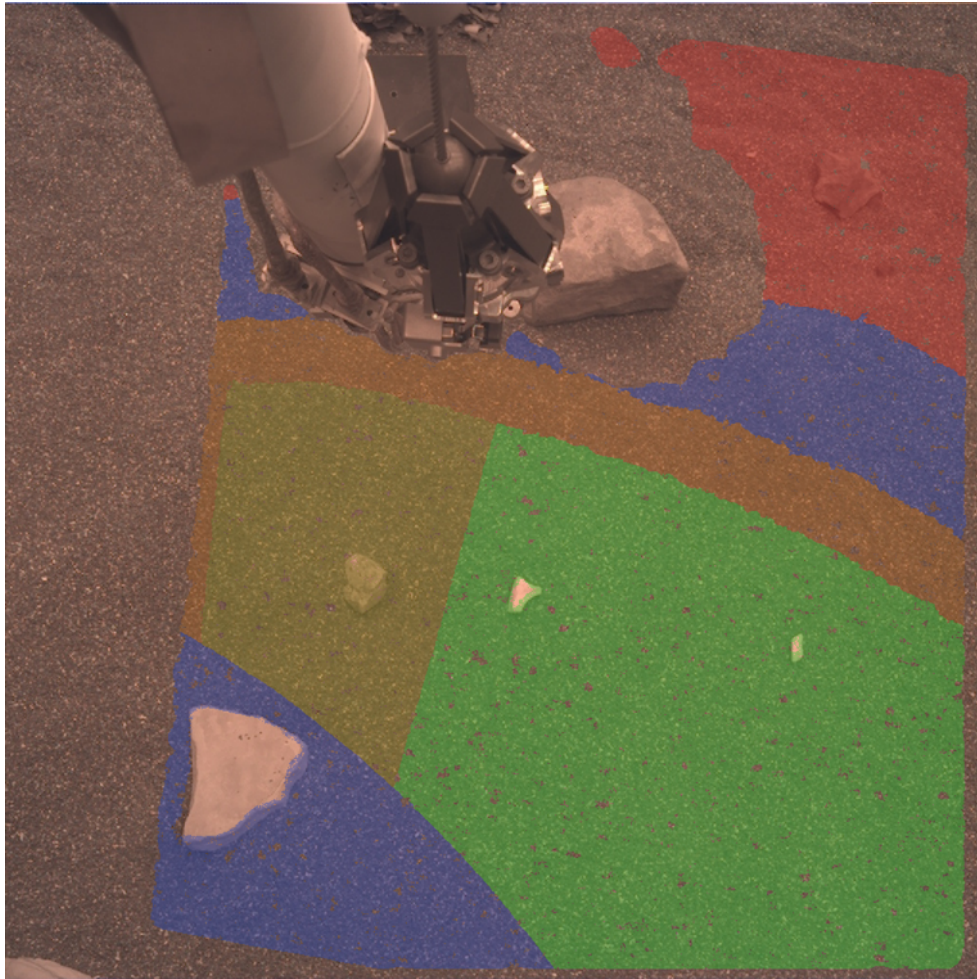


Figure 5-11: Workspace Reachability for HP3 (WSH) Data Overlaid onto Image EDR. Green indicates nominal zone for HP3 deployment (5), yellow is ICC partially occluded (6), orange is the kinematically reachable zone (4), blue is HP3 footprint boundary (2), and red is outside the HP3 boundary (1). Numbers refer to left column of Table 5-2.

5.2.4.3 Instrument Tilt RDR (TFH, TFS, TFW)

The instrument tilt (TF*) products show the tilt an InSight instrument would have if it were placed at the given pixel. Tilts are defined as the angle relative to vertical (the gravity vector), without regard for the direction of tilt. Tilt files are 3-band, 32-bit floating point images, with tilt values expressed in degrees. The first band is a status (flag or goodness) band indicating whether or not the result is within limits. As with all InSight placement products, 5 represents a good result (see Table 5-4 for the rest of the values). The second and third bands are the minimum and maximum tilts, respectively. The tilt products all take into account all clock angle rotation combinations, foot sinkage, and placement margin of the instruments at each pixel, returning the worst-case tilt (highest value) found. Table 5-3 shows the contents of each band.

Note that minimum tilt is only the minimum tilt actually found during the analysis, which is done at

quantized clock angles rather than continuously throughout the range. If all tilts are in roughly the same direction, minimum tilt reflects the actual minimum, as would be expected. However, if the tilts are in different directions, it is highly likely that 0 would be a possible tilt somewhere within the range and the minimum should thus be 0. But the program does not look for this case, instead returning the minimum of the actual computed tilts.

For SEIS and WTS, tilt is computed by taking the Z value of each foot, constructing vectors between them, and using the cross-product to determine the normal of the plane defined by those points. As noted above, this is done for each combination of settled and unsettled feet, and the min and max tilts across all combinations are gathered.

Since HP3 has 4 feet, the 4 foot points are unlikely to lie in a single plane; it is possible for the HP3 to sit unstably on any of its 3 feet. For that reason, the above computation is performed for all combinations of 3 of the 4 feet. Thus the best (min) and worst (max) cases are determined no matter which feet are actually stably on the ground. This means that each combination of 3 feet is also analyzed for each combination of settled and unsettled feet. In reality, the compliance of the ground is likely to allow all 4 feet to sit on the ground, but this algorithm ensures we look at the worst case.

An example is shown in Figure 5-12.

Table 5-3: Band Assignments for Tilt Products

Band	Value
1	State, or Goodness flag
2	Minimum Tilt
3	Maximum Tilt

Table 5-4: Instrument Tilt State Assignments

State Value	Meaning
0	No value
1	n/a
2	n/a
3	Exceeds tilt threshold
4	n/a
5	Within all limits (good)

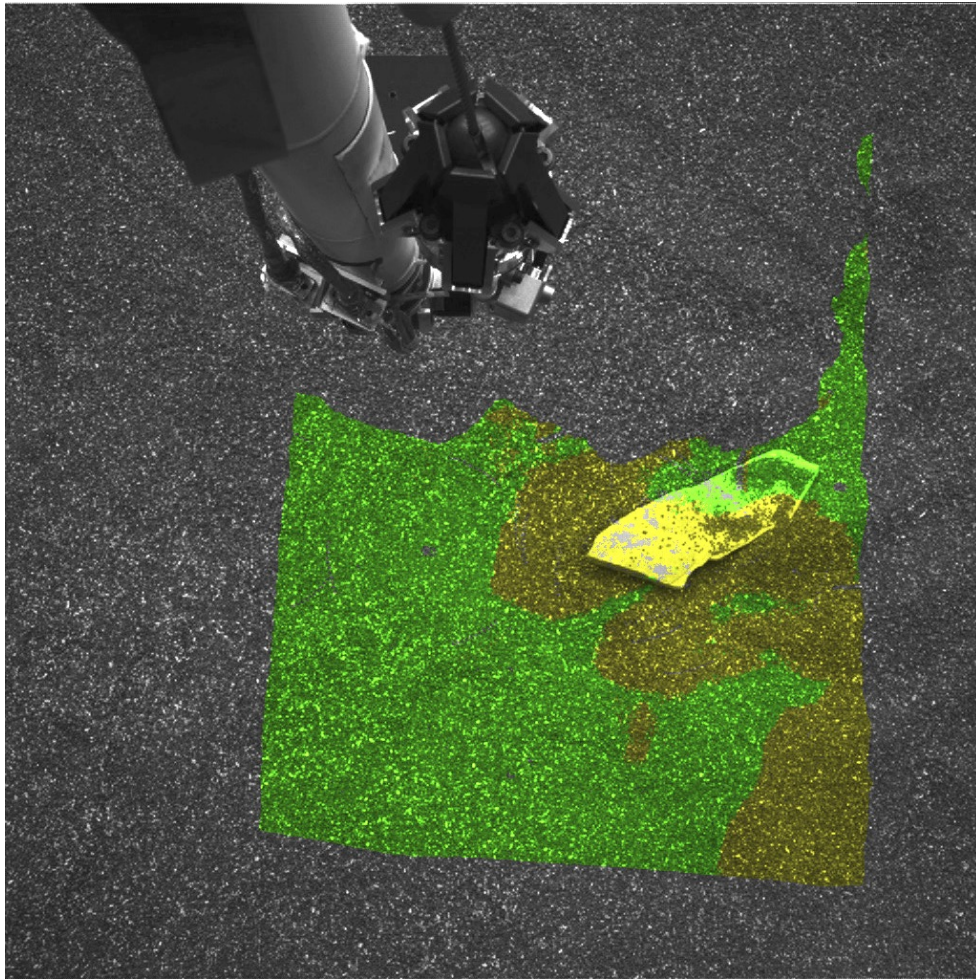


Figure 5-12: Instrument Tilt for SEIS (TFS) Data Overlaid onto Image EDR. Green indicates good (state=5).

5.2.4.4 *Delta Tilt RDR (TDS)*

The delta-tilt is a comparison between the tilt of the SEIS and the tilt of the WTS that covers it. See Figure 5-13 for an example. This is important for instrument deployment to avoid SEIS and WTS from ever contacting each other. It is computed by first computing tilt as described in Section 5.2.4.3 for the SEIS. At each SEIS clock point, the tilt of the WTS is computed. This also follows the above tilt algorithm, except the WTS is clocked independently.

At each combination of SEIS and WTS tilts, the delta tilt is computed, as the arc cosine of the dot product between the two normal vectors. The min and max delta tilt is then computed across all combinations of clocks for both instruments. Just like tilt, the file is a 3-band float product; the first band is the state band (Table 5-4), the second is the minimum delta-tilt value, and the third is the maximum (see Table 5-3).

It is important to note that this is not finding the min/max tilt of each instrument and then computing the

angle between them. It is computing the angle between each and every combination of clock angles (subject to the step size and range parameters), and determining the min/max of those.

Note that delta-tilt was removed as a component of the overall goodness map right before landing, due to analysis showing that it was not actually needed to meet instrument placement requirements. The product itself is still generated, however.



Figure 5-13: Instrument Delta Tilt for SEIS/WTS (TDS) Data Overlaid onto Image EDR. Green indicates good (state=5).

5.2.4.5 Instrument Normal and Instrument Location RDRs (UIH, UIS, UIW, UPH, UPS, UPW, ZIH, ZIS, ZIW, XPH, XPS, XPW)

Four additional products are generated as a by-product of tilt generation: Instrument Normal and Instrument Position, each of which are expressed in both Site and Lander frames. The Site frame ones are needed as inputs by the roughness algorithm (Section 5.2.4.6), but it is more efficient to compute them as part of tilt. The Lander frame ones are generated for the convenience of the deployment team.

Note that none of these products directly relate to deployment requirements. Thus there are no thresholds, no status band, and they are not included in the goodness maps (Section 5.2.4.7).

The Instrument Normal in Site frame (UI*) files are 3-band float unit-vector normal files in the same format as the UVW product (Section 5.2.3.4). However, the normal is calculated in a special way. It is the normal not of the surface, but of the instrument itself, when located at the given pixel. It is measured in Site frame.

Examination of the tilt algorithm shows that the instrument normal is not unique - the combinations of clock angles, sinkage, and multiple feet lead to a variety of possible surface normals. However, the roughness computation needs one "most likely" normal. Therefore, the one closest to the center of the clock range, with no sinkage, is chosen. It is not feasible to save the normal for every clock angle, and the difference should be negligible in most cases for any terrain that is suitable for instrument deployment. The program analyzes all combinations of 3 feet, and takes the average normal across those combinations (applies only to HP3, since SEIS and WTS have only 3 feet). Note that no sinkage and full sinkage result in the same normal, since all feet are assumed to sink the same amount.

The Instrument Normal in Lander frame (UP*) files are similar to UI*, except they are measured in Lander frame. They express the same vector, just measured in a different frame. UP* files are typically made only for mosaics, not for single stereo pairs.

The Instrument Z in Site frame (ZI*) files are 1-band float files in the same format as the ZZZ product (Section 5.2.3.2). They contain the Z position of the instrument if it was placed at that location, as measured in Site frame. The Z position is the coordinate of the point directly below the grapple, in the plane of the feet ("below" according to the instrument, thus this is a specific and constant point with respect to the instrument, regardless of instrument tilt – it is the point used for all instrument computations). As with UI*, this takes the center of the clock range and does not settle the feet.

Because the surface normal is an average, using it with the foot locations of the HP3 creates more than one solution. In all cases the foot position is used to fix the Z value of a plane that is normal to the instrument normal, and then the Z position of the center of the instrument (grapple point) is computed. The final Z value is the lowest (highest Z since +Z is down) of these computed positions - again, modeling the worst case (since lower means more interaction with the terrain).

The Instrument Position in Lander frame (XP*) files are similar to ZI* except they contain the entire XYZ coordinate of the point below the grapple, instead of just the Z component. The value is also expressed in Lander frame, rather than Site. XP* files are typically made only for mosaics, not for single stereo pairs.

5.2.4.6 Instrument Roughness RDR (RUH, RUS, RUW, RUK)

The instrument roughness products (RU*) contain the surface roughness under the specified instrument's (i.e. SEIS, HP3, WTS) body and feet, see example product Figure 5-14. The roughness of the body (footplane) and feet (footpatch) are computed, using slightly different algorithms.

The resulting roughness products are 3-band files of type REAL, containing a state (goodness) band indicating whether the data met roughness criteria, as well as the actual body and feet roughness values in meters (see Table 5-5 for band assignments and Table 5-6 for state band values).

At each pixel in the output file, the program uses the UI* and ZI* files to “place” the instrument at that point, which means defining a plane for the feet and body. The terrain under that plane is then analyzed, using one of two roughness algorithms described below. The results are reported in the output file, along with a “status” band, which indicates whether or not the point meets the roughness thresholds.

Roughness is computed by analyzing each of the circles comprising the feet or body models separately. The largest roughness value from any of the circles is reported as the final value. If any of the circles happen to hit a hole in the XYZ data, then the entire roughness calculation fails.

Feet (footpatch) roughness is simply the maximum extent of any points (after outlier rejection) above or below the instrument plane. In other words, the maximum value above the plane, minus the minimum value below the plane. This is computed along the instrument normal. This means that feet roughness considers both hills and valleys, which is appropriate when considering foot placement (the foot can sit on top of a hill, but you also don't want it spanning too large a valley, since the foot could slip into it). Note that the Instrument Z value is not relevant for feet roughness; the max - min spread is the same regardless of where the plane is in the Z direction.

For the instrument body (footplane) roughness, we are concerned about clearance for the belly of the instrument. Thus valleys are not relevant, as the instrument can clear them, but hills can be hazardous. Therefore for the body algorithm, we look only at excursions *above* the plane defined by the instrument feet. The roughness is defined as the maximum distance (after outlier removal) of any point above this plane. The Instrument Z value is thus critical for this calculation, as it defines where 0 is.

Settling also comes into play here, as settling will depress all of the feet and thus lower the plane. In principle the feet can sink unequally, but the worst-case scenario is if they all sink the same (this maximizes the “hill” roughness), so that is all that is considered. A foot settling value is essentially equivalent to just adding a constant to all the body roughness values.

Note that all other things being equal and with Z at the mean value, body roughness will be half as big as feet roughness, since only positive excursions are considered for the body rather than both positive and negative.

An additional product, Roughness for Skirt (RUK) has not yet been defined as of this writing. It is being considered by the operations team for development. It is included here as a placeholder; if it is created it will be included in the archive and the definition will be included either in an updated version of this document, or in a PDS errata.

Table 5-5: Band Assignments for Roughness Products

Band	Value
1	State, or Goodness flag
2	Body (footplane) Roughness
3	Feet (footpatch) Roughness

Table 5-6: Instrument Roughness State Assignments

State Value	Meaning
0	No value
1	Exceeds both body and feet limits
2	Exceeds body limit only
3	Exceeds feet limit only
4	n/a
5	Within all limits (good)

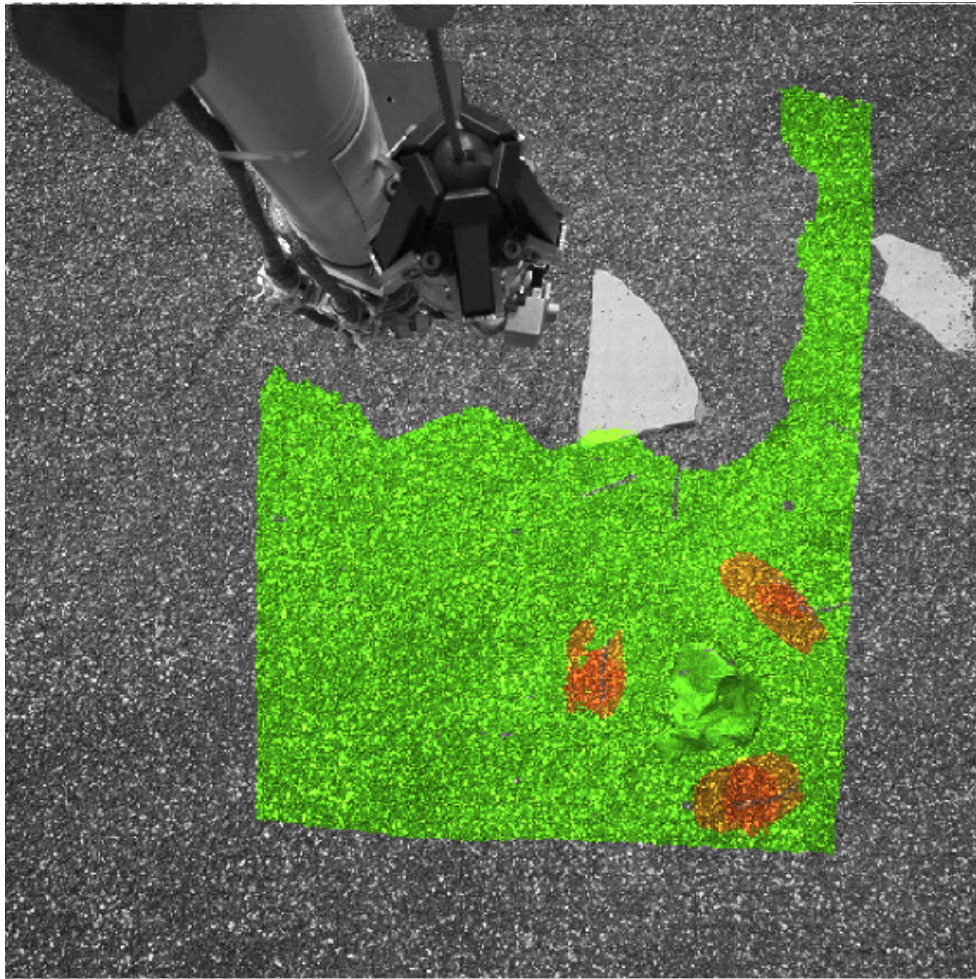


Figure 5-14: Instrument Roughness for SEIS (RUS) Data Overlayed onto Image EDR. Green indicates good (state=5).

5.2.4.7 Instrument Placement Goodness RDR (GUH, GUS)

The Goodness Product (GU*) is an all-encompassing goodness rating product that summarizes the status band for a list of input products. For InSight, this means reading in the status band for from each of the placement product RDRs (i.e. tilt, workspace reachability, and roughness), and giving an overall rating based on individual results of each. This product is intended as the first “go to” product to assess overall terrain accessibility for deployment, before delving into details.

There are two versions of this product for InSight, HP3 (GUH) and a combined SEIS/WTS (GUS) product. These products are single-band byte images. See Table 5-7 for the meaning of the goodness values, and Figure 5-15 for an example product.

Note that Delta-Tilt was removed as a component of the GUS right before landing; it was formerly value 4 in Table 5-7.

Table 5-7: Placement Goodness Values

Value	Deployable Instrument	
	HP3 (GUH)	SEIS/WTS (GUS)
0	No data, or outside mask	No data, or outside mask
1	Multiple inputs are bad	Multiple inputs are bad
2	Roughness is bad	Roughness of either SEIS or WTS is bad
3	Tilt is bad	Tilt of either SEIS or WTS is bad
4	n/a	n/a
5	All good	All good

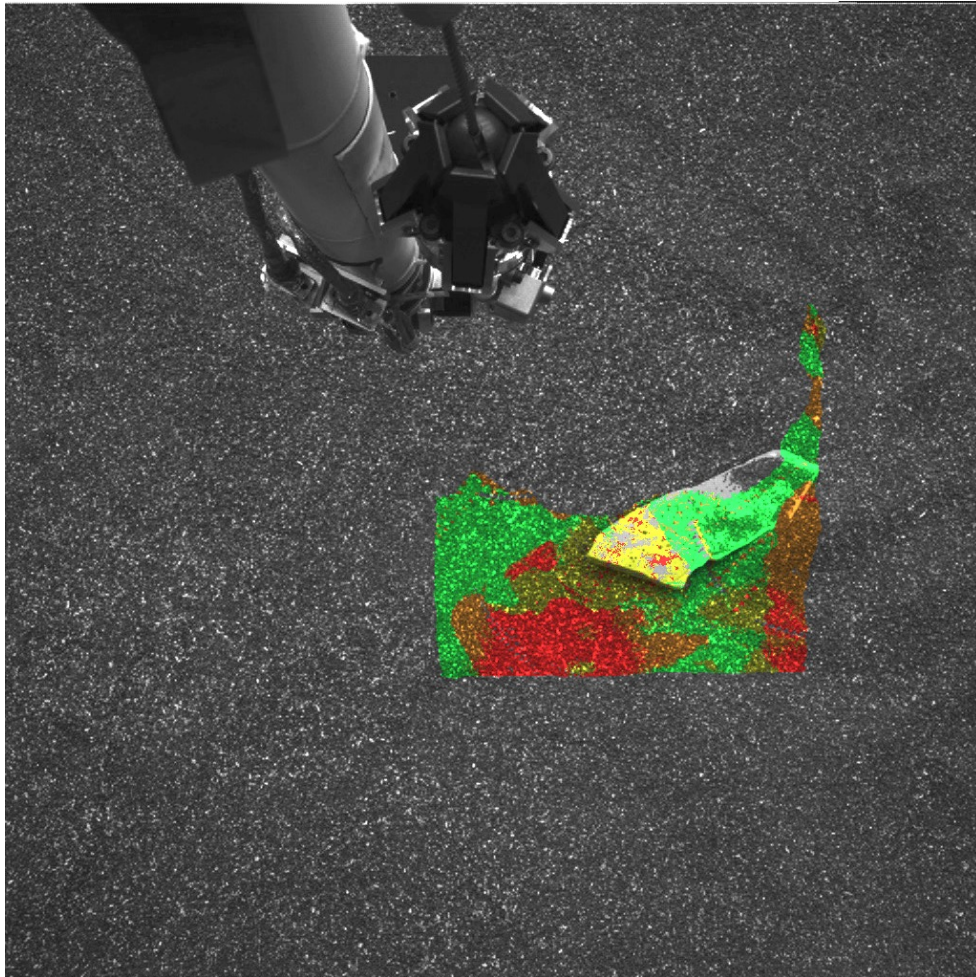


Figure 5-15: Goodness for HP3 (GUH) Data Overlayed onto Image EDR. Green indicates good (state=5).

5.2.4.8 Arm Reachability RDRs (ARM)

The Arm Reachability maps contain information about whether or not the instruments on the Arm (such as the camera or the scoop) can "reach" (contact or image) the object or location represented by each pixel in the scene. They are derived from the XYZ and Surface Normal (UVW) products.

Arm reachability products are not produced for InSight but are retained as a placeholder.

Note that reachability maps may include masks (MAR), be masked (ARK), and be overlayed (ARO).

5.3 Terrain Model (Mesh) Data Product Specification

Terrain models, also called Meshes, are high level products containing geometric description of the surface consisting of triangles, with image data mapped as a texture to each triangle. These meshes

provide a convenient way for visualization programs to view the surface in 3D, and to view it from any angle. Meshes are particularly suited to incorporating other geometry, such as a CAD-derived articulated model of the spacecraft and science targets, in the same visualization and simulation environment.

Meshes are generated by triangulating point clouds (from XYZ images, and a coregistered “skin” image) using volume-based surface extraction. The original image is used as a texture map to add detail and color to the polygonal surface representation. Any version of the image, including RDR’s or overlaid RDR’s (such as a grid derived from XYZ) may be used as the mesh skin. The operational pipeline will use the white balanced gamma-corrected (CWG) files for the default mesh skin. However, any other type may be generated, used, and be archived.

Meshes used in operations are stored in Open Inventor binary format, along with ancillary files, for use by the RSVP program [28]. This format is rarely used in the 3-D visualization and modeling world, and documentation on it is sparse (and far beyond the scope of this document to explain). This makes the format unsuitable for the PDS archive.

Meshes included with the PDS archive are delivered in the commonly-used OBJ format (originally developed by Wavefront, now an open format) [29]. Files for both formats are discussed in this section.

5.3.1 Per-XYZ Terrain RDRs, Operations format

The files described in this section are created for operations, and are not delivered to PDS.

For every XYZ RDR created, the following terrain products are generated, which follow the Single-frame RDR filename convention (see Section 4.3.1). These files are stored alongside the image files in the SPeX file store.

- a) `*.tar` - A standard Unix tar file containing a collection of tiles representing spatial subdivision of a point cloud. Each tile is a separate file within the tar-file. Each tile contains vertices that define the terrain in multiple Levels of Detail (LOD). From the vertices, triangles are striped for rendering efficiency. Note that tar-files are not used for unified mesh creation and though self-contained, serve only as intermediate products to produce a single optimized Open Inventor file. The tiles within the tar file are in the ASCII version of Inventor format.
- b) `*.iv` - Open Inventor terrain representation of a point cloud defined in the XYZ RDR. It is a single file generated by combining all tiles contained in the tar-file described above, making sure that texture images shared by multiple tiles are loaded only once by using references, and storing them as an Open Inventor binary file. It constitutes the per-XYZ mesh product.
- c) `*.rgb` - Image file in 8-bit per channel SGI RGB format that is used as the Texture Map for the per-XYZ mesh product.
- d) `*.ht` - Height Map (digital elevation model) derived from the XYZ RDR, stored as an image file. It is in VICAR format, with 3 bands. Band 1 contains actual height data. Band 2 fills areas for which there is no actual data using interpolation. Band 3 provides a metric of how close a pixel value is to the actual data.

A subset of these files is also stored under the unified terrain mesh, see next section.

5.3.2 “Unified” Terrain RDRs, Operations format

The files described in this section are created for operations, and are not delivered to PDS.

Just as individual images can be combined into image mosaics, per-XYZ meshes can be combined into unified terrain meshes. These are the ultimate terrain products used by arm planners during tactical operations.

The terrain products listed below follow the unified mesh product filename convention (see Section 4.3.4), which differs from the Single-frame filenames carried by the per-XYZ terrain products. The files are stored in the mesh directory of the SPeX.

- a) * .iv - Open Inventor file in ASCII format that contains references to all individual binary per-XYZ Open Inventor files in the “wedge” subdirectory.
- b) * .mod - ASCII file that contains references to all corresponding individual per-XYZ “.ht” Height Maps files in the “wedge” subdirectory as well as XY extent and scale of each per-wedge Height Map.

There is a “wedge” subdirectory under “mesh” on the SPeX which contains copies of the “.iv”, “.ht”, and “.rgb” files described in the previous section. The copy avoids having to provide a complex relative pathname in the unified mesh files.

5.3.3 Per-XYZ Terrain RDRs, PDS format

The files described in this section are included in the PDS delivery.

For every XYZ RDR created, the following terrain products are generated, which follow the Single-frame RDR filename convention (see Section 4.3.1). These files are stored alongside the image files in the SPeX file store.

- a) * .obj – The mesh in OBJ format. The OBJ format was originally developed by Wavefront but is now an open standard. OBJ files are ASCII-only files that are described briefly below.
- b) * .mtl – The “materials” file for OBJ. This is a text file, named by the “mtllib” field in the OBJ file, which contains the filename of the image textures or “skins” (“.png” files) used by this mesh.
- c) * .png – The “skin” file(s), or texture files, needed by the mesh, in Portable Network Graphics (PNG) format.

The subset of OBJ used by InSight is described briefly below. For a full treatment, see [29].

The OBJ file is made up of a collection of lines, where each line has a type code followed by data specific to that type code. Within each line, items are tab separated (except for the type code, which is followed by a space-tab to work around a bug that existed in Meshlab at the time this was developed).

- a) `mtllib` – Specifies the name of the “.mtl” file to use.
- b) `usemtl` – Specifies the actual material to use within the “.mtl” file (in other words, which PNG texture file to use).
- c) `v` – List of geometric vertices. From the OBJ format standpoint, these coordinates are arbitrary and unitless, but for InSight they are expressed in meters (using the coordinate system specified in the filename). Each line has “x y z” coordinates. Vertices are the corners of the triangles that make up the mesh.
- d) `vt` – List of vertex texture coordinates. These refer to the sample and line coordinates (in that order) of a vertex in the texture (skin) file. Coordinates are normalized to the range [0...1].
- e) `f` – List of faces (triangles) for the mesh. Each vertex of the triangle is listed in the form “a/b/” where a is the index into the vertex array (“v” list, start counting at 1) and b is the index into the texture array (“vt” list, start counting at 1). OBJ format enables the normal for each vertex to be specified as well, but that is not used for InSight.

An example OBJ file (elided because it is very long) is shown below:

```
mtllib D053L8127T596979590RAS_F0101_0060M1.MTL
usemtl D053L8127T596979590RAS_F0101_0060M1
v -189.179840 38.377743 -13.768595
v -189.182159 38.380280 -13.766765
v -189.182007 38.379631 -13.767196
...
v -189.053146 38.387295 -13.771981
vt 0.227 0.021
vt 0.227 0.020
...
vt 0.934 0.976
f 303348/1/ 303353/2/ 303349/3/
f 303348/4/ 303377/5/ 303353/6/
...
f 86686/1806172/ 86715/1806173/ 86687/1806174/
```

From the PDS standpoint, the OBJ file is a collection of tab-separated tables. Each of the major sections (v, vt, f) are considered as a table with three or four elements (the type code and the parameters) and are described as such in the PDS label. (The one-line `mtllib` and `usemtl` entries are ignored by the label).

The MTL file is a simple text file that looks like this:

```
newmtl D053L8127T596979590RAS_F0101_0060M1
  Ka 0.2 0.2 0.2 1
  Kd 0.8 0.8 0.8 1
  Ks 0 0 0 1
  map_Kd D053L8127T596979590RAS_F0101_0060M1.PNG
```

The `newmtl` line provides the name used in the `usemtl` statement in the OBJ. `Ka`, `Kd`, and `Ks` define the ambient, diffuse, and specular colors of the material (see the OBJ docs) and are not used since there is a texture for every vertex in the mesh. The `map_Kd` line specifies the name of the PNG file.

If there is more than one texture file, that block can be repeated multiple times, but this is not an expected use case for InSight meshes.

5.3.4 “Unified” Terrain RDRs, PDS format

There are two ways that “unified” terrain meshes may be produced on InSight. The first is to generate a “unified” OBJ file combining multiple XYZ input files, in much the same way as the operations product. Such files would follow the same OBJ format as described above, except they would use the unified mesh filename convention.

The second is to create a Meshlab [30] “project” file (“.mlp”), which simply loads each of the per-image OBJ files as a unit in Meshlab, a popular open-source mesh viewing tool. In that way, it is analogous to the unified “*.iv” file (item “a” in Section 5.3.2). The file is an XML file which looks like the following:

```
<!DOCTYPE MeshLabDocument>
<MeshLabProject>
  <MeshGroup>
    <MLMesh filename=" D052L8127T596979382XYZ_G0101_0060M1.obj" label="
D052L8127T596979382XYZ_G0101_0060M1.obj">
      <MLMatrix44>
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
      </MLMatrix44>
    </MLMesh>
    <MLMesh filename=" D053L8127T596979590XYZ_G0101_0060M1.obj" label="
D053L8127T596979590XYZ_G0101_0060M1.obj">
      <MLMatrix44>
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
      </MLMatrix44>
    </MLMesh>
  </MeshGroup>
  <RasterGroup/>
</MeshLabProject>
```

The MLMesh element is repeated for each XYZ image in the mesh. From the PDS standpoint, this is treated as a simple text file.

Note that the MeshLab Project files are stored in a different directory (rdr/mesh) than the OBJ/MTL/PNG files (rdr/idc) in both the operational store and the archive. The MeshLab Project files

assume all the files are in the same directory, however. Thus, users may need to copy the relevant files to a single directory before loading in MeshLab.

InSight is expecting to use the second form (Meshlab file) systematically, although the first form (large OBJ) may be created from time to time.

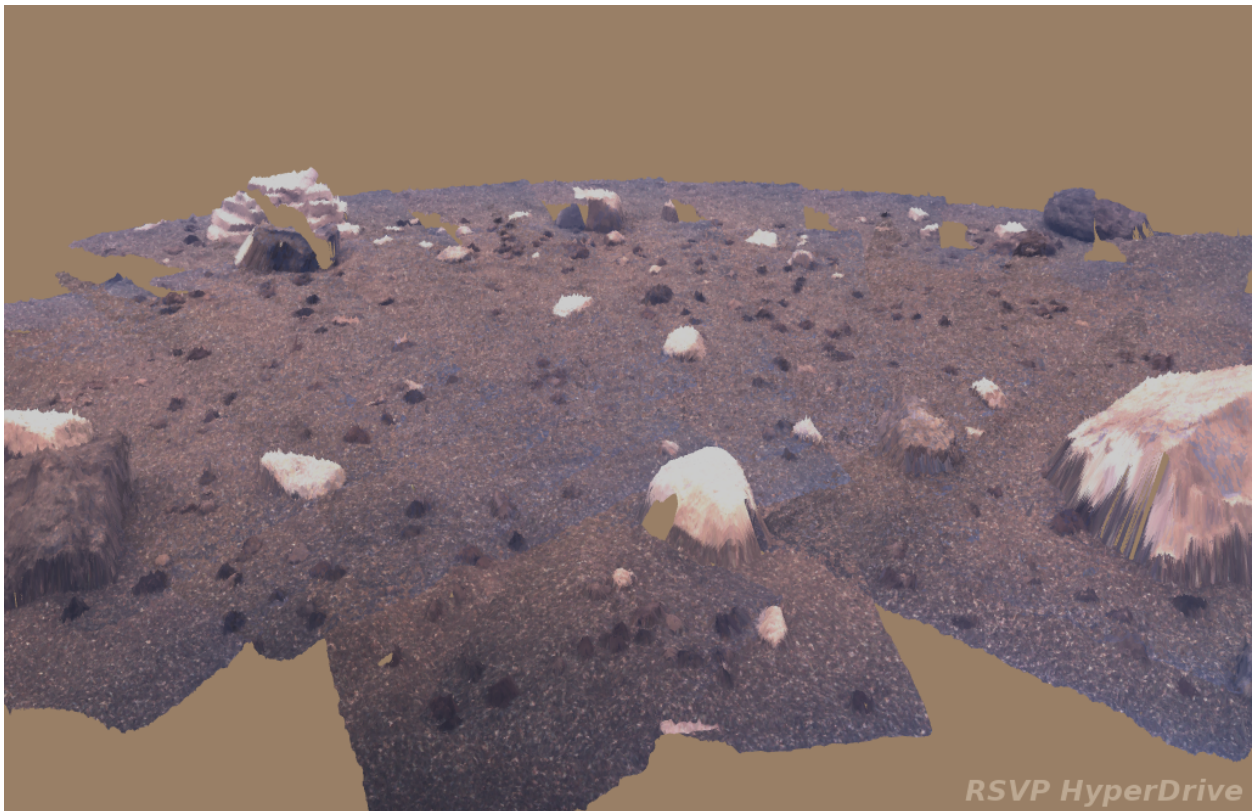


Figure 5-16: DeploymentWorkspace Rendered in Unified Terrain Mesh

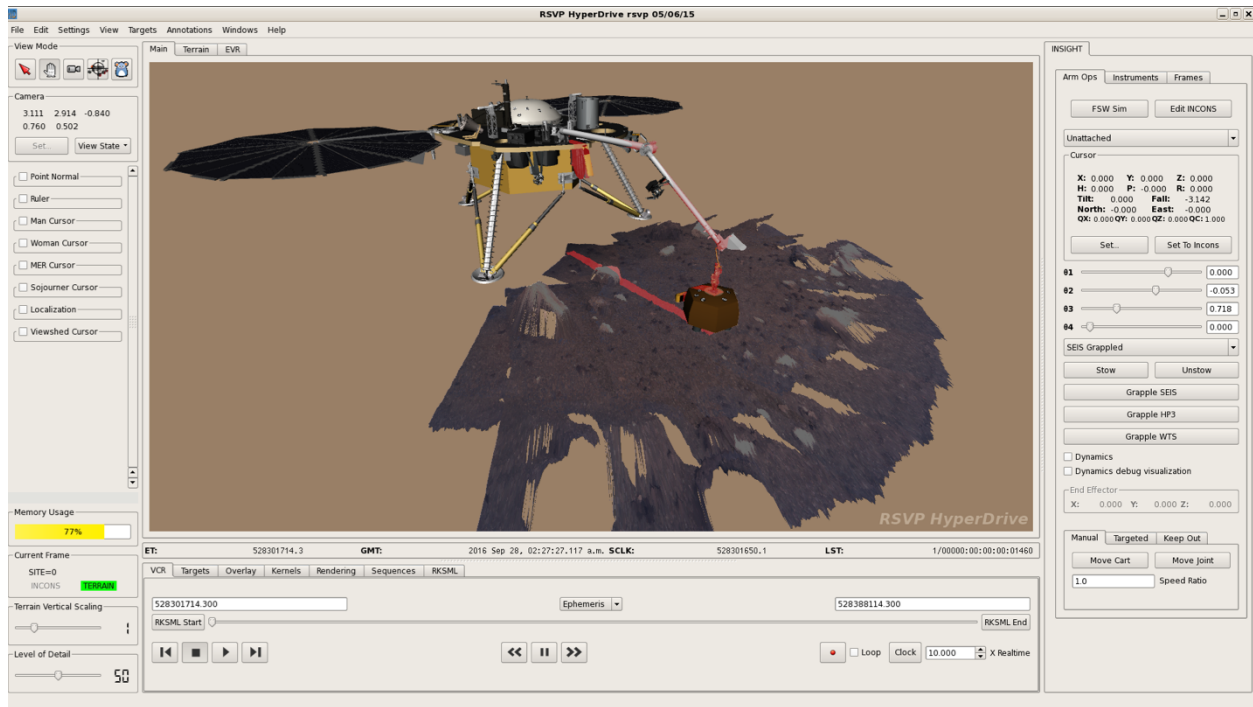


Figure 5-17: Instrument Deployment Location Rendered in Unified Terrain Mesh

5.4 Mosaic RDR Data Product Specification

This section discusses the process of mosaicking multiple frames into a single RDR product using some projection. The text largely reflects the methods applied by MIPL. It should be noted that governing methods and software can differ between MIPL and other operations subsystems or science instrument teams; the algorithms followed by other teams may not be the same as described here.

5.4.1 Overview of Mosaics

Mosaics embody several important properties, making them very useful products. They assemble small pieces into a larger field of view, creating a better contextual view of the lander's surroundings. Certain projections create a level horizon, removing lander tilt. Mosaics can be calibrated so directions such as north and east can be determined - or they can be made relative to the lander to visualize forward and right. They can provide overhead views (Vertical, Polar, or Orthorectified projections) to help understand the local environment and place the instruments. They can be made from different types of data (such as slope). They can combine different filters to create color. They can coregister data from different cameras, including orbital images. Finally, they are the signature products for public outreach.

Mosaics can be assembled autonomously, or with manual assist to correct pointing errors and brightness mismatches. They can consist of two frames (or sometimes even one), or hundreds of images. In short, mosaics are among the most important imaging products for any *in-situ* mission.

5.4.2 Pointing Correction

An optional (but important) first step in mosaic production is pointing correction. This is used to minimize geometric seams (discontinuities) between frames. The results of pointing correction are used in mosaics, but they can also be fed back into the RDRs (often at the ILUT stage), resulting in adjusted XYZ and other derived values and corrected meshes. This is particularly important for InSight, because its single arm mounted camera design leads to more error in stereo baseline knowledge and thus more errors in the meshes. Pointing correction can alleviate these errors and thus lead to more accurate meshes.

There are several methods by which improved pointing of the cameras can be determined. The most common method is to pick tiepoints between image pairs, either automatically or with manual assistance, and use those in a global cost function minimization (or bundle adjustment) to determine the corrected pointing parameters. Another possibility is to analyze the shape of XYZ data in the overlap region, again using an error minimization process to derive updated pointing parameters. Pointing parameters can also be determined manually.

There are three methods of using tiepoints, that differ in what function is being minimized during bundle adjustment. One, called the “traditional” method, projects the tiepoint from one image to the surface model, then projects it into the second image. The difference (in pixels) between where the projection intersects the image, and where the tiepoint predicts it should intersect, is the error. Errors across all tiepoints are added to create the final error metric. This method works very well for traditional mosaic projections, such as cylindrical, vertical, and polar, because it precisely mimics the mosaic process itself (Section 5.4.3). However, it does not work well for stereo, especially for arm-mounted stereo like InSight uses, because it tends to remove the very parallax that makes stereo analysis work.

The second method is called “miss-distance” tiepoints. In this, rays from each of the two images making up the tiepoint are projected into space. If pointing was perfect, these rays would cross; that’s fundamentally how XYZ points are determined. To the extent they miss each other, that represents pointing error. So, the error metric is the sum of the “miss distances” in XYZ space between the two rays, across all of the tiepoints. This method works better for stereo because it preserves parallax; the fact that no surface model is used means there is no parallax distortion. However, it does not work as well for traditional, non-stereo mosaics.

The third method is called full bundle adjustment. It is similar to the “traditional” method in the sense that the sum of differences between tiepoint pixel locations and projected locations is minimized. The main difference is that instead of using a surface model, the XYZs from each tiepoint are optimized simultaneously with the pointing adjustment. Initial estimates of XYZ are obtained from the tiepoints, that is, rays from each of the two images are projected into space and the XYZ is defined as being at the midpoint between the two rays at closest approach (if pointing was perfect, the two rays would intersect).

Regardless of method, the result of the bundle adjustment is encapsulated in a pointing correction or "nav" file. A more detailed description of this file is provided in Section 5.4.6 but fundamentally, this

file contains, for each image being corrected, the original pointing parameters and the revised pointing parameters.

Pointing parameters are simply a set of numbers which represent how the camera is pointing in the lander frame, reduced to available degrees of freedom. These are used as inputs to the kinematics procedures, which in turn derive the camera model. The set of pointing parameters, together with the kinematics algorithm, is referred to as a pointing model. The job of a pointing model is to take a calibration camera model and transform it using the pointing parameters to create a transformed camera model, which represents the specific image in question.

A given camera may have multiple available types of pointing models. For example, typical mast-mounted cameras have a standard model with two parameters: azimuth and elevation actuator angles, but also have another model that adds “twist” (or frame rotation) about the pointing axis. For Insight’s IDC, there are two models: one with six parameters (the XYZ position plus the three Euler angles describing the orientation) and one with seven (XYZ plus the four components of a quaternion – there are not actually 7 degrees of freedom since there is some correlation amongst the 4 quaternion numbers, but it is a convenient format to use). The ICC models have zero parameters, since they are rigidly attached to the lander body. Other pointing models may be created as well.

The set of available pointing model types and their full descriptions are outside the scope of this SIS; they are defined in a PDS documentation file.

For the mosaic process, the “nav” file is sufficient to describe the pointing parameters. However, if other corrected RDRs need to be produced, such as XYZs or meshes, then the pointing parameters must be stored in the label. This is accomplished via the POINTING_MODEL_NAME and POINTING_MODEL_PARAMS labels. When recomputing a camera model, if these labels are present they should be used in preference to the normal method of pointing via labels in the ARTICULATION_DEVICE_STATE groups. For most users, however, the GEOMETRIC_CAMERA_MODEL should be used directly; this will be updated properly with respect to the corrected pointing.

5.4.3 Brightness Correction

Mosaics often exhibit visible differences in brightness and contrast at the seams between frames. This is reduced by the use of radiometric and color correction, which removes the effects of the camera and provides (in the case of radiometric correction) a measure of the actual light entering the camera. However, these do not take into account photometric effects – differences in scene brightness due to time of day or light scattering properties of the objects in the image. Even with perfect radiometric correction, there are often still visible seams due to photometric effects. Also, radiometric correction is not perfect; most notably, lens flares from looking toward the sun are not removed. For these reasons, many mosaics go through a brightness correction process.

The brightness correction process used by MIPL for InSight is similar in some respects to the pointing correction process. First, “tiepoints” analogues are gathered, called overlaps. In areas where two or more frames of the mosaic overlap, the mean and standard deviation of the overlapping regions are

determined for each image. The size of the overlap area is limited, so if there is a large overlap between two images it will be broken into several smaller overlaps (this provides more opportunity for outlier rejection).

Then, an overall multiplicative (contrast) and additive (brightness) factor is determined for each image. The factors chosen minimize the “tiepoint error”, i.e. they attempt to match the statistics in the overlaps across all images. This uses a function minimization process similar to bundle adjustment. The number of pixels in each overlap is used to weight the overlaps such that each pixel has the same effect on the result. During this process, outlier overlaps are rejected. These outliers are common when the lander is being imaged, because of both the prevalence of specular reflections, and parallax effects.

Each image has a single multiplicative and additive factor applied to the entire image – corrections within the image are not performed in this algorithm. It can thus handle differences in intrinsic brightness fairly well, but does not work well with nonlinear effects such as uncorrected vignetting, photometric scattering differences, lens flares, etc.

The results are stored in a brightness correction file, which is analogous to the “nav” file in pointing correction.

For color images, this process is performed on the intensity channel after converting the image to HSI (hue, saturation, intensity) space. Hue and saturation are generally not corrected, nor do they generally need to be. It is possible to correct the R,G,B channels separately instead, but this has not worked well in practice.

It is also possible to correct using the mean and standard deviation of the entire image rather than overlap areas, but this is rarely done in practice.

5.4.4 Mosaicking Method for Most Mosaics

The process used by the MIPL software to create mosaics (except for orthorectified projection) is described below. It consists of several steps. Conceptually, one can think of the process as adjusting the pointing of the inputs, projecting them down to a surface, and looking at the result from a different point of view (the output projection). In reality, the process is run in reverse for ease of interpolation (this is described below).

- A. ***Output Projection Determination*** - The output projection is determined based on the geometry of the inputs. The labels describing the projection are described in detail in Appendix C. The output projection parameters are determined by analysis of the inputs to give the "best" resulting mosaic, but can be overridden by the user. The determination process is outside the scope of this document; the results are what is important and they are in the label.
- B. ***Surface Determination*** - A surface model is critical for mosaics (except for orthorectified mosaics, which don't use one). This is a mathematical surface, which approximates the actual scene. To the extent that the scene differs from the surface model, distortion and uncorrectable seams due to parallax can result.

Usually the surface model is a flat plane, with normal pointing upwards. This can be adjusted, however, to better match the scene. Regardless, the results are documented in the SURFACE_MODEL_PARMS group.

There are five potential surface models in the MIPL software: PLANE, INFINITY, SPHERE, SPHERE1, and SPHERE2. See SURFACE_MODEL_TYPE in Appendix C for description. Note that an appropriate surface model is often determined automatically as part of the pointing correction process, and the surface model can be stored in the “nav” file. Almost all mosaics are created using the PLANE model.

- C. **Computation of Output View Ray** - For each pixel in the output mosaic, a view ray in 3-D space is constructed. How this view ray is constructed depends on the projection type. In this section, the pixel is at location (i,j) in 0-based coordinates, with i corresponding to sample and j to line. (0,0) is in the upper-left-hand corner. Capitalized values represent VICAR label items from the SURFACE_PROJECTION_PARMS group. Unit and coordinate system conversions are applied as necessary but are not specified here. The coordinate system used is defined by REFERENCE_COORD_SYSTEM_* in SURFACE_PROJECTION_PARMS.
- D. **Projection from Output to Surface** - Once the view ray is determined, it is projected out until it intersects with the surface model. The resulting point in XYZ space is used in the next step. If the ray does not intersect the surface, the point is assumed to be at infinity in the direction the view ray is pointing. Exception: as mentioned below, the Vertical projection will reverse the direction of its view ray; infinity is assumed only if they both miss.

Note that the INFINITY surface model guarantees the ray will miss the surface at all times.

The difference between the SPHERE1 and SPHERE2 models is that, if the ray intersects the spherical surface more than once, SPHERE1 will take the first intersection, while SPHERE2 will take the second. For normal situations, SPHERE1 thus roughly models a convex hill, while SPHERE2 roughly models a concave crater when the lander is outside the sphere.

- E. **Projection from Surface to Input** - The XYZ location (or direction for the infinity case) is then back-projected into each input image in turn, using the corresponding input camera model. The first input for which the resulting pixel coordinate is inside the image (excluding border pixels which are thrown away) and non-0 stops the process; that is the image from which the output pixel value is taken. Values of 0 in the input image are ignored, with the effect that they are transparent.

This process results in stacking the images such that the first one in the input list of images "wins". There is no feathering of overlaps; the first image is "on top" of all the others, and an image completely covered by preceding images will not be used at all.

- F. ***Interpolation and Storage of the Result*** - Finally, a bilinear interpolation is (optionally) performed on the input image to reduce aliasing, based on the 4 pixels surrounding the back-projected location. The result of this interpolation is the value of the output pixel.

Bilinear interpolation is optional, but is normally done for image mosaics. Mosaics of other data types such as XYZ or Surface Normal (UVW) generally have interpolation turned off to avoid aliasing from interpolation with invalid pixels.

5.4.5 Mosaicking Method for Orthorectified Mosaics

The orthorectified mosaic is projected to a plane in a similar manner to the orthographic or vertical projections. However, unlike the other projections, the XYZ locations of the pixels are taken into account. This is what allows parallax to be removed. In a nutshell, the X and Y coordinates directly define where the pixel goes in the output, rather than projecting rays down to a surface model. The color, or DN, of the output comes from the associated "skin" image (or the Z value of the XYZ for DEMs), which could be a traditional image or any RDR type. The challenges in this algorithm are dealing with overlaps, and appropriately sizing the pixels in the output to avoid salt-and-pepper effects (the farther away the pixels are, the farther apart they are in the orthorectified projection).

Unlike the other mosaic programs, marsortho proceeds input-to-output. It loops through each pixel of each input. XYZ coordinates are converted to a common output coordinate frame as needed. The output pixel "hit" by this input pixel is determined. If two or more pixels (either from the same image or different inputs) "hit" the same output pixel, the "winner" is determined according to one of several algorithms:

- MIN, MAX: The pixel with the minimum or maximum Z value is chosen.
- AVG: The DN is computed as the average of all overlapping inputs.
- FIRST, LAST: The first or last image in the image list wins. FIRST mimics how all the other MIPL mosaic programs work.
- CLOSEST: The pixel closest to its camera is the one that is chosen.

Once the winners are determined, the regions around the pixels are filled in. The region potentially covered by an input pixel in the output is determined based on geometry, using the angular size of the pixel and distance to the camera. This determines how many output pixels can potentially be affected by this input pixel. The region size can be adjusted via parameters; generally it is enlarged in order to fill small holes due to missing pixels, while leaving larger holes in place. Any output pixel not directly "hit" is filled by looking at all such regions covering the pixel in question and using the one whose center is closest to it.

In reality, winner determination and filling are computed simultaneously, but the effect is as described above.

Afterwards, the program optionally performs morphological erosion to remove the repeated-rectangle pattern at the borders of the mosaic, which occurs when the pixel region is enlarged.

5.4.6 Mosaic Ancillary Files

A number of ancillary files are used to support mosaicking, and contain parameters and information describing how the mosaic was produced. With these, it is possible to maintain traceability and provenance for each pixel in a standard mosaic back to the source image. Ancillary files might not necessarily have the same basename as the mosaic, as several mosaics might share an ancillary file. In such cases, the ancillary file's name may indicate a different product type, projection, eye, filter/color, or geometric or brightness correction than the target mosaic's filename. The other filename fields should always match. The history label of the mosaic indicates which ancillary files were used.

It should be noted that many if not most mosaics are produced at least partially by hand, which explains most of the inconsistencies noted below. The general case is described, but as with any hand work not all conforms exactly.

These files are not described completely here, but we hope the descriptions are sufficient to be able to decipher them:

- a) **List files** - With a ".LIS" extension, list files are simple text files containing the names of the images making up the mosaic, one per line. The first name in the list references the image frame that is "on top" in the mosaic product, covering the image frames that are referenced in the list below it. The list files often contain full pathnames to disks on the operational data store (SPeX), directory paths which are not part of a PDS-released archive volume. However, the filenames themselves, minus directory paths, usually will be part of an archive volume. Occasional mosaics may have list files comprised of names for private copies of images (e.g. with "/home" in the pathname) that are the result of different scenarios of special processing: specified and unspecified. In the former case, such image files will be in an archive volume with filenames that carry a character flagging the type of special processing. In the latter case, no special character is present in the filename, though the file's name and/or metadata label will identify the ultimate source image of the unspecified processing. Note that a mosaic's list file content of filenames (minus directory paths) is also referenced in the mosaic's label using keyword INPUT_PRODUCT_ID.
- b) **Nav files** - With a ".NAV" extension, nav files are XML files describing the pointing corrections that have been applied to images in a mosaic, as well as the surface model. The prologue contains identifying information.

Pointing correction works by applying a set of pointing parameters (e.g. mast azimuth and elevation, or XYZ position and Euler angles) to a kinematics algorithm and using that to re-point the camera model. See definition of keyword POINTING_MODEL_NAME in Appendix C for more information.

For each image (<solution> element), the Site and Drive components of the RMC are listed, followed by image identifier information. This identifier information contains the original pointing parameters, which allows the same correction to be applied to e.g. the left and right eyes (regardless of image ID information).

Following that are the updated pointing parameters, and then the revised (re-pointed) camera model. In rare cases, nav files are edited by hand, which puts the accuracy of the camera model update at risk. This is unlikely to occur for InSight, however.

At the end is usually a “surface_model” element describing the surface model determined by the MIPL software program MARSNAV or MARSNAV2 (which create the nav files). This indicates the surface model used during nav file creation (MARSNAV2 does not use a surface model). This is then almost always the surface model used during mosaic processing (unless overridden on the command line). In any case, the SURFACE_MODEL_PARMS group of the mosaic product’s label describes the model used for the mosaic.

- c) **Tiepoint files** - With a “.TIE” or “.TPT” extension, tiepoint files are XML files containing image tiepoints used as input to program MARSNAV/MARSNAV2. These can be automatically or manually selected. The prologue relates each image ID to a key, which is used throughout the remainder of the file.

Each tiepoint has a left and right key, and then 1-based coordinates of the point in the corresponding files of the tiepoint. Just <left> and <right> should be used; <projected> has little value. In the flags, “quality” represents the quality of correlation match when the tiepoint was correlated (scale 0-1). The “interactive” flag has little archival value, it is used during the tiepoint gathering process. It does not (as the name suggests) indicate whether the tiepoint was automatically selected or manually tweaked.

Tiepoint type “0” is by far the most common and is a standard image tiepoint. Type 1 is occasionally used; it ties an image to a specific XYZ location (“fiducial” tiepoint); the coordinate frame for these points should be defined in the prologue. Other types are rarely used and their descriptions are beyond the scope of this document. Full descriptions are in the help documentation for VICAR program MARSNAV.

- d) **Brightness Correction files** - With a “.BRT” extension, brightness correction files are XML files containing information used to correct the brightness and contrast of images in a mosaic relative to one another. They are similar in concept and structure to nav files. After the prologue, each image has one <brt_solution>. The most common correction type, LINEAR, specifies an overall additive and multiplicative factor to be applied to each image (MULT is applied first, then ADD). These factors are echoed in the IMAGE_RADIANCE_FACTOR and IMAGE_RADIANCE_OFFSET keywords in the mosaic label. The HSI_LIN type is similar, except the correction (for color images) is done in Hue-Saturation-Intensity (HSI) space, with the correction applied to Intensity only. See also BRIGHTNESS_CORRECTION_TYPE.
- e) **Brightness Overlap files** - With a “.OVR” extension, brightness overlap files are XML files containing information about image statistics in overlap areas, used to create the brightness correction files. They are similar in concept to tiepoint files, except the “tiepoints” are the mean and standard deviation of small areas of overlapping pixels for the mosaic.

They start with a prologue defining the image ID-to-key mapping, as with nav files. Each overlap then has a number of images involved in the overlap, the number of pixels, and a “radius” which is a general description of the maximum size of the overlap. This is followed by the key, mean and standard deviation of the overlapping area in each image. The line and sample coordinates are provided for an arbitrary point in the overlap, just to help locate where

the overlap is. The actual shape of the overlap (or the location of the coordinate relative to the overlap) is not specified.

Overlap type “0” is a standard overlap. Type “1” gives the mean and standard deviation not of an overlap, but of the image as a whole (thus there is only one image underneath <overlap>). Type “2” is like type “0” but has mean and standard deviation in HSI space (intensity only). Type “3” is like type “1” but using HSI space. See the help documentation for MIPL software program MARSBRT for full details.

5.4.7 Cylindrical Projection Mosaic RDR

Cylindrical projections are the most common method for viewing non-stereo panoramas.

The MIPL method for creating a Cylindrical projection involves computing the azimuth and elevation of the view ray, as follows:

$$\text{azimuth} = i / \text{MAP_RESOLUTION} + \text{START_AZIMUTH}$$

$$\text{elevation} = (\text{ZERO_ELEVATION_LINE} - j) / \text{MAP_RESOLUTION}$$

The view ray emanates from the point PROJECTION_ORIGIN_VECTOR.

Figure 5-18 shows such a mosaic overlaid onto azimuth and elevation grid lines. In this case each pixel represents a fixed angle in azimuth and elevation. Rows are of constant elevation in the selected coordinate frame. In this case, a Site frame was used, so the horizon is level, and columns begin clockwise from Mars south (centered on north). Figure 5-19 shows a cylindrical projection of the lander deck. Grid lines are optional when the mosaic is created, and may or may not appear in any given mosaic at the discretion of the creator (most commonly, grids are “underneath” the image, showing only where the mosaic is blank).

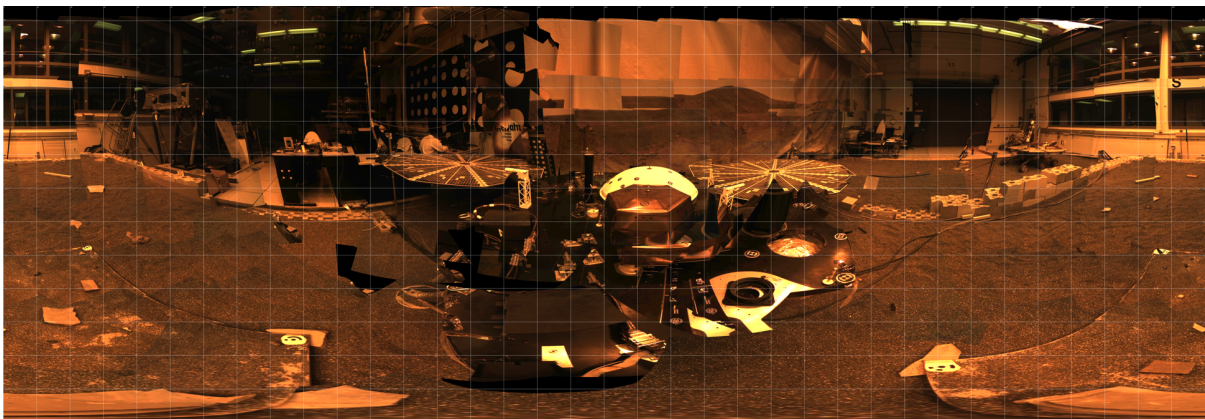


Figure 5-18: Cylindrical Projection Mosaic of the Lander Environment (from IDC)

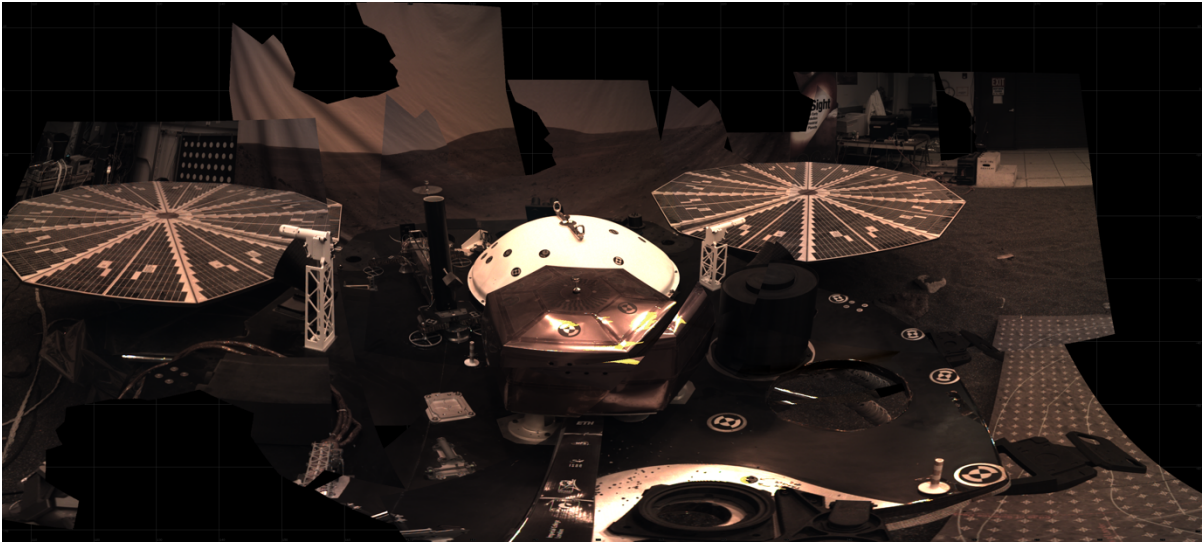


Figure 5-19: Cylindrical Projection Mosaic of Lander Deck

5.4.8 Camera Point Perspective Mosaic RDR

Figure 5-20 shows a Camera Point Perspective mosaic. It is a perspective projection with horizontal epipolar lines. The mosaic behaves as though the "camera" which acquired the image frames was an instrument with a much larger field of view. For InSight, this type of mosaic is typically in the Lander Frame and thus may have a tilted horizon if the lander is not level.

Point-perspective mosaics give the most natural view of small areas and are suitable for stereo viewing, but cannot be used for wide fields of view.

MIPL creates the Camera Point Perspective by using the output camera model (described by the `GEOMETRIC_CAMERA_MODEL` group in the output mosaic) to project the pixel into space. The origin of the view ray is thus the C point of the camera model, with the ray's direction being determined by the camera model. See Section 5.2.1.3 and references [15] through [23] for the mathematics.

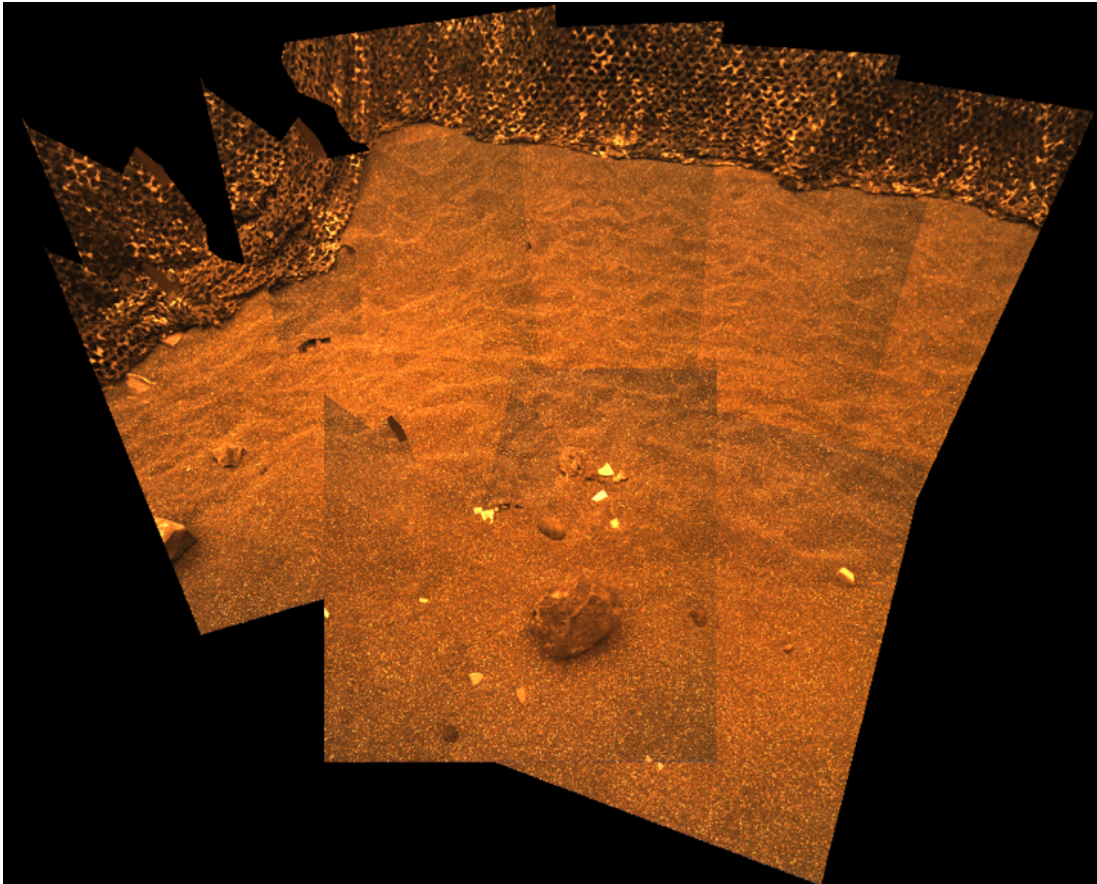


Figure 5-20: Point-perspective mosaic

5.4.9 Cylindrical-Perspective Projection Mosaic RDR

Cylindrical-Perspective mosaics are used for large stereo panoramas, and work across a full 360 degrees of azimuth. Stereo is preserved because a baseline separation is maintained between the camera eyes at different azimuths.

This projection is the most complicated projection to create. Each column i (counting from 0) in the output mosaic is assigned its own camera model. This is done in several steps:

- 1) Compute initial camera model. This model is a CAHV linearized model derived from the first input to the mosaic, re-pointed to azimuth 0 and elevation PROJECTION_ELEVATION. This model is stored in the GEOMETRIC_CAMERA_MODEL label group.
- 2) The instantaneous field of view of the "central" pixel (at the point where the A vector intersects the image plane) is computed using the formula:

$$\text{ifov} = \text{atan} (1.0 / |(\vec{H} - \vec{A} * (\vec{H} \cdot \vec{A}))|)$$

where the “•” indicates the scalar dot product of the two vectors A and H.

Alternatively, this can be derived from the image size and azimuthal extent (where the azimuths are adjusted by 360 degrees such that the result is minimally positive):

$$\text{ifov} = (\text{STOP_AZIMUTH} - \text{START_AZIMUTH}) / \text{LINE_SAMPLES}$$

- 3) The azimuth of the column is computed:

$$\text{azimuth} = \text{START_AZIMUTH} + i * \text{ifov}$$

- 4) The initial camera model is re-pointed using kinematics as described above under the pointing correction section, using the above azimuth and PROJECTION_ELEVATION. This results in the final camera model for the column.

Step 4 is difficult to duplicate for reconstructing the set of camera models. For that reason, an alternate method is described in this paragraph. The resulting models are exact for mast-mounted cameras with no backlash correction; they are a close approximation for other cases. For the purposes of this discussion, the InSight IDC acts like a mast-mounted camera when gathering a 360-degree panorama. In general, for mast-mounted cameras, the C points of the column camera models describe a ring in space, whose diameter is approximately the baseline between the cameras. This ring is described by PROJECTION_ORIGIN_VECTOR (center), PROJECTION_AXIS_OFFSET (radius), and CAMERA_ROTATION_AXIS_VECTOR (orientation of the ring axis). These together simulate the kinematics motion of a mast-mounted camera in lander frame. To compute the camera model for the azimuth defined in Step 3, take the camera model from the label, and rotate the entire camera model around the camera rotation axis by the azimuth amount, using the ring center as the pivot point. The C point will remain on the ring, while the camera pointing (close to but not identical to the A vector) will remain approximately tangent to the ring at that point. After this, compute the rotation required to transform CAMERA_ROTATION_AXIS_VECTOR into PROJECTION_Z_AXIS_VECTOR (which can be done by taking the cross product to get the rotation axis and the dot product to get the rotation amount). Then rotate the camera model by this amount, again using the ring center as the pivot point. This has the effect of tilting the entire ring so it is perpendicular to the PROJECTION_Z_AXIS_VECTOR. This last rotation is often used to remove the effect of lander tilt, resulting in a flat horizon with the camera model baselines (vector between the left and right eyes) aligned with the horizon (technically, perpendicular to the Z axis in Local Level frame). For this "untilt" case, the PROJECTION_Z_AXIS_VECTOR is the Local Level frame's Z axis expressed in Lander frame. Note that PROJECTION_ELEVATION and PROJECTION_LINE are measured before this "untilt" rotation takes place, so they end up describing a sinusoid in the final mosaic when untilt is used.

Once the camera models have been defined, the mosaic proceeds through each pixel as with the other projections. The view ray is computed as described below (A, H, and V come from the column's camera model):

$$\begin{aligned} x_center &= \vec{A} \cdot \vec{H} \\ y_center &= \vec{A} \cdot \vec{V} \end{aligned}$$

$$\begin{aligned} \text{samp} &= \mathbf{x_center} \\ \text{line} &= \mathbf{y_center} + \mathbf{j} - \text{PROJECTION_ELEVATION_LINE} \end{aligned}$$

where the “•” indicates the scalar dot product of two vectors. This (samp, line) coordinate is then projected into space using the column's camera model, and this projection becomes the view ray. The origin of the view ray is the column's C point. See Section 5.2.1.3 and references [15] through [23] for the mathematics of camera models.

Figure 5-21 shows a Cylindrical-Perspective projection in which a 360 degree view can be viewed in stereo. This is a perspective projection similar to Figure 5-20 except that the mosaic acts like a pinhole camera which follows the mosaic in azimuth while maintaining camera baseline separation. If the mosaic is generated with no tilt correction (i.e., CAMERA_ROTATION_AXIS_VECTOR and PROJECTION_Z_AXIS_VECTOR are the same) and the lander is tilted, the horizon will not be level, instead being sinusoidal. This preserves epipolar alignment and allows for better stereo viewing of the panorama. However, for aesthetic reasons, Cylindrical-Perspective mosaics are often created by “untilting” the lander as described above. In these cases, the horizon will be level, but stereo alignment may be compromised due to parallax effects in areas where the surface model does not closely match the actual surface. Additionally, the overall baseline between the cameras may be adjusted via the ring radius (PROJECTION_AXIS_OFFSET). This has the effect of enhancing or reducing the overall disparity, which can result in better stereo viewing in some cases. This baseline adjustment may create similar parallax effects in areas where the surface model does not match the actual surface.

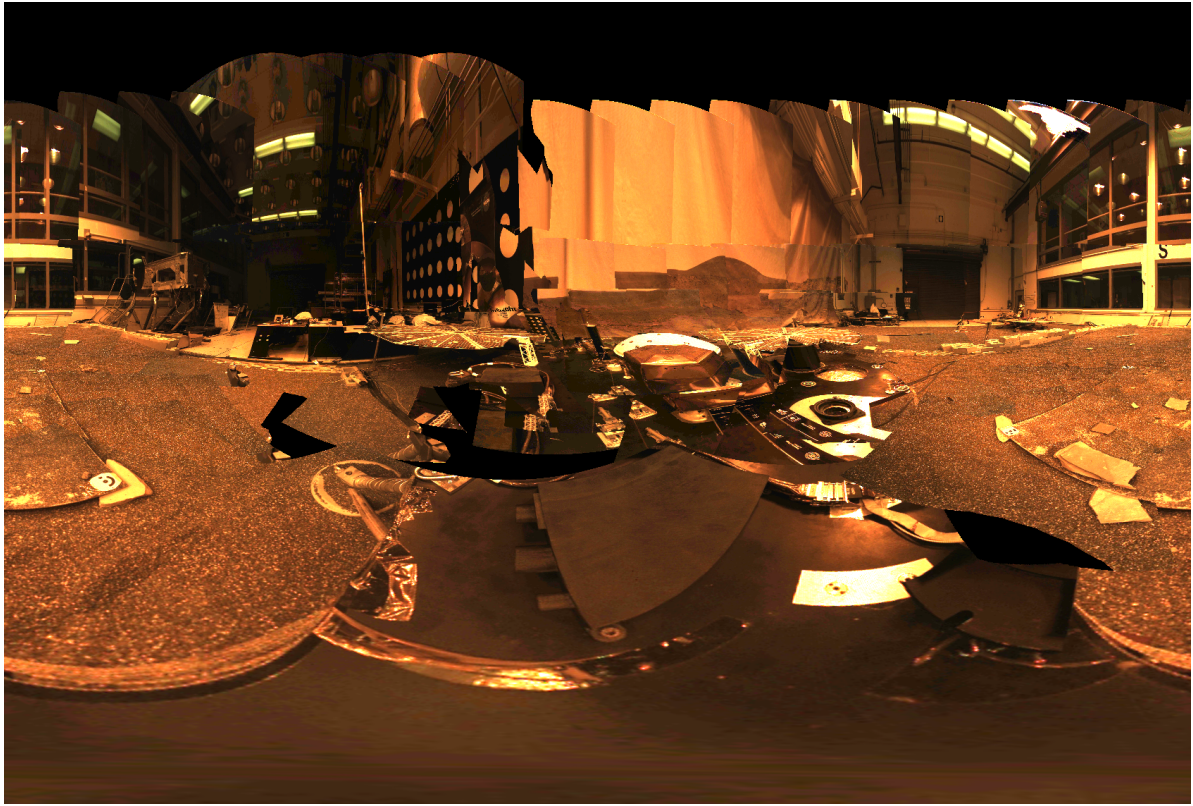


Figure 5-21: Cylindrical-Perspective Mosaic

5.4.10 Polar Projection Mosaic RDR

Polar mosaics create a quasi-overhead view that still allows viewing all the way to the horizon.

MIPL creates the Polar projection by computing the azimuth and elevation of the view ray as follows:

```

x = i - SAMPLE_PROJECTION_OFFSET
y = LINE_PROJECTION_OFFSET - j
range = sqrt(x*x + y*y)
elevation = range / MAP_RESOLUTION - 90 degrees
azimuth = REFERENCE_AZIMUTH + (90 degrees - atan2(y, x))

```

The view ray emanates from the point PROJECTION_ORIGIN_VECTOR.

Figure 5-22 shows a Polar projection. Concentric circles represent constant projected elevation. Mars nadir is at the convergent center and the horizon is corrected for lander tilt. North is up.

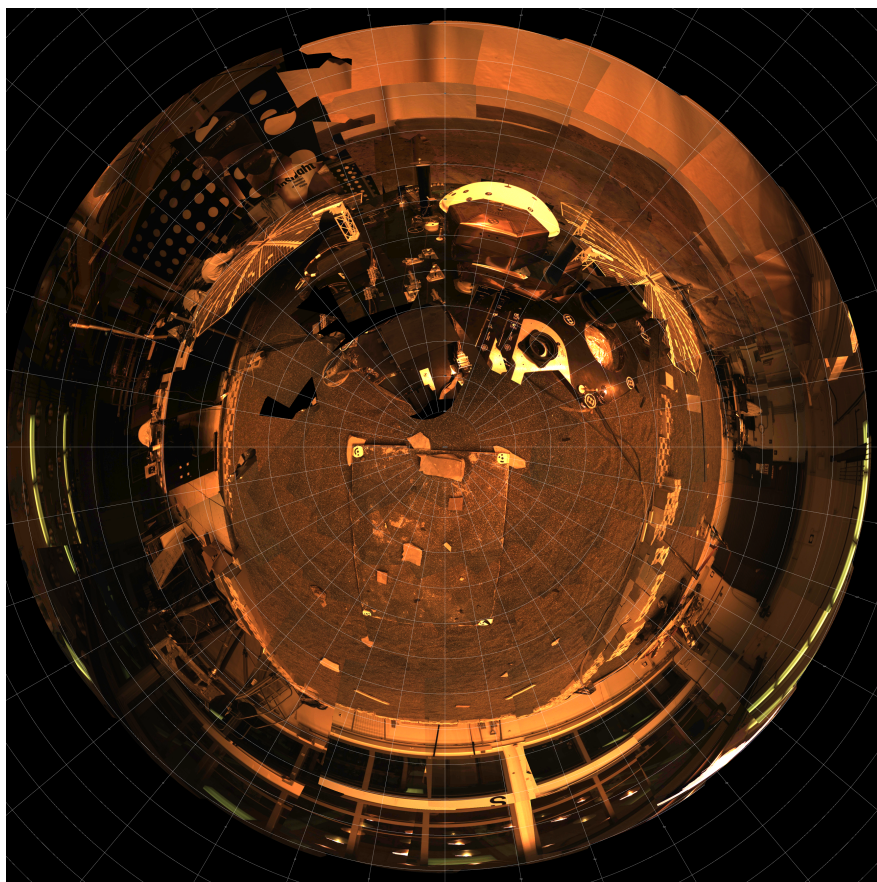


Figure 5-22: Polar Projection Mosaic of Lander and Workspace

5.4.11 Vertical Projection Mosaic RDR

Vertical mosaics provide a view of the surroundings as if you were looking straight down. They are thus quite useful for establishing the environmental context or comparing with orbital images, but suffer from severe distortion with any variance of the scene from the surface model. In particular, rocks are severely elongated, and the terrain is not taken into account (contrast this with Orthorectified, Section 5.4.13, where terrain is taken into account).

MIPL creates the Vertical projection as follows:

nl = number of lines in the mosaic (LINES)
ns = number of samples in the mosaic (LINE_SAMPLES)
 $x = (nl/2 - j) * MAP_SCALE$
 $y = (i - ns/2) * MAP_SCALE$

The view ray emanates from $(x, y, 0)$ and points straight down $(0,0,1)$. If the ray misses the surface in step D of Section 5.4.3 above, it is changed to point straight up $(0,0,-1)$.

Figure 5-23 and Figure 5-24 show a vertical view. It assumes that the field is a plane tangent to the Martian surface with up pointing north. This is not an orthorectified rendering but is still useful in many situations. Specifically, the vertically-projected ICC image from sol 0 will be the only workspace image available for the first few days after landing.

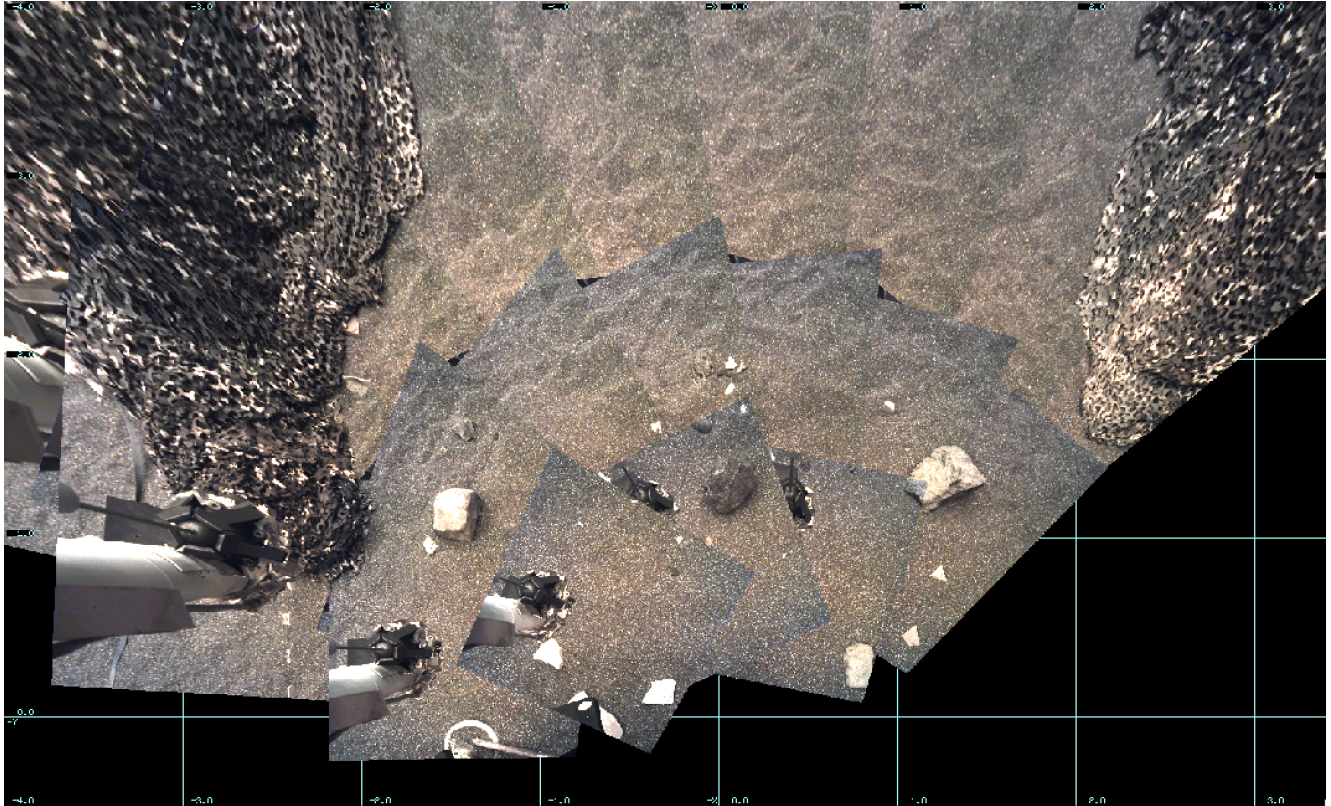


Figure 5-23: Vertical Projection Mosaic of Instrument Deployment Workspace. Arm is not masked off for purposes of illustration.

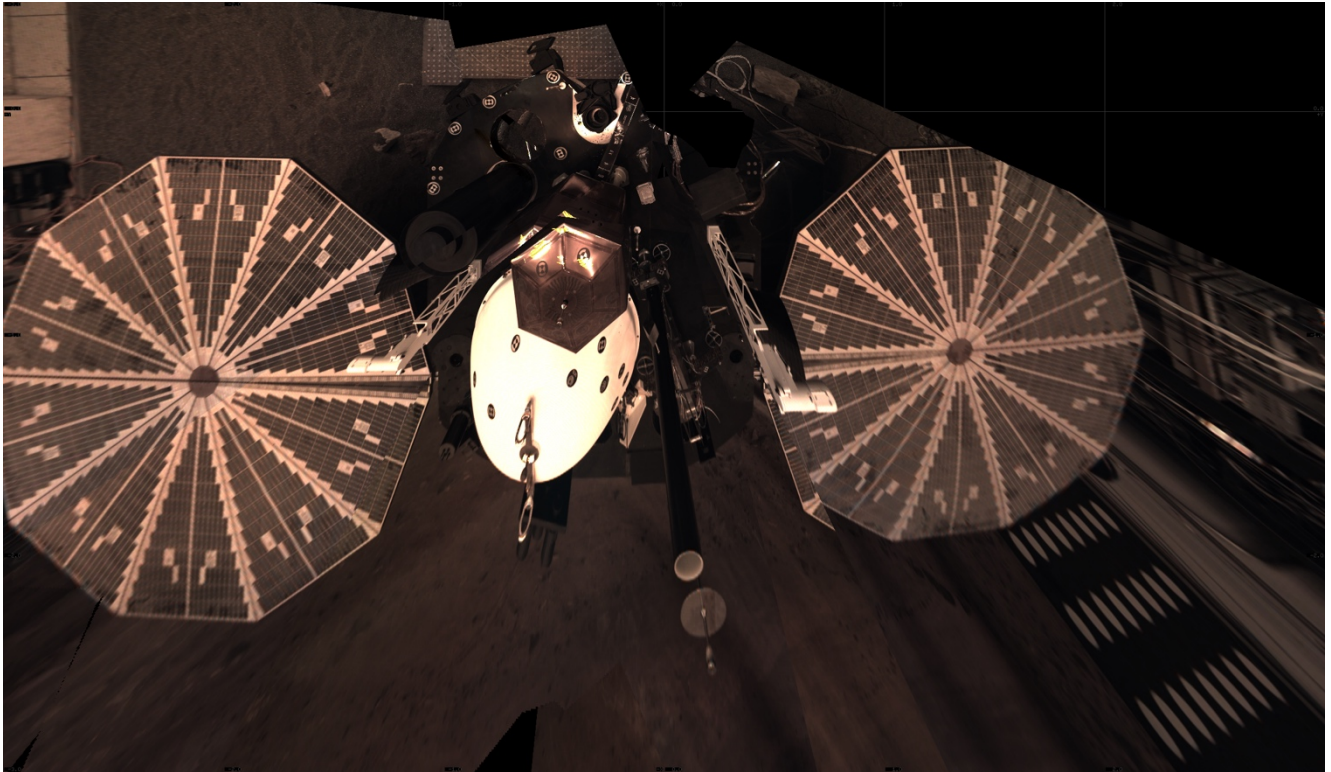


Figure 5-24: Vertical Projection Mosaic of Lander Deck

5.4.12 Orthographic Projection Mosaic RDR

The Orthographic projection is a generalization of the Vertical projection. It differs from Vertical in that an arbitrary projection plane can be specified.

If O is the point specified by the PROJECTION_ORIGIN_VECTOR and \hat{X} and \hat{Y} are the unit vectors given by PROJECTION_X_AXIS_VECTOR and PROJECTION_Y_AXIS_VECTOR respectively, then an arbitrary point P will have projection coordinates (X, Y) as follows:

$$X = (\vec{P} - \vec{O}) \cdot \hat{X}$$

$$Y = (\vec{P} - \vec{O}) \cdot \hat{Y}$$

where the “ \cdot ” indicates the scalar dot product of two vectors. PROJECTION_Z_AXIS_VECTOR is the direction of projection; the three vectors form a right-handed orthonormal basis.

All of these quantities must be specified with respect to a single frame defined by the REFERENCE_COORD_SYSTEM_NAME and REFERENCE_COORD_SYSTEM_INDEX. Additional relevant parameters for the projection are MAP_SCALE, X_AXIS_MINIMUM, X_AXIS_MAXIMUM, Y_AXIS_MINIMUM, and Y_AXIS_MAXIMUM.

A Vertical projection is the same as Orthographic with $\text{PROJECTION_X_AXIS_VECTOR} = (1,0,0)$, $\text{PROJECTION_Y_AXIS_VECTOR} = (0,1,0)$, and $\text{PROJECTION_Z_AXIS_VECTOR} = (0,0,1)$.

5.4.13 Orthorectified Projection Mosaic RDR

Orthorectified mosaics will be the most common mosaic projection used during the InSight Instrument Deployment planning. They are used to show a “true” view of the scene from a different point of view, without distortion due to parallax. The point of view is usually overhead, resulting in an image suitable for comparison to satellite images and mapping. The removal of parallax necessarily leads to holes or gaps in the mosaic, which do not occur with the other projections.

The Orthorectified mosaic is projected to a plane in a similar manner to the Orthographic or Vertical projections. However, unlike any of the other projections, the XYZ location of the pixels are taken into account. This is what allows parallax to be removed.

A simple way to think of this, for the case of an Orthorectified-Vertical projection, is to attach the XYZ coordinate (derived from stereo analysis) to each input image pixel, chop off the Z coordinate, and use the XY coordinates as the position in the output image. The more general (non-vertical) case is similar in concept, just rotate the XYZ values to the frame defined by the projection plane first.

The most common projection plane is to look straight down, which corresponds to the same point of view as the Vertical projection. For this case, $\text{PROJECTION_X_AXIS_VECTOR} = (1,0,0)$, $\text{PROJECTION_Y_AXIS_VECTOR} = (0,1,0)$, and $\text{PROJECTION_Z_AXIS_VECTOR} = (0,0,1)$.

See Section 5.4.5 for details on how orthorectified mosaics are created.

Figure 5-25 shows an orthorectified mosaic of the workspace.

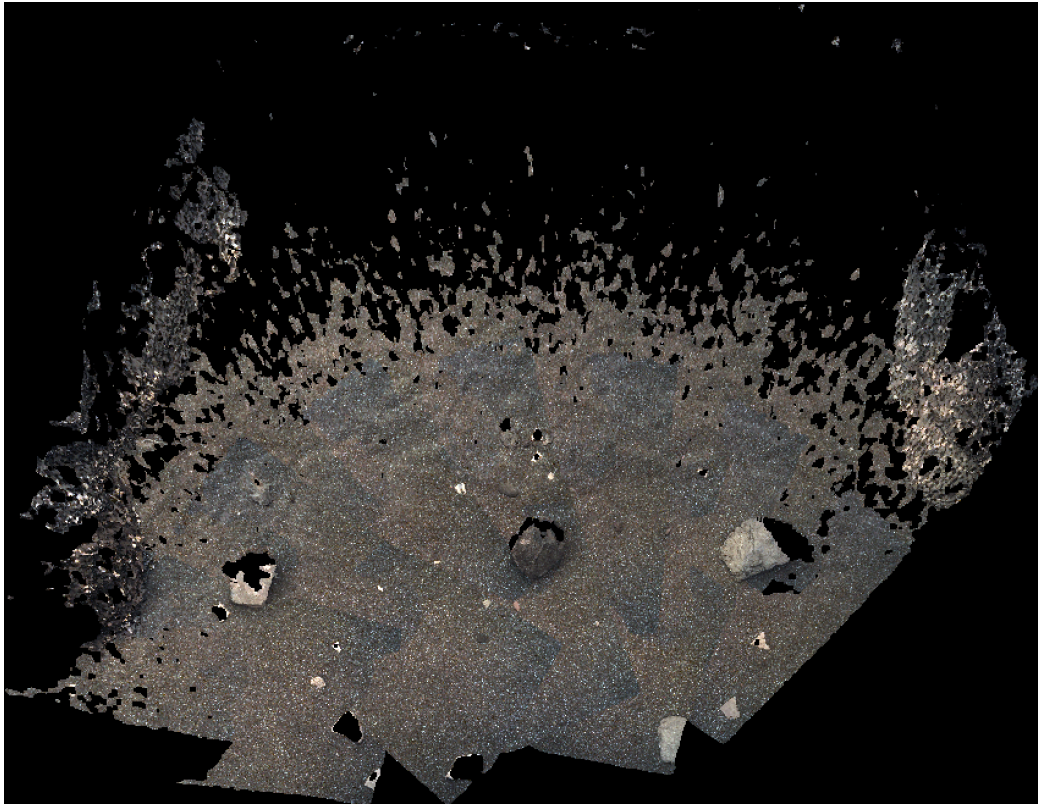


Figure 5-25: Orthorectified Projection Mosaic of Workspace

5.4.14 Non-image Mosaics

Non-image Mosaics will be very common for the InSight Mission. Normally mosaics are created using traditional images, where each pixel is either a raw or radiometrically corrected intensity value. However, mosaics can also be created using other types of pixels. In fact, any of the RDR types using an image format (e.g. not meshes) can be mosaicked. The most useful of these are mosaics of XYZ and the Instrument placement RDRs including instrument tilt, workspace reachability, surface roughness, and goodness.

For example, an XYZ mosaic contains XYZ values for each pixel in the mosaic rather than intensity values. The inputs to the mosaic program are XYZ files (or individual X, Y, or Z components), and the pixels are interpreted in the same way - as the coordinate of the corresponding pixel in Cartesian space. Like XYZ images, the mosaics may consist of a single 3-band file with X, Y, and Z components, or separate 1 band files for each component. A Z-only mosaic of a Vertical or Orthorectified-Vertical projection creates a digital elevation model (DEM) - approximate in the Vertical case, correct for Orthorectified-Vertical.

Care must be taken while producing these mosaics to ensure that a consistent coordinate system and data type are used for all the input images. No transform is done on the data; the output mosaic may have only one coordinate system in which the values are defined, and one DERIVED_IMAGE_TYPE.

Non-image mosaics are often created without interpolation; the nearest pixel is used instead. This avoids aliasing effects when pixels are interpolated with neighboring invalid pixels.

For the InSight mission, some non-image mosaics are created by running RDR programs on the mosaic, rather than mosaicizing the RDRs run on individual images. This is generally done for the instrument placement products (see Section 5.2.4) in order to avoid edge effects on the single-frame RDRs.

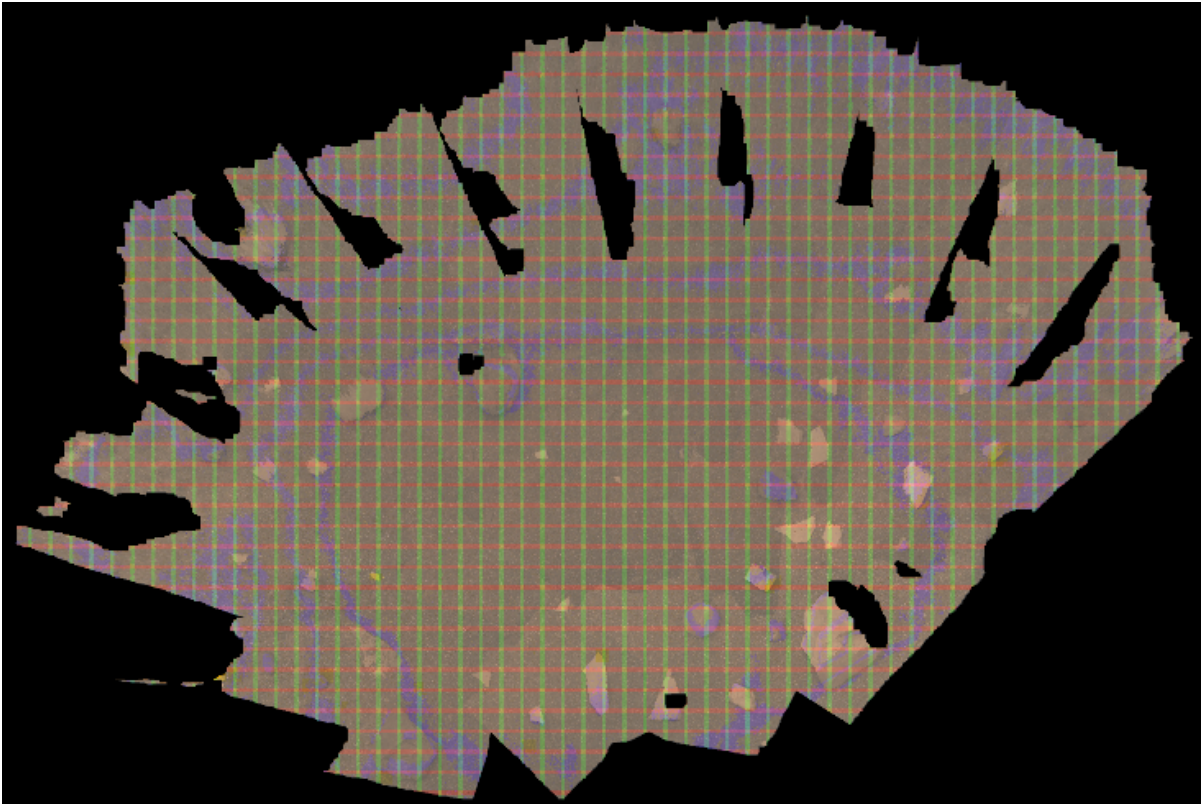


Figure 5-26: XYZ Mosaic Overlaid onto Orthorectified Mosaic. Red, green, blue are lines of constant X, Y, Z respectively (so blue is a traditional contour map).

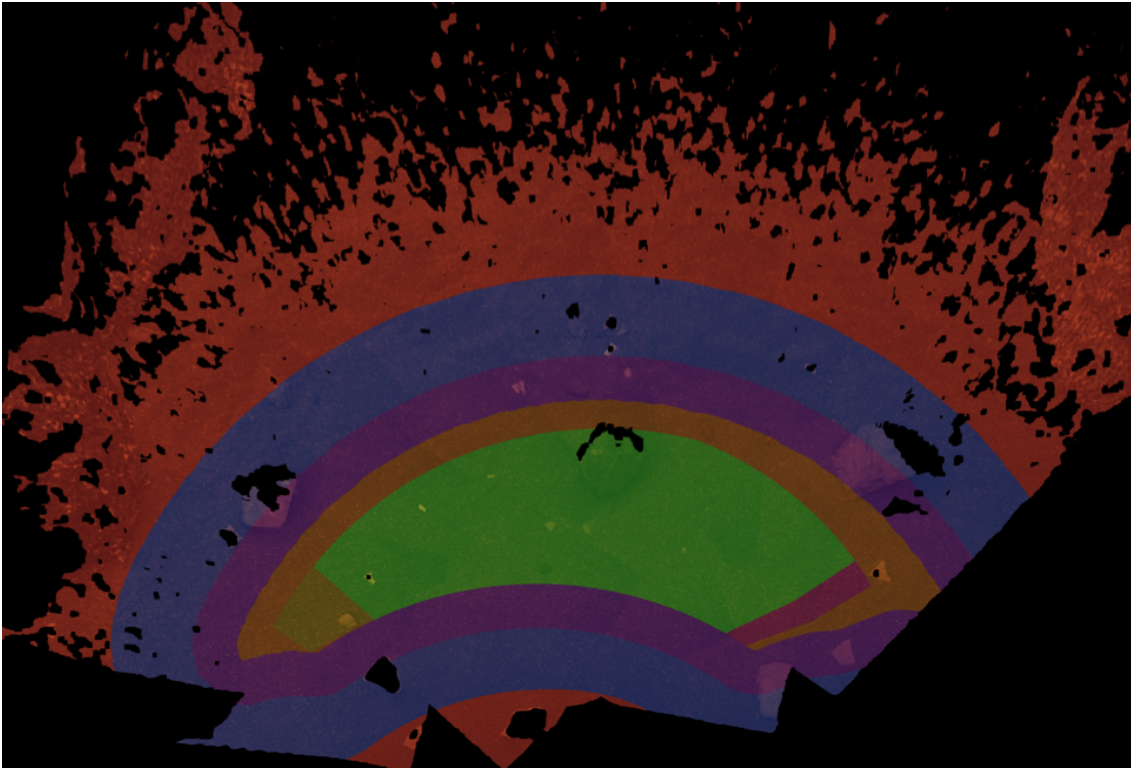


Figure 5-27: WSS Orthorectified Mosaic Overlayed onto Orthorectified Mosaic. Green, yellow, an magenta indicate deployable area; see Table 5-2 for meanings of colors.

5.4.15 Mosaic Index Files (IDX, ICM)

On occasion, mosaic index files (IDX, ICM) will be created to go along with mosaics. These files are the same size and geometry as the corresponding mosaic. Together, they indicate where each pixel came from in the source images.

IDX (index) files show which image the pixel came from. The value is an integer, 1-n, which corresponds to the position of the image in the input .LIS file (and also the position in the INPUT_PRODUCT_ID label). If two images overlap, the one “on top” (the mosaic programs do not generally blend images) is the one shown in the index file.

ICM (coregistration map) files show which pixel in that image the input came from. Like a disparity map, the two bands indicate the line and sample (respectively) coordinates of the input pixel. The coordinates are floating-point values, because the output pixels rarely map to integer input coordinates. The pixel value actually used depends on the mosaic algorithm, but is generally bilinear interpolation.

5.5 Browse Images

Although the VICAR format used to deliver images is simple, it is not supported by current-generation Web browsers. In order to make the data more accessible, “browse” images are created for some data products. These browse images are in PNG (Portable Network Graphics) format, a popular and standard format for images on the Web. Importantly, PNG files are losslessly compressed.

Browse images can be produced from any of the product types that are actual images (non-byte images are converted to byte first). Non-image types, such as XYZ, are not supported by PNG, but a browse image could be made from an overlay in a manner similar to marsviewer. A browse image for any type may thus appear in the archive. However, browse images are systematically produced from only a subset of image types. As of this writing, browse images are made from EDRs and RDRs whose DERIVED_IMAGE_TYPE is "IMAGE" (except for "CNF" types which are not in an RGB-type colorspace and are thus not amenable to being made into browse images). This includes image mosaics.

Browse images are not considered science data (Observational products in PDS). Rather, they are Supplemental products, providing alternative ways of looking at the data. The filename of a browse image is identical to the Observational product from which it is derived, except for ".PNG" rather than ".VIC" as the file extension. The label of a browse product refers to the Observational product from which it is derived, and vice-versa.

6. STANDARDS USED IN GENERATING PRODUCTS

6.1 PDS Standards

The camera instrument EDR data product complies with Planetary Data System 4 standards for file formats and labels, as specified in the PDS4 Standards Reference [34].

6.2 Time Standards

The EDR PDS label uses keywords containing time values. Each time value standard is defined according to the keyword description. See Appendix C.

6.3 Coordinate Frame Standards

This section describes the primary coordinate systems defined for surface operations, which are listed in Table 6-1 and illustrated in Figure 6-1 below.

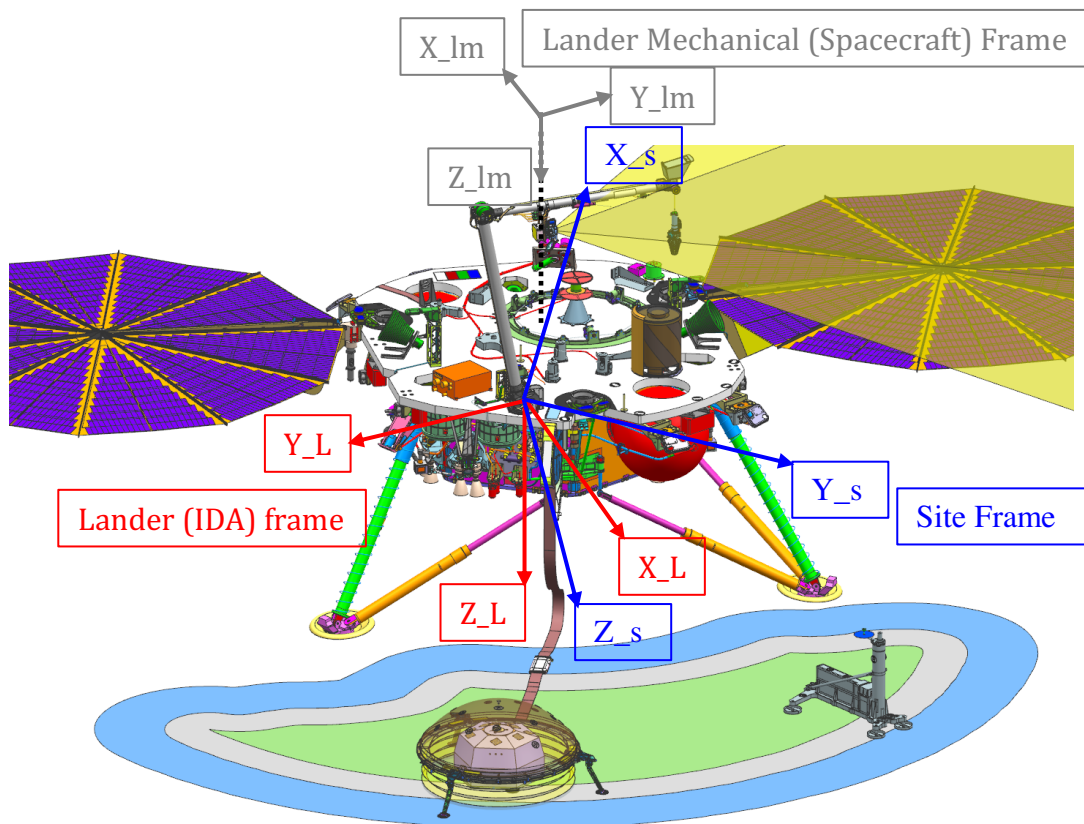


Figure 6-1: InSight Coordinate Systems. Red captions show axes for Lander frame, blue for Site frame, and gray for Lander Mechanical.

Table 6-1: Coordinate Frames Used for Surface Operations

Imaging-Related Coordinate Frames		Coordinate Frame Origin	Coordinate Frame Orientation
Name	Label Keyword Value		
Lander Frame (aka IDA frame)	“LANDER_FRAME”	Attached to Lander at intersection of IDA torso joint rotation axis (shown in Figure 6-1 as point where the red and blue arrows start)	Attached to and rotates with spacecraft. +Z axis is normal to deck surface and points from that surface downward. +X axis is perpendicular to Z axis, perpendicular to the solar array yoke, and points out into workspace +Y completes the right-handed frame.
Lander Mechanical Frame (aka Spacecraft frame) <i>NOTE: Not used in Surface Operations, for reference only</i>	Does not appear in label	Centered on launch vehicle separation plane 956.056 mm above Lander deck	Aligned with Lander. 180 degree rotation w.r.t. the Lander frame. +Z axis is normal to deck surface and points from that surface downward. +X axis is perpendicular to Z axis, perpendicular to solar array yoke and points toward deck side opposite to IDA. +Y completes the right-handed frame.
Site Frame	“SITE_FRAME”	Coincident with Lander frame (unless lander moves)	Aligned with surface. +Z axis points downward to Nadir (gravity vector). +X axis points North, perpendicular to Z axis. +Y completes the right-handed frame and points East.
Local Level Frame	“LOCAL_LEVEL_FRAME”	Coincident with Lander frame (even if lander moves)	Same as Site frame. +Z axis points downward to Nadir (gravity vector). +X axis points North, perpendicular to Z axis. +Y completes the right-handed frame and points East.
Arm Frame	ARM_FRAME	For images, attached to the camera (see text)	For images, attached to the camera (see text)

6.3.1.1 Lander (IDA) Frame

The Lander frame, also known as IDA or Payload is the one used for all surface operations and commanding. It is the only frame understood by the flight software for camera pointing and arm

commanding. It is analogous to the Rover Nav frame on MSL. The Lander frame is attached to the lander, and moves with it should the lander move while on the surface. Its origin is defined as the intersection of the RA torso joint rotation axis and the deck surface. The +Z axis points out the bottom of the spacecraft, perpendicular to the lander deck (as with all Mars surface mission frames, +Z points down). The +X axis is perpendicular to +Z (thus, parallel to the deck) and points out into the workspace. It is also perpendicular to the solar array yoke. +Y completes the right-hand frame, and thus points to the right of the workspace from the point of view of the lander.

Do not confuse Lander frame with the Lander Mechanical frame.

6.3.1.2 Lander Mechanical Frame

The Lander Mechanical (aka Spacecraft) frame is not used for any surface operations, and will not appear in the label. For reference, its origin is at $X=-0.775084$, $Y=-0.283360$, $Z=-0.956056$ meters as measured in the Lander frame (corresponding to a bracket on the launch vehicle). It is rotated 180 degrees with respect to the Lander frame.

6.3.1.3 Site Frame

The Site frame is used for geological mapping, and for instrument product placement analysis. It is coincident with the Lander frame at landing time, however +X points north, +Z points down according to the gravity vector, and +Y points east, completing the right-hand frame. The spacecraft quaternion defines the relationship between Site and Lander frames, and is determined by the spacecraft team after landing. The quaternion file is delivered in the calibration collection.

In the highly unlikely event the lander moves, the Site frame does not move; it is fixed once the lander lands.

6.3.1.4 Local Level Frames

Local Level frame is coincident with Lander frame, but points north/east/down like Site frame. The difference between Local Level and Site is that in the unlikely event the lander moves, Local Level frame moves with the lander, whereas Site does not.

6.3.1.5 Arm Frame

The Arm frame defines the location and orientation of one of the “tools” on the arm. For images, the selected tool will always be the IDC camera, thus the Arm frame in the label defines the location and orientation of the camera. This information is used to derive the IDC camera model, but the Arm frame itself is not explicitly used in operations. Figure 6-2 shows all the arm tool frames in a side view; Figure 6-3 shows them in a top view.

6.3.1.6 “Rover” Motion Counter and Quaternion

It is important to realize that Local Level and Site are purely ground constructs: the lander knows nothing of them. They exist to provide a georeferenced frame, and to accommodate the possibility of the lander moving after landing, due to settling, arm operations, or other factors. They also make certain

kinds of observations more convenient to command; thus ground tools such as RSVP and the MIPL tools should be able to convert to and from them. Local_Level and Site are indexed via a "Rover Motion Counter" (RMC) (the name is the result of MER/MSL heritage). For InSight, the RMC has only two values: Site and Position. Position is used to track small motions of the lander, while a new Site may in principle be declared to zero out those motions for ease of commanding. However, it is anticipated that no more additional Site frames will be declared subsequent to landing.

The orientation of the Local_Level and Site frames is derived from on-board sensors and from analysis of terrain surrounding the lander (compared to orbital images). This is encapsulated in a quaternion, which captures the rotation between Lander and Site frames. This quaternion is supplied by the guidance team shortly after landing, and appears in the Lander frame definition in the label. There is a possibility that early images may have an incorrect quaternion, but we expect to reprocess these with the correct quaternion once it is available.

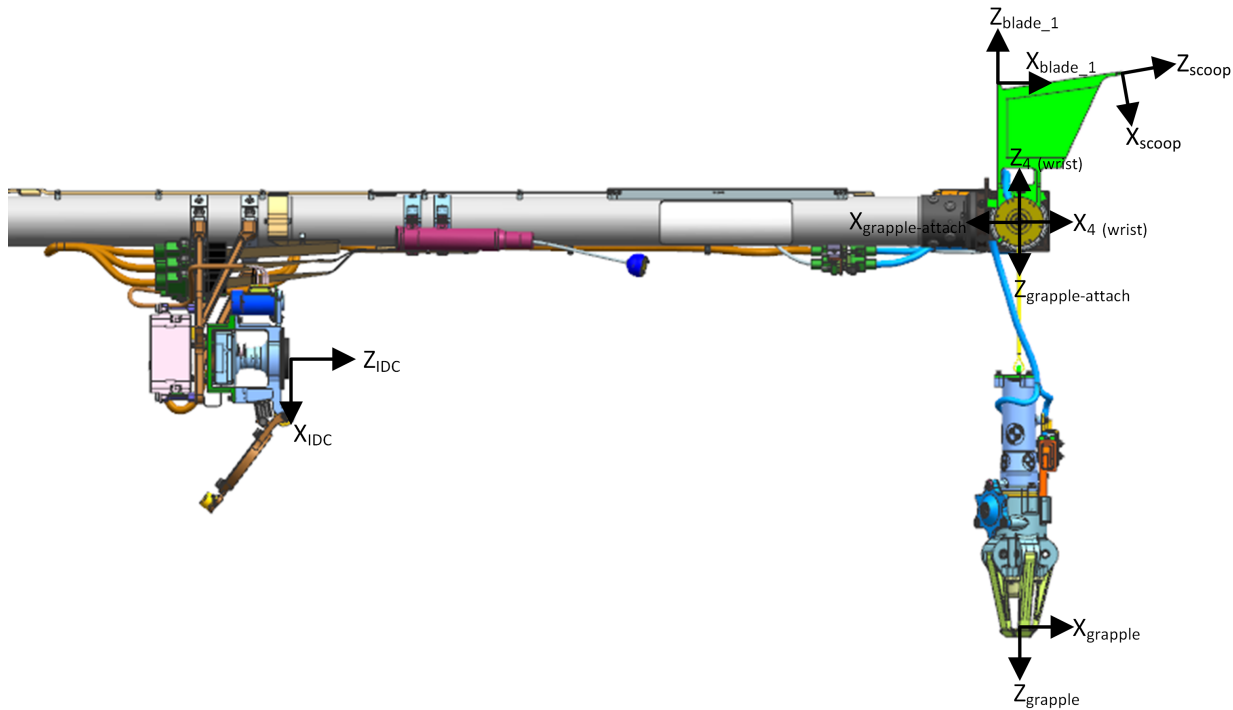


Figure 6-2: Arm coordinate systems (side view)

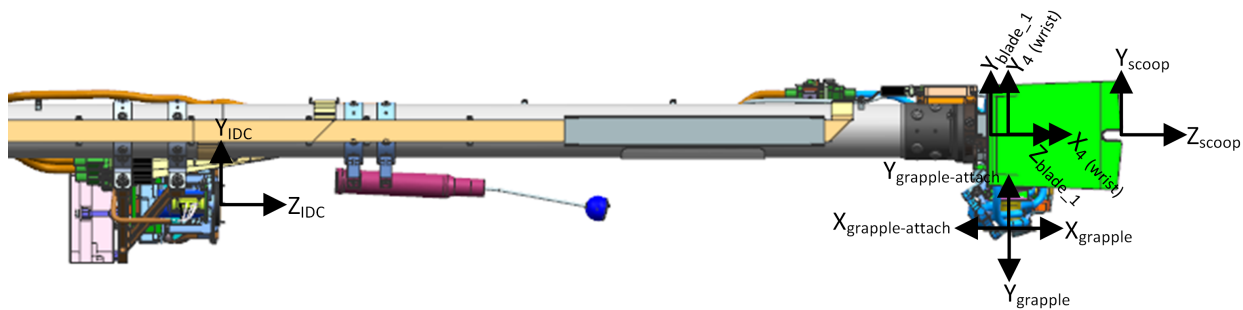


Figure 6-3: Arm coordinate systems (top view)

7. APPLICABLE SOFTWARE

Camera instrument downlink processing software is focused on rapid reduction, calibration, and visualization of images in order to make discoveries, to accurately and expeditiously characterize the geologic environment around the Lander, and to provide timely input for operational decisions concerning arm navigation and instrument deployment.

7.1 Data Generation Programs

Table 7-1 lists (in no particular order) the primary software tools that will be used to process and manipulate downlinked camera instrument imaging data. It lists the primary programs but is not exhaustive.

Table 7-1: Key Software Tools for Camera Payload Downlink Processing

Name	Description	Primary Development Responsibility
TELEMPROC	Fetches the image Standard Formatted Data Unit (SFDU) records from instrument telemetry packets, reconstructing the image data into a PDS-labelled image EDR data product. VICAR code.	Cecilia Cheng, Galen Hollins, Payam Zamani (JPL / MIPL)
Mars Program Suite	<p>Creates image RDRs and mosaics. Stereo analysis, image mosaiciking, 3D terrain building, calibration. VICAR code.</p> <ul style="list-style-type: none"> • MARSCAHV – Generates a geometrically corrected version of the EDR, applying the C, A, H and V camera model vectors. • MARSRAD – Generates a radiometrically corrected image from a single input EDR. • MARSJPLSTEREO – Generates a disparity map from a stereo pair of input EDRs, applying a 1-D correlator (fast). • MARSCOR3 – Generates a disparity map from a stereo pair of input EDRs, applying a 2-D correlator (more robust). • MARSXYZ – Generates an XYZ image from an input disparity map. • MARSRANGE – Generates a Range image from an input XYZ map. • MARSINVERTER – Generates inverse lookup table (ILUT) products. • MARSDEBAYER – Generates de-Bayered images. • MARSERROR – Generates XYZ and range error maps. • MARSUVW - Generates a Surface Normal image, wherein XYZ is computed normal to the surface. • MARSSLOPE – Generates slope maps. • MARSMASK – Generates image mask files. • MARSDISPCOMPARE – Checks consistency for left->right and right->left stereo image correlations. • MARSRELABEL – Updates camera models or coordinate systems in a label. • MARSDISPINVERT – Does L->R to R->L inversion of a disparity map. • MARSUNLINEARIZE – Converts a disparity map from linearized to unlinearized geometry. • MARSMAP – Generates a Cylindrical, Polar or Vertical projection mosaic from a list of input EDRs. • MARSMOS – Produces pinhole camera mosaics using uncorrected input images and CAHVOR camera model. 	Bob Deen (JPL / MIPL)

Name	Description	Primary Development Responsibility
	<ul style="list-style-type: none"> • MARSMCAULEY – Generates a combination Cylindrical-Perspective projection mosaic from a list of input EDRs. • MARSORTHO – Generates orthorectified mosaics. • MARSTIE – Generates pointing corrections (tiepoint file) from an overlapping set of input EDRs. • MARSAUTOTIE – Automatic tiepoint generation. • MARSNAV – Corrects pointing of images, producing a nav file. • MARSNAV2 – Uses different algorithm to correct image pointing. • MARSBRT – Generates brightness/contrast correction for mosaics. • MARSCOLOR – Does color space conversions • GAMMA – Gamma-encodes (or decodes) an image • NSYTFOOT – Generates instrument footprints. • NSYTGOOD – Generates goodness maps • NSYTROUGH – Generates instrument placement roughness. • NSYTTILT – Generates instrument tilt maps. • NSYTWKSP – Generates workspace reachability products. • xvd – VICAR image display program 	
CRUMBS	<ul style="list-style-type: none"> • CRUMBS – 3-D terrain building software 	Oleg Pariser (JPL / MIPL)

7.2 Data Viewing and Conversion Software

Table 7-2 lists software tools used by the operations team to view InSight image data, and the status of the software (not all is available to the public at this time).

Table 7-2: Software Used to View Data

Name	Description	Availability
Marsviewer	Java-based tool to view EDRs and RDRs, with appropriate visualization for all RDR types	Available from PDS
Web Marsviewer	JavaScript-based version of Marsviewer, suitable to run in browsers. Includes tools to assist with instrument placement.	Not at this writing; likely to become available from PDS in the future
WebGIS	JPL-developed web-based Geographic Information System tool used to view workspace mosaics and visualize instrument placement.	Not at this writing; may become available from PDS in the future
RSVP	Robot Sequencing and Visualization Program. Internal JPL tool to create command sequences and visualize how they interact with the terrain (via Inventor-format terrain meshes)	Not available to public
Meshlab	Open-source, standard tool to view 3D meshes. Useful for viewing OBJ format meshes.	Widely available open source meshlab.net
xvd	VICAR image display program.	Available as part of VICAR open source www-mipl.jpl.nasa.gov/vicar_open.html
Transcoder	Converts images from one format to another while preserving metadata.	Available as part of VICAR open source www-mipl.jpl.nasa.gov/vicar_open.html
Transform	PDS tool to convert data into other common formats.	Available from PDS https://pds.jpl.nasa.gov/tools/about/transform

7.3 Applicable PDS Software Tools

InSight images can be viewed using Marsviewer, a display tool developed by JPL/MIPL. It is available from the PDS Imaging Node (<https://pds-imaging.jpl.nasa.gov>). Images can also be viewed in any other PDS4 or VICAR compatible image viewer.

VICAR itself is available in open-source form, from JPL (https://www-mipl.jpl.nasa.gov/vicar_open.html), or from the PDS Imaging Node. As of this writing, the Mars-specific programs are not included in the open source release, although that may change in the future.

7.4 Software Distribution and Update Procedures

The Mars Image Processing Program Suite can be made available to researchers and academic institutions. Refer to the MIPL Web site at <http://www-mipl.jpl.nasa.gov> for contact information. As of this writing, the Mars suite is not available to the general public, although this may change in the future; see the website for current status.

APPENDIX A. DETAILED STRUCTURE OF THE IMAGE_ID FIELD IN IDPH

The image_id field is a 32-bit value that serves as a round-trip token. It is assigned by ground tools as part of the command, and appears unmodified in the returned telemetry header (IDPH). The intent behind InSight’s usage of the image_id is to provide information to guide automated ground processing of the image. One of the most important uses is to group stereo frames (see Section 5.2.3.1.1) but it also commands automated meshes and mosaics.

Field	Sequence ID	Mesh ID	Mosaic ID	Stereo ID	Eye
Bit	31-20	19-14	13-8	7-1	0

SEQUENCE ID: Context specific, Defined by uplink team
MESH ID: 0= no mesh, 1-9 = reusable meshes ID that applies to observation SOL, 10-63=unique across a range of SOLs
MOSAIC ID: 0= no mosaic, 1-9 = reusable mosaic ID that applies to observation SOL, 10-63=unique across a range of SOLs
STEREO ID: 0= mono, 1-127 = unique stereo pairs within observation SOL
EYE: 0 = left, 1= right, 0 = mono if STEREO ID =0

Notes:

1	Mosaic ID and Mesh ID values in the high ranges are allowed to roll over during the mission. When that happens, use of a configuration file will define the SOL epoch boundaries. Automated multi-sol products are not allowed to cross epoch boundaries.
2	Observation SOL is the day number in which the image was taken.
3	The upper range of the reusable ID for mesh and mosaic may not remain at "9". Therefore it should not be hard coded
4	Matching algorithm for stereo matching should use Eye, Stereo ID, Mesh ID, Sequence ID and SOL
5	A stereo pair must match all of: STEREO ID, SEQUENCE ID, MESH ID, and SOL. MOSAIC ID is not included in the match.
6	The SEQUENCE ID must match (in addition to MESH/MOSAIC ID) in order to be included in a mosaic or mesh
7	MOSAIC/MESH IS cannot roll over; if more IDs are needed use a different SEQUENCE ID.
8	Sequence IDs containing alpha characters are allowed; they represent ground-assigned values
9	Stereo IDs containing alpha characters are allowed; they represent ground-assigned values.
10	Both ground-assigned value case are for RDRs only; the SCLK can be used to trace back to the original image.
11	All 5 subfields (sequence, mesh, mosaic, stereo, eye) will appear as independent label values (including ground-assigned values).
12	The raw IMAGE_ID number will also appear as a label value (not modified by ground-assigned values).

APPENDIX B. CALIBRATION FILES

This Appendix describes the calibration files that are included with the PDS delivery.

Because the format of calibration files is basically arbitrary, they are delivered in the format needed by the Mars program suite (Table 7-1) that generates the bulk of these products. If and when the Mars program suite is made publicly available, these files will be usable with it as-is. The information in the files is useful outside of the Mars program suite context, as they provide information necessary to reproduce or validate the processing in the archive.

From the PDS point of view, they are all plain text supplemental files or (in some cases) images. Some of the text files are in XML format, but they are treated as plain text for PDS purposes. Because PDS reserves the use of the extension “.xml” for label files, the two types of files that should be named “.xml” are instead named “.xmlx” (this is noted again in the descriptions below). These need to be renamed to “.xml” for use with the Mars program suite.

The PDS labels on these files are minimal. Since they effectively *are* metadata, there is little additional metadata that would be useful in the label. All label files have the same basename (including extension) as the file they point to, but have “_lbl.xml” appended to the filename. The “_lbl” clearly distinguishes them from the “.xmlx” files mentioned above. All files with a “_lbl.xml” extension are PDS4 label files, and can be ignored if using these files with the Mars program suite.

The calibration files are organized into five directories. An overview of each type of file is described below.

Many of the filenames start with a “venue”. This venue is either “NSYT”, for the flight hardware, or “NSYTTBC”, for the (color) testbed at JPL. (There is also a “NSYTTB” for an obsolete testbed using non-color cameras; these files are not included but the name may appear in comments). As of this writing, the testbed data is not archived, but the calibration data is supplied anyway, in case there is a desire to archive testbed data in the future. Some of the filenames start with “NSYT” as a mission name rather than a venue; these are generally distinguished by the lack of a corresponding “NSYTTBC” file. The inclusion of the mission name allows the multimission Mars software suite to work with data from any mission without collision among calibration files.

In many cases, there are two copies of the cal files. One is the original file as supplied by the camera calibration engineer (J. Maki). The other is a copy of the file, with the filename needed for use with the Mars software suite. In the operations system, the “copy” is actually a softlink to the original file. Because PDS does not support softlinks, it is a copy in the archive. These are noted as “original” and “softlink” files in the writeups below.

It should be noted that the basic layout of these files is the same for all users of the Mars suite: MER, MSL, Phoenix, and Mars 2020. The details differ a little, however.

Top-Level Directory

README.TXT: This file contains the history of changes to the calibration files, as well as current status. It is included mostly for historical interest.

camera_models Directory

`NSYT_SN_xxxx.cahvor{e}`: These files contain the calibration camera models for each camera. The calibration camera model is transformed by the inverse of the calibration pose and then again by the actual pose of the arm in order to create the camera model for any given image. These are “softlink” files (see Appendix introduction). The SN (serial number) relates to the `camera_mapping` file (described below). The actual camera models are the C,A,H,V,O,R, and optionally E vectors described in the file, and are the only part of the file that is actually used. The rest of the values are covariance matrices, residuals, and other information about the camera model, that is output by the camera calibration software developed by Todd Litwin at JPL.

`INSIGHT_CAL_* .cahvor{e}`: These are “original” camera model files (see Appendix introduction). The `CAL_nnn` allows for multiple recalibrations.

flat_fields Directory

`NSYT_FLAT_SN_xxxx.IMG`: These are flat field images for each camera. The serial number relates to the `camera_mapping` file. The image is divided by the flat field in order to remove flat field effects. These are float-format images, with an average value near 1.0. They are in the same orientation as the EDR and all the RDRs (i.e. the 90-degree rotation (Section 5.1.1) has been applied). These are “softlink” files (see Appendix introduction).

Others: Other files in this directory are the “original” flat field files (see Appendix introduction).

ilut Directory

`venue_LUTn.txt`: These are the inverse lookup table (ilut) files needed to create the ILT products (Section 5.2.2.1). The format is simple: each line has “input_DN output_DN” which converts from 0-255 to 0-4095. The filename convention supports multiple iluts, but only one is expected to be used for InSight. These are “softlink” files (see Appendix introduction).

`insight_ilut.txt`: This is the “original” ilut file (see Appendix introduction).

param_files Directory

`venue_arm_mask.VIC`: This is an image file used to mask off the arm for the various masked RDRs. Any non-0 value is masked off.

`venue_camera_mapping.xmlx`: This file contains information for each camera on the venue. The information includes names and ID’s, serial number, filter status, camera model type, nominal size, and color status. The serial number is used as a lookup key in several other calibration files. These files need to be renamed with a “.xml” extension if they are to be used with the Mars software suite.

`venue_color.parms`: This file contains parameters describing what was done onboard to the image (preboost, color matrix, and companding (lookup) table). These values are not included in

telemetry, so this file lists the values. They are indexed by `sclk` (spacecraft clock) to accommodate changes over time, although no changes in these values are anticipated.

`venue_idc.point`: This file contains pointing parameters for the IDC. The calibration position and quaternion define the pose of the arm when the camera models were created. The calibration model is transformed by the inverse of this before being transformed using the actual arm pose for a given image. The pointing error values are used by the pointing correction mechanism (Section 5.4.2) as the initial perturbation for each parameter.

`venue_wksp_*.txt`: These files define the polygons for the workspace reachability product (Section 5.2.4.2). Each line defines an XYZ coordinate in Lander frame around the edge of the polygon (Z is ignored). The nominal files assume no lander tilt. After landing, if there is significant tilt, the files will be replaced based on the actual tilt. Tilt matters because the grapple hangs with gravity. These are “softlink” files (see Appendix introduction).

`NSYT_flat_fields.parms`: This file defines the calibration coefficients for each camera. The responsivity values are used to convert DN/sec (after dividing by exposure time) to radiance units for radiometrically corrected products (Section 5.2.2.2); the units are thus $W/m^2/nm/steradian/(DN/sec)$. The color conversion matrices are used to make corrected color products (Section 5.2.2.2.2).

`hp3_*.txt, seis_*.txt, seis_wts*.txt wts_*.txt`: These are the “original” workspace polygon files (see Appendix introduction).

`make_poly_links.csh`: This is a Unix C-shell script to create softlinks for a given set of polygon files. It could be used to re-create softlinks if desired, but is not otherwise needed.

rmc Directory

`venue_rmc_file.xmlx`: This file contains the spacecraft quaternion, which defines its orientation and tilt. These values are assigned after landing. These files need to be renamed with a “.xml” extension if they are to be used with the Mars software suite. The name “rmc” means Rover Motion Counter, which is heritage from other Mars missions.

APPENDIX C. LABEL KEYWORDS

A table listing the PDS4 and VICAR label attributes (keywords) and classes is available in two separate files within the document collection:

pds4_attribute_definitions_sort_pds.pdf
pds4_attribute_definitions_sort_vicar.pdf

The two files have the same contents, but the first is sorted by PDS attribute or class, while the second is sorted by VICAR keyword.

An additional two files are the same tables in HTML format.

pds4_attribute_definitions_sort_pds.html
pds4_attribute_definitions_sort_vicar.html

These tables are created by examining a set of sample labels (incorporating all types of products being created) in order to determine the PDS classes and attributes that are used by InSight. This list is then cross-referenced against the PDS data dictionaries and property maps to find the definitions, children, valid values, and data types.

The first column contains the name of the PDS attribute (“keyword”) or class (container), and the PDS dictionary it comes from. Along with that is the VICAR keyword and property name (section of the label) from which the values are derived. Not every entry has a VICAR keyword; some entries are merely containers, others contain constants or values that are derived in other ways.

The definition is in the second column. There are two components to many definitions. The first, which is always present, is the standard PDS definition that applies to all missions, from the PDS data dictionary. The second (in italics) is an InSight-specific nuance to the definition, providing additional context that applies specifically to InSight.

The third column is broken up into several pieces. The first is the XPath. This gives the “path” of where the item can be found in the label, tracing the hierarchy from the root (often but not always Product_Observational) down to the item itself. Each level in this hierarchy is a hyperlink, which can be clicked on to go directly to that item’s definition.

Underneath the XPath is a field whose content varies based on the type. For attributes (“keywords”), this contains the valid values, when such are defined by the PDS dictionary (InSight-specific valids may not be listed here; see the InSight definition nuance). For classes (containers), the valid children are listed. Those that are blue hyperlinks are actually used by InSight; clicking on them will go to that item’s definition. Those that are not blue are defined by the PDS dictionary but are not used by InSight.

Finally, also underneath the XPath field is another column containing (for attributes only) the data type and units. All attributes should have a data type, but only a few have units defined.

This label table is the primary source of information regarding the metadata in the labels. The rest of this document describes things at a high level; the label table defines specifically what each label item means.

The table can be used in either direction. Given a label item you don't understand, you can look it up in the table (sorted either by PDS or VICAR name, depending on which you're looking at) to find the definition. In some cases you may need to go up the hierarchy to find a meaningful definition (for example the definition for "x" is not particularly useful, but the parent or grandparent should describe what the full x,y,z, value is being used for). Alternatively, given an item in the table, you can find it in the label by following the XPath – looking down the hierarchy of elements until you find the item. Note that not all keywords are in any given label; the table encompasses images products, browse images, mosaics, meshes, and calibration files.

The cross-reference between PDS name and VICAR keyword can also be useful for comparing values across similar missions (MSL, MER, Phoenix, etc) that use PDS3 (the VICAR and PDS3 keywords are generally the same).