Mars 2020 (M2020)

Software Interface Specification

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

Mission: M2020 Date: September 1, 2020

Module ID: D-99960

Module Type (REFerence Only or MISsion-specific info included): MIS

Reference Module ID: N/A Date: N/A

Signatures

GDS Generating Elements:

Instrument Data Subsystem (IDS) Stirling Algermissen	Subsystem Lead Date

Science Receiving Elements:

M2020 Deputy Project Scientist Ken Williford	Deputy Project Scientist Date

GDS Receiving Elements:

M2020 GDS System Manager Maher Hanna	System Manager Date
M2020 GDS System Engineer Guy Pyrzak	System Engineer Date

Concurrence:

M2020 Imaging Investigation Scientist Justin Maki	Investigation Scientist Date
M2020 Mastcam-Z Investigation Scientist Justin Maki	Investigation Scientist Date
M2020 SuperCam Investigation Scientist TBD	Instrument Scientist Date
M2020 SuperCam Instrument Lead Dorothea Marcia Delapp	Instrument Lead. Date
M2020 MEDA Investigation Scientist Manuel de la Torre Juarez	Instrument Scientist Date
M2020 SHERLOC Investigation Scientist Kevin Hand	Instrument Scientist Date

This document has been reviewed and determined not to contain export controlled technical data.

M2020 PIXL Investigation Scientist
Yang Liu

____
Instrument Scientist Date

PDS Program Manager
Tim McClanahan

____
Manager Date

PDS Geosciences Node Manager
Ray Arvidson

____
Manager Date

PDS Discipline Node Manager
Lisa Gaddis

____
Manager Date

This document has been reviewed and determined not to contain export controlled technical data.

Mars 2020 Project

Software Interface Specification (SIS)

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

Version 1.0

Prepared by:
Nicholas A. Ruoff
Robert G. Deen
Oleg Pariser

Custodian:
Oleg Pariser

JPL D-99960

September 1, 2020

Jet Propulsion Laboratory
California Institute of Technology

This document has been reviewed and determined not to contain export controlled technical data.

CHANGE LOG

DATE	SECTIONS CHANGED	REVISION
2019-11-17	Appendix B – updated for new IDPH (IMF) for all instruments and compression (JPEG) in FSW S6.0.1. Updated keyword sources for ZCAM and SHERLOC. Added new "FOCUS" keywords for ZCAM and SHERLOC.	0.7
2019-11-19	Appendix B – added ZCAM and SHERLOC valid values for COMMAND_INSTRUMENT_ID. Added PIXL Mini-Header Keywords	0.7
2020-02-20	Appendix B – updates for all instruments.	0.8
2020-04-30	Updated all tables and figures.	0.8
2020-05-04	Appendix B – updated keyword sources for PIXL, SCAM, ZCAM, and MEDA.	0.9
2020-05-06	Extensive reformatting. Consolidating of EDR info that spans all instruments.	0.9
2020-05-10	Appendix B – updates for telemetry sources. Add SHERLOC Engineering products (pass-through).	0.9
2020-5-15	All sections	0.9
2020-05-24	Appendix B - removed HDR group and placed HDR_* keywords back into OBSERVATION_REQUEST_PARMS. ACTIVITY_ID replaced keyword REQUEST_ID for all instruments.	1.0
2020-05-25	Updated "camera specific" portion of filename for SuperCam to reflect the "point number" of the RMI image. Updated time fields in filename for ZCAM/SHERLOC video. Added "RTT" and "PMC" to the RMC for PIXL. Updated the "Tool" CS group to reflect the name of the tool being used. Keyword OFFSET_MODE_ID changed to DC_OFFSET for all instruments.	1.0
2020-09-01	Reformated Content Table Updated Signatorees list Added Tables and Figures Incorporated internal reviewers inputs Added Appendix C	1.0

OPEN ISSUE ITEMS

REVISION	OPEN ISSUE	CLOSED

Table of Contents

CHANGE LOG I

OPEN ISSUE ITEMS II

ACRONYMS AND ABBREVIATIONS VIII

1. INTRODUCTION 12

1.1 PURPOSE AND SCOPE 12
1.2 CONTENTS 13
1.3 CONSTRAINTS AND APPLICABLE DOCUMENTS 13
1.3.1 RELATIONSHIPS WITH OTHER INTERFACES 15

2. INSTRUMENT OVERVIEW 16

2.1 ENGINEERING CAMERA INSTRUMENT SUITE 19
2.1.1 HAZARD AVOIDANCE CAMERA (HAZCAM) 20
2.1.2 NAVIGATION CAMERA (NAVCAM) 22
2.1.3 SAMPLE CACHING SYSTEM CAMERA (CACHECAM) 23
2.2 SUPERCAM REMOTE MICRO-IMAGER (RMI) 25
2.3 MASTCAM-Z CAMERA INSTRUMENT SUITE 26
2.4 SHERLOC CAMERA SUITE 29
2.4.1 SHERLOC-WATSON 29
2.4.2 AUTOFOCUS AND CONTEXT IMAGER (ACI) 30
2.5 PIXL MICRO CONTEXT CAMERA (MCC) 31
2.6 MEDA SKY CAMERA 32
2.7 EDL CAMERA SUITE 33
2.7.1 PARACHUTE UPLOOK CAMERAS (PUC) 34
2.7.2 DESCENT STAGE DOWNLOOK CAMERA (DDC) 34
2.7.3 ROVER UPLOOK CAMERA (RUC) 34
2.7.4 ROVER DOWNLOOK CAMERA (RDC) 34
2.7.5 EDL MICROPHONE 34
2.8 LANDER VISUSAL SYSTEM CAMERA (LCAM) 35
2.9 HELICOPTER IMAGING SUITE 35
2.9.1 HELICOPTER NAVIGATION CAMERA 35
2.9.2 HELICOPTER RETURN TO EARTH CAMERA 35
2.10 CAMERA FLIGHT SOFTWARE PROCESSING 36
2.10.1 JPEG FOR ECAM AND SUPERCAM 36
2.10.2 COLOR ICER FOR SUPERCAM 37
2.10.3 COMPRESSION FOR MEDA 37

3. GENERAL DATA PRODUCT OVERVIEW 38

3.1 DATA PROCESSING LEVELS 38

3.2 PRODUCT LABEL AND HEADER DESCRIPTIONS 39
3.2.1 OVERVIEW OF LABELS 39
3.2.2 ODL LABEL 39
3.2.3 PDS4 LABEL 42
3.2.4 VICAR LABEL 42
3.2.5 MAPPING OF ODL AND VICAR LABELS 43
3.2.6 MAPPING OF PDS4 AND VICAR LABELS 45
3.2.7 SUPERCAM FITS FILES 45
3.3 BINARY DATA STORAGE CONVENTIONS 45
3.3.1 BIT AND BYTE ORDERING 45

4. EDR PRODUCT SPECIFICATION 46

4.1 EDR GENERAL PROCESSING 46
4.2 EDR PRODUCT TYPES 48
4.2.1 IMAGE EDR 48
4.2.2 IMAGE SUPPORT EDRS 51
4.2.3 ENGINEERING EDRS 51
4.2.4 OTHER EDRS 53
4.3 EDR PRODUCT FORMAT 53
4.3.1 ENGINEERING CAMERA INSTRUMENT SUITE 55
4.3.2 SUPERCAM RMI EDR 57
4.3.3 MASTCAM-Z (ZCAM) 57
4.3.4 SHERLOC CAMERA INSTRUMENT SUITE 58
4.3.5 PIXL MCC 58
4.3.6 MEDA SKYCAM 59
4.3.7 EDL CAMERA SUITE 59
4.3.8 LVS 59
4.3.9 HELICOPTER 59
4.4 EDR PRODUCT STRUCTURE 59
4.4.1 IMAGE EDR 61
4.4.2 IMAGE SUPPORT EDR 62
4.4.3 ENGINEERING EDRS 62
4.5 EDR PRODUCT VALIDATION 63

5. FDR PRODUCT SPECIFICATION ***NOT PART OF PEER REVIEW* 65**

5.1 FDR GENERAL PROCESSING 65
5.2 ECAM FDR PROCESSING (RECONSTRUCTED IMAGES) 65
5.2.1 TILE RECONSTRUCTION TYPES 65
5.2.2 ECAM FUNDAMENTAL TILE DATA RECORD (TDR) 65

6. RDR PRODUCT SPECIFICATION * NOT PART OF PEER REVIEW *** 67**

6.1 RDR GENERAL PROCESSING 67
6.1.1 IMAGE RDRS 67
6.2 RDR PRODUCT TYPES 71
6.2.1 IMAGE RDRS 71

6.3 RDR PRODUCT FORMAT **101**
6.3.1 IMAGE RDRS 101
6.4 RDR PRODUCT STRUCTURE **105**

7. STANDARDS USED IN GENERATING PRODUCTS **107**

7.1 FILE NAMING STANDARDS **107**
7.1.1 EDR/RDR SINGLE-FRAME FILENAME 107
7.1.2 MOSAIC RDR FILENAME 120
7.1.3 UNIFIED TERRAIN MESH RDR FILENAME 127
7.2 PDS STANDARDS **134**
7.3 TIME STANDARDS **134**
7.4 COORDINATE FRAME STANDARDS **134**
7.4.1 ROVER NAVIGATION (ROVER NAV) FRAME 135
7.4.2 ROVER MECHANICAL (ROVER MECH) FRAME 136
7.4.3 LOCAL LEVEL FRAME 136
7.4.4 SITE FRAME 137
7.4.5 RSM FRAME 138
7.4.6 ARM FRAMES 138

8. APPLICABLE SOFTWARE **138**

8.1 UTILITY PROGRAMS **138**
8.2 APPLICABLE PDS SOFTWARE TOOLS **140**
8.3 SOFTWARE DISTRIBUTION AND UPDATE PROCEDURES **140**

APPENDIX A. DATA PRODUCT AND DATA PRODUCT OBJECT NAMES **141**

APPENDIX B. PRODUCT LABEL KEYWORD DEFINITIONS, VALUES, AND SOURCES **146**

APPENDIX C. CALIBRATION PARAMETERS **148**

LIST OF FIGURES

Figure 2-1 – Example Bayer filter pattern 18
Figure 2-2 – The cameras on M2020 Rover 18
Figure 2-3 – Engineering Camera hardware design 19
Figure 2-4 – Front Hazcam locations 21
Figure 2-5 – Front Hazcam stereo pair mapping 21
Figure 2-6 – Rear Hazcam location 22
Figure 2-7 – Navcam location 23
Figure 2-8 – CacheCam location 24
Figure 2-9 – SuperCam Mast Unit 25
Figure 2-10 – Mastcam-Z Spectral Response 28
Figure 2-11– WATSON and ACI Cameras of SHERLOC instrument suite. 30
Figure 2-12 – DPU, CHU, FLI and two SLIs mounted with CHU 31
Figure 2-13 – LED distribution on the FLI and corresponding LED ID 32
Figure 2-14 – RDS-Skycam with FOV 33
Figure 2-15 – EDL Cameras overview 33
Figure 4-1 – Instrument Data Flow to IDS 47
Figure 4-2 – IDS EDR generation data flow 48
Figure 4-3 – ZCAM – RGB in Bayer Pattern Layout on CCD 58
Figure 4-4 – Dual Label EDR with detached PDS4 label 61
Figure 6-1 – VICAR image processing flow diagram 68
Figure 6-2 – FHAZ EDR (left) vs FHAZ Linearized RDR (right) 69
Figure 6-3 – RMI radiometric processing flow diagram 73
Figure 6-4 – XYZ data masked and overlaid onto image EDR 77
Figure 6-5 – Range (distance) data overlaid onto image EDR 79
Figure 6-6 – Arm reachablilty qualities overlaid onto image EDR 84
Figure 6-7 – Rover location rendered in a unified terrain mesh 87
Figure 6-8 – Arm payload location rendered in a unified terrain mesh 87
Figure 6-9 – Cylindrical projection mosaic (CYL) 93
Figure 6-10 – Camera perspective mosaic (PER) 94
Figure 6-11 – Cylindrical-perspetive projection mosaic (CYP) 96
Figure 6-12 – Polar projection mosaic (POL) 97
Figure 6-13 – Vertical projection mosaic (VRT) 98
Figure 6-14 – Orthorectified projection mosaic (ORR) 100
Figure 6-15 – RDR Product Structures (TBD) 106
Figure 7-1 – Rover navigation coordinate frame (RNAV) 135
Figure 7-2 – Yaw, Pitch and Roll Definition 136
Figure 7-3 – Site and Rover frame examples 137

LIST OF TABLES

Table 1-1 – Product and software interfaces to this SIS........ 15
Table 2-1 – Tabulation of M2020 Camera Instruments........ 17
Table 2-2 – Hazcam operational characteristics........ 20
Table 2-3 – Navcam Operational Characteristics........ 22
Table 2-4 – CacheCam Operational Characteristics........ 23
Table 2-5 – SCAM RMI Operational Characteristics........ 25
Table 2-6 – RMI Compression Modes........ 26
Table 2-7 – Mastcam-Z Operational Characteristics........ 26
Table 2-8 – Mastcam-Z Spectral Filters and Bandpass........ 27
Table 2-9– ZCAM known focal length to Zoom Motor Counts........ 28
Table 2-10 – SHERLOC-WATSON Operational Characteristics........ 29
Table 2-11 – SHERLOC-ACI Operational Characteristics........ 30
Table 2-12 – SkyCam Operational Characteristics........ 32
Table 2-13 – EDL Cameras Operational Characteristics........ 34
Table 3-1 – PDS4 Processing Levels for Instrument Experiment Data Sets........ 38
Table 3-2 – ODL Class to VICAR Property Set Mappings........ 44
Table 3-3 – M2020 Image EDR/RDR Bit Ordering........ 46
Table 4-1 – List of EDR Types and Formats........ 53
Table 4-2 – DC Offset specification for the Engineering Cameras........ 56
Table 6-1 – Default surface roughness parameters and threshholds........ 81
Table 6-2 – Arm reachability bit assignements per configuration (TBD)........ 83
Table 6-3 – Arm reachability values........ 83
Table 6-4 – Product ID color codes........ 101
Table 6-5 – M2020 camera image EDR/RDR product binary formats........ 102
Table 6-6 – M2020 camera non-image RDR binary formats........ 105
Table 7-1 – Camera instrument single-frame fileanaming convention........ 108
Table 7-2 – Camera instrument mosaic filenaming convention........ 120
Table 7-3 – Camera instrument unified terrain mesh filenaming convention........ 128
Table 7-4 – Coordinate frames used for M2020 surface operations........ 134
Table 8-1 – Key software toolsets........ 138

ACRONYMS AND ABBREVIATIONS

ACA	Adaptive Caching Assembly
ACI	Autofocus and Context Imager
APID	Application Process Identifier
APSS	Activity Planning and Sequencing Subsystem
ASCII	American Standard Code for Information Interchange
ASTTRO	Advanced Science Targeting Tool for Robotic Operations
ATLO	Assembly, Test, Launch and Operations
BITCAR	Bit-Carousel
BU	Body-Unit (SuperCam)
CAHV	Center, Axis, Horizontal, Vertical (camera model)
CAHVOR	Center, Axis, Horizontal, Vertical, Optical, Radial (camera model)
CAHVORE	Center, Axis, Horizontal, Vertical, Optical, Radial, Entrance (camera model)
CBLO	Core Bit Lock Out
CCD	Charged Coupled Device
CCBU	Chemistry Camera Body Unit
CCMU	Chemistry Camera Mast Unit
CEDL	Cruise Entry Descent and Landing
CMOS	Complementary Metal Oxide Semiconductor
SHA	Collection and Handling for Interior Martian Rock Analysis
CNES	Centre National d'Etudes spatiales (French Space Agency)
CODMAC	Committee on Data Management and Computation
CSV	Comma-separated-value
DDC	Descent Stage Downlook Camera
DEA	Digital Electronics Assembly
DEM	Digital Elevation Map
DN	Digital Number
DOY	Day of Year
DP	Data Product (telemetry)
DPO	Data Product Object
DTE	Direct to Earth
DVT	Data Validity Time
ECAM	Engineering Camera
EECAM	Enhanced Engineering Camera
EDL	Entry, Descent and Landing
EDR	Experiment Data Record
EHA	Engineering, Housekeeping & Accountability (EH&A)
EM	Engineering Model

EMD	Earth Metadata file (".emd")
EPDU	End-of-Product PDU
ERT	Earth Received Time
FDD	Functional Design Document
FDR	Fundamental Data Record
FEI	File Exchange Interface
FGICD	Flight-Ground ICD
FHAZ	Front Hazard Avoidance Camera
FM	Flight Model
FOV	Field of View
FPGA	Field Programmable Gate Array
FSW	Flight Software
GSFC	Goddard Space Flight Center
GDS	Ground Data System
GSE	Ground Support Equipment
Hazcam	Hazard Avoidance Camera
HGA	High Gain Antenna
IC	Inlets Cover
ICD	Interface Control Document
ICER	Image compression algorithm (not an acronym)
ID	Identification
IDPH	Instrument Data Product Header
IDS	Instrument Data Systems
IFOV	Instantaneous Field of View
ILUT	Inverse Lookup Table
IPE	Integrated Planning and Execution (MS element)
IRAP	Institut de Recherche en Astrophysique et Planétologie
ISIS	Integrated Software for Imagers and Spectrometers
IVP	Inertial Vector Propagation
JPEG	Joint Photographic Experts Group (compression)
JPL	Jet Propulsion Laboratory
LANL	Los Alamos National Laboratory
LIBS	Laser-Induced Breakdown Spectrometer (SuperCam)
LOCO	LOw-COmplexity, LOssless COmpression
Mastcam-Z	Mastcamera-Zoom
MCZ	Mastcam-Z (**obsolete: use ZCAM**)
MER	Mars Exploration Rover
MIC	Microphone
MIPL	Multimission Instrument Processing Laboratory

MOS	Mission Operations System
MPCS	Mission data Processing and Control Subsystem
MPDU	Metadata Protocol Data Unit
MPF	Mars Pathfinder
MS	Mission System
MU	Mast Unit (SuperCam)
M2020	Mars 2020 Rover
MSSS	Malin Space Science Systems
NASA	National Aeronautics and Space Administration
Navcam	Navigation Camera
OCS	Operations Cloud Store
ODL	Object Description Language
OPGS	Operations Products Generation Subsystem
PDC	Parachute Downlook Camera
PDS	Planetary Data System
PDU	Protocol Data Unit
PPDU	Product Data Protocol Data Unit
PRT	Platinum Resistance Thermometer
PSDD	Planetary Science Data Dictionary
PUC	Parachute Uplook Camera
RA	Robotic Arm
RCE	Rover Compute Element
RDC	Rover Downlook Camera
RDR	Reduced Data Record
RHAZ	Rear Hazard Avoidance Camera
RMI	Remote Micro-Imager (SuperCam)
ROI	Region of Interest
RPS	Rover Planning System
RSM	Remote Sensing Mast
RTO	Real Time Operations (MS element)
RUC	Rover Uplook Camera
RSVP	Rover Sequencing and Visualization Program
SAPP	Surface Attitude, Positioning and Pointing
SCID	Spacecraft ID
SCLK	Spacecraft Clock
SCM	Spacecraft Configuration Manager
SCS	Sample Cache System
SFDU	Standard Format Data Unit
SHA	Sample Handling Arm

SHERLOC	Scanning Habitable Enviroments with Raman and Luminescence for Organics and Chemicals
SIS	Software Interface Specification
SOH	State of Health (SuperCam)
SOL	Mars Solar Day
SOWG	Science Operations Working Group
SPaH	Sample Processing and Handling
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events kernels
SRAM	Static Random Access Memory
SuperCam	Super Camera Instrument
SwRI	Southwest Research Institute
TBD	To Be Determined
TBU	Testbed Unit
TDR	Tile-fundamental Data Record
TDS	Telemetry Delivery Subsystem
TRDR	Tile Reduced Data Record
UDR	Unprocessed Data Record
USGS	United States Geological Survey
VCID	Virtual Channel Identifier
VICAR	Video Image Communication and Retrieval
WATSON	Wide Angle Topographic Sensor for Operations and eNgineering
WEB	Warm Electronic Box
ZCAM	Mastcam-Z

1. INTRODUCTION

1.1 Purpose and Scope

The purpose of this Data Product Software Interface Specification (SIS) is to provide consumers of M2020 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. Content in this document supports EDR and RDR data products generated by the Instrument Data Subsystem (IDS) for the following instruments:

- Engineering Camera (ECAM) suite:
    a. Navigation Cameras (Navcams)
    b. Hazard Avoidance Cameras (Hazcams)
    c. Sample Caching System Camera (CacheCam)

- SuperCam Remote Micro-Imager (RMI)

- Mastcam-Z (ZCAM) instrument suite:
    a. Mastcam-Z Left (ZCAM-L)
    b. Mastcam-Z Right (ZCAM-R)

- SHERLOC camera suite
    a. SHERLOC-WATSON
    b. SHERLOC-ACI

- PIXL Micro Context Camera (MCC)

- MEDA Skycam

- EDL camera suite
    a. Parachute Uplook Cameras (PUC)
    b. Descent Stage Downlook Camera (DDC)
    c. Rover Uplook Camera (RUC)
    d. Rover Downlook Camera (RDC)

- Landing Visual System (LVS)

- Helicopter camera suite
    a. Helicopter Navigation Camera
    b. Helicopter Return To Earth Camera

This document covers both operations and archive use cases for Mars 2020 camera data. The users for whom this SIS is intended thus include the Mars 2020 operations, the Mars 2020 science team, and the general public via the PDS Archive.

In this document, the EDR data product is the raw, uncalibrated, uncorrected image data acquired by the M2020 instrument. It may include decompression if there was data product compression performed onboard the rover by the instrument, including decompanding, de-Bayering, and tile reassembly. 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 are derived directly from one or more image EDR or image RDR data product(s). They are comprised of radiometrically decalibrated and/or camera model corrected and/or geometrically altered (including reprojected) versions of the raw camera data, as well as higher-order processed results (e.g. XYZ points and surface normal).

1.2 Contents

This Data Product SIS describes how the EDR data products are acquired by the M2020 cameras and how it is processed, formatted, labeled, and uniquely identified, and how the image RDR data products are derived from image EDR or image 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. Processing is described at a high level, and full definitions of all metadata keywors are provided.

1.3 Constraints and Applicable Documents

This SIS is meant to be consistent with the contract negotiated between the M2020 Project and the M2020 Principal Investigators (PI) for the Engineering cameras, the SuperCam RMI, the Mastcam-Z cameras, the Sherloc-Imaging (WATSON) and ACI cameras, the PIXL Micro Context Camera, and the MEDA Sky Camera, in which reduced data records and documentation are explicitly defined as deliverable products. Because this SIS governs the specification of data products used during mission operations, any proposed changes to this SIS must be impacted by all affected software subsystems observing this SIS in support of operations (e.g., RPS, IDS, SOAS).

Product label 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 to (new) unrecognized keywords. Similarly, entirely new products may be added over time.

This Data Product SIS is responsive to the following M2020 documents:

1. Mars 2020 CEDL Engineering Camera Imaging Functional Design Description (FDD), “Baseline Release, Rev A”, Daniel Petrizzo, JPL D-95849
2. Mars 2020 Surface Engineering Camera Imaging Functional Design Description (FDD), “Baseline Release, Rev A”, Daniel Petrizzo, JPL D-96164
3. Mars 2020 Mars Helicopter System Functional Design Description (FDD), “Baseline Release, Rev A”, Daniel Petrizzo, JPL D-99937
4. Mars 2020 CEDL and Surface SuperCam Functional Design Description (FDD), “Baseline Release, Rev A”, Ivair Gontijo, JPL D-95868
5. Mars 2020 CEDL and Surface Mastcam-Z Functional Design Description (FDD), “Baseline Release, Rev A”, Zach Bailey, JPL D-95850
6. Mars 2020 CEDL Surface SHERLOC Functional Design Description (FDD), “Baseline Release, Rev A”, Zach Bailey, JPL D-95867
7. Mars 2020 CEDL and Surface MEDA Functional Design Description (FDD), “Baseline Release, Rev A”, Christina Diaz, JPL D-95851
8. Mars 2020 CEDL and Surface PIXL Functional Design Description (FDD), “Baseline Release, Rev A”, Joan Ervin, JPL D-95857
9. Mars 2020 CEDL Camera Functional Design Description (FDD), “Baseline Release, Rev A”, Danielle Nuding, JPL D-95844

10. Mars 2020 Flight-Ground Interface Control Document (FGICD), “Volume 1, Downlink, Rev A, Version 1.0”, Biren Shah, JPL D-95521, October 3, 2017

Additionally, this SIS is also consistent with the following Planetary Data System documents:

11. Planetary Science Data Dictionary Document, Version 1.81, November 24, 2010.
12. Planetary Science Data M2020 Local Data Dictionary, Version 1.0, January 15, 2013.
13. Planetary Data System Archive Preparation Guide, Version 1.4, JPL D-31224, April 1, 2010.

Finally, this SIS makes reference to the following documents for technical background information:

14. A System for Extracting Three-Dimensional Measurements from a Stereo Pair of TV Cameras, Y. Yakimovsky and R. Cunningham, January 7, 1977.
15. Camera Calibration, D. Gennery, JPL IOM 347/86/10, February 5, 1986.
16. Sensing and Perception Research for Space Telerobotics at JPL, D. Gennery et al., *Proceedings of the IEEE Intern. Conf. on Robotics and Automation,* March 31 - April 3, 1987.
17. Camera Calibration Including Lens Distortion, D. Gennery, JPL D-8580, May 31, 1991.
18. Algorithm for Using CAHV to Determine SGI Graphics Viewpoint and Perspective, B. Bon, JPL IOM 3472-91-057, August 6, 1991.
19. Inclusion of Old Internal Camera Model in New Calibration, D. Gennery, JPL IOM 386.3-94-001, February 22, 1994.
20. “Least-Squares Camera Calibration Including Lens Distortion and Automatic Editing of Calibration Points”, Calibration and Orientation of Cameras in Computer Vision, D. Gennery, ISBN 3-540-65283-3, 2001.
21. Computations for Generalized Camera Model Including Entrance, Part 1 and Part 2, D. Gennery, unpublished, May 23, 2001.
22. Generalized Camera Calibration Including Fish-Eye Lenses, D. Gennery, JPL D- 03-0869, 2002.
23. Issues with Linearization, R. Deen, JPL Docushare Collection 2700, File 75670, 2003.
24. Mastcam-Z Multispectral Imaging on the Mars Science Laboratory Rover: Wavelength Coverage and Imaging Strategies at the Gale Crater Field Site, J.F. Bell III et al., *43rd Lunar and Planetary Science Conference*, 2012.
25. Anderson, R.C., et al., Mars Science Laboratory Participating Scientists Program Proposal Information Package, December 14, 2010.
26. Deen, R.G. and J.J. Lorre (2005), Seeing in Three Dimensions: Correlation and Triangulation of Mars Exploration Rover Imagery, submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, Hawaii.
27. Mars Exploration Rover (MER) Project ICER User’s Guide, Aaron Kiely, MER 420-8-0538, JPL D-22103, January 5, 2004.
28. 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.
29. Bell III, J.F., A. Godber, S. McNair, M.C. Caplinger, J.N. Maki, M.T. Lemmon, J. Van Beek, M.C. Malin, D. Wellington, K.M. Kinch, M.B. Madsen, C. Hardgrove, M.A. Ravine, E. Jensen, D. Harker,R.B. Anderson, K.E. Herkenhoff, R.V. Morris, E. Cisneros, and R.G. Deen, The Mars Science Laboratory Curiosity rover Mast Camera (Mastcam) instruments: Pre-flight and in-flight calibration, validation, and data archiving, Earth and Space Science, 4, doi:10.1002/2016EA000219, 2017
30. Bell III, J.F., et al., The Mars 2020 Rover Mast Camera Zoom (Mastcam-Z) Multispectral, Stereoscopic Imaging Investigation, in preparation for Space Science Reviews, 2020.
31. Hayes, A.G., et al., Pre-Flight Calibration of the Mars 2020 Rover Mastcam Zoom (Mastcam-Z) Multispectral, Stereoscopic Imager, in preparation for print, 2020

32. Maki, J.N., et al., The Mars 2020 Engineering Cameras and Microphone on the Perseverance Rover: A Next-Generation Imaging System for Mars Exploration, Manuscript Draft for Space Science Reviews, 2020.

1.3.1 Relationships with Other Interfaces

Changes to this EDR/RDR data product SIS document affect the following products, software, and/or documents.

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)
M2020 Camera Instrument EDRs • Cachecam • Hazcam • Navcam • Supercam RMI • Mastcam-Z • SHERLOC-WATSON • SHERLOC-ACI • MEDA Skycam • PIXL • EDL Cam • LVS • Helicopter		P	MIPL (JPL)
RSVP		S	RSVP Dev Team (JPL)
ASTTRO		S	ASTTRO Dev Team (JPL)
RGIS		S	RGIS Dev Team (JPL)
Mars Program Suite • m2020edrgen • m20filter • m20reach • marsautoloco • marsautotie • marsautotie2 • marsbias • marsbrt • marsc2uvw • marscahv • marscheckcm • marschkovl • marscolor • marscoordtrans • marscor2 • marscor3 • marscorr • marscsv • marsdebayer • marsdepth • marsdispcompare • marsdispinvert		S	MIPL (JPL)

Name	Type
	P = product
	S = software
	D = document
			Owner

• marsdispwarp • marsecorr • marserrdisp • marserror • marsfakedisp • marsfidfinder • marsfilter • marsget_cm • marsifoot • marsigood • marsint • marsinterp • marsinverter • marsinvrange • marsirough • marsitilt • marsjplstereo • marsmake_cm • marsmap • marsmask • marsmcauley • marsmesh • marsmos • marsnav • marsnav2 • marsortho • marsproj • marsprojfid • marsrad • marsrange • marsreach • marsrefmesh • marsrelabel • marsremos • marsrescale • marsrfilt • marsrough • marsseedgen • marsslope • marstie • marstiexyz • marstile • marsunlinearize • marsunmosaic • marsuvw • marsuvwproj • marsuvwrot • marsxyz • marsxyzmerge			

2. INSTRUMENT OVERVIEW

In this section, overviews are provided for the all of M2020 camera instruments mentioned in section 1.1. The M2020 rover instrument payload includes 25 individual cameras. The main differences between these instruments are in the optics, mounted position, and articulation methods. The cameras are color, except for the SHERLOC-ACI, PIXL MCC, MEDA SkyCam, and Helicopter Navigation camera. Of the 25 cameras, there are 5 sets of stereo pairs and 15 single cameras, as listed in Table 2-1 below.

Table 2-1 – Tabulation of M2020 Camera Instruments

M2020 Camera Instrument	Location	Number
Navcam	Stereo pair on RSM	2
Front Hazcam	Stereo pair at front of WEB	4
Rear Hazcam	Stereo pair at rear of WEB	2
CacheCam	Monoscopic inside WEB	1
SuperCam RMI	Monoscopic on RSM	1
Mastcam-Z	Stereo pair on RSM	2
SHERLOC-WATSON	Monoscopic on robotic arm turret	1
SHERLOC-ACI	Monoscopic on robotic arm turret	1
MEDA SkyCam	Monoscopic on rover deck	1
PIXL MCC	Monoscopic on robotic arm turret	1
EDL PUC	Monoscopic on top of backshell	3
EDL DDC	Monoscopic on bottow of descent stage structure	1
EDL RUC	Monoscopic on top of rover deck	1
EDL RDC	Monoscopic on side of WEB	1
LVS CAM	Monoscopic on the front left corner of the rover body	1
Helicopter Navigation Cam	Downward- looking on Helicopter Body	1
Helicopter Return To Earth Cam	Side-looking on Helicopter Body	1
	Total	**25**

There are several color capable cameras onboard the rover and they all utilize a Bayer filter/pattern, consisting of repeated squares of red (R), green (G), and blue (B) filters, as shown in figure 2-1.

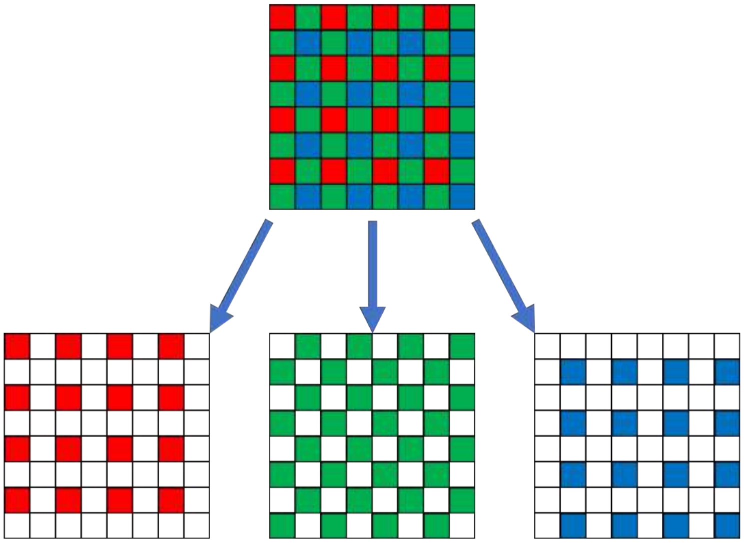


Figure 2-1 – Example Bayer filter pattern

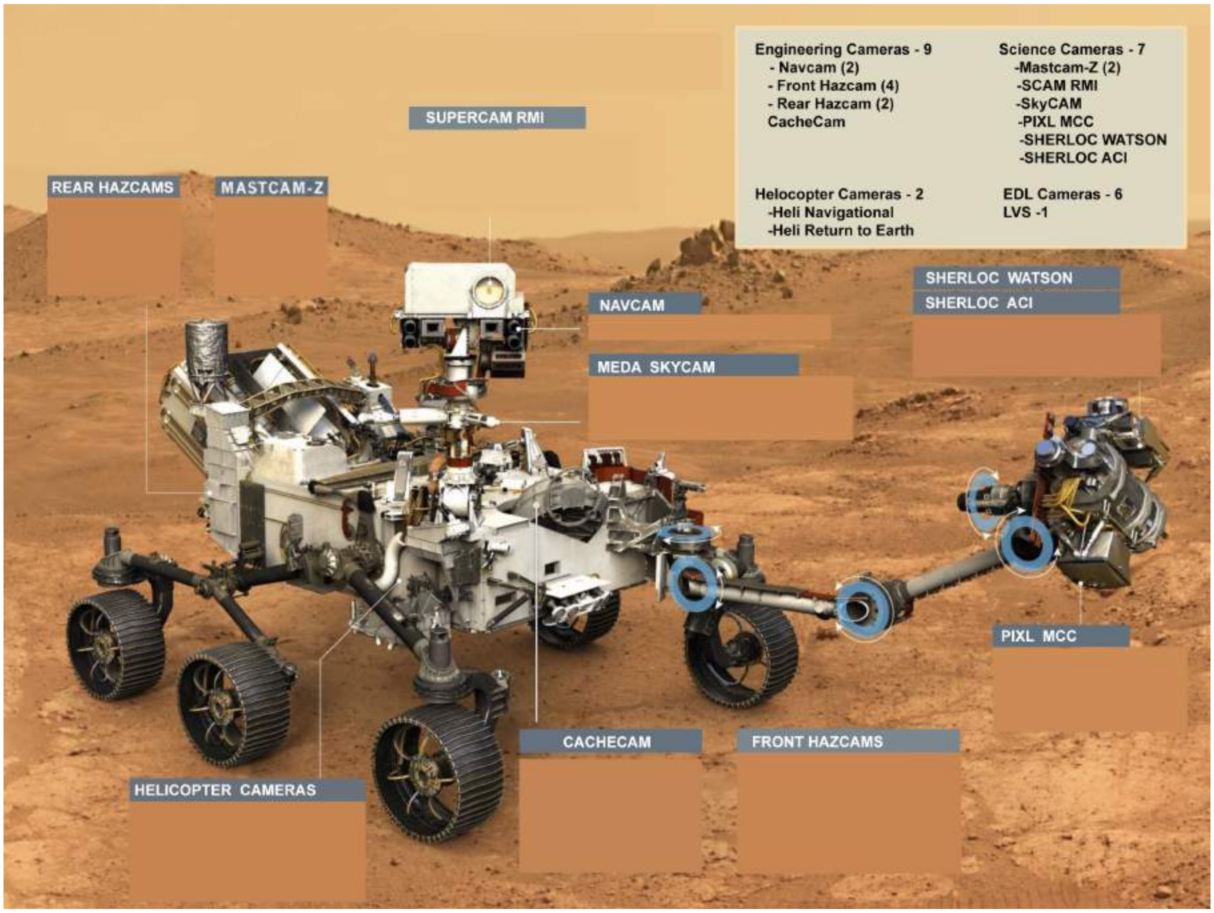


Figure 2-2 – The cameras on M2020 Rover

2.1 Engineering Camera Instrument Suite

Each Engineering Camera (ECAM) is composed of a detector and electronics within a single mechanical housing, specifically: 1) an optical lens assembly, 2) a CMOS image sensor and 3) electronics that includes, a Field Programmable Gate Array, memory, and a legacy interface electronics to the rover.). All of the EECAMs use identical 5120 x 3840, 20 megapixel global shutter CMOS image sensors with a Bayer filter/pattern as shown in figure 2.1. Because of the electronics commonality, image date from all engineering cameras are functionally equivalent. The cameras have an external film heater that warms up the electronics to above the minimum operating temperature of -55 deg. C. The rover provides supply voltage of +5.5 V to the cameras. Each camera weighs less than 750 grams and uses approximately 4.5 Watts of power.

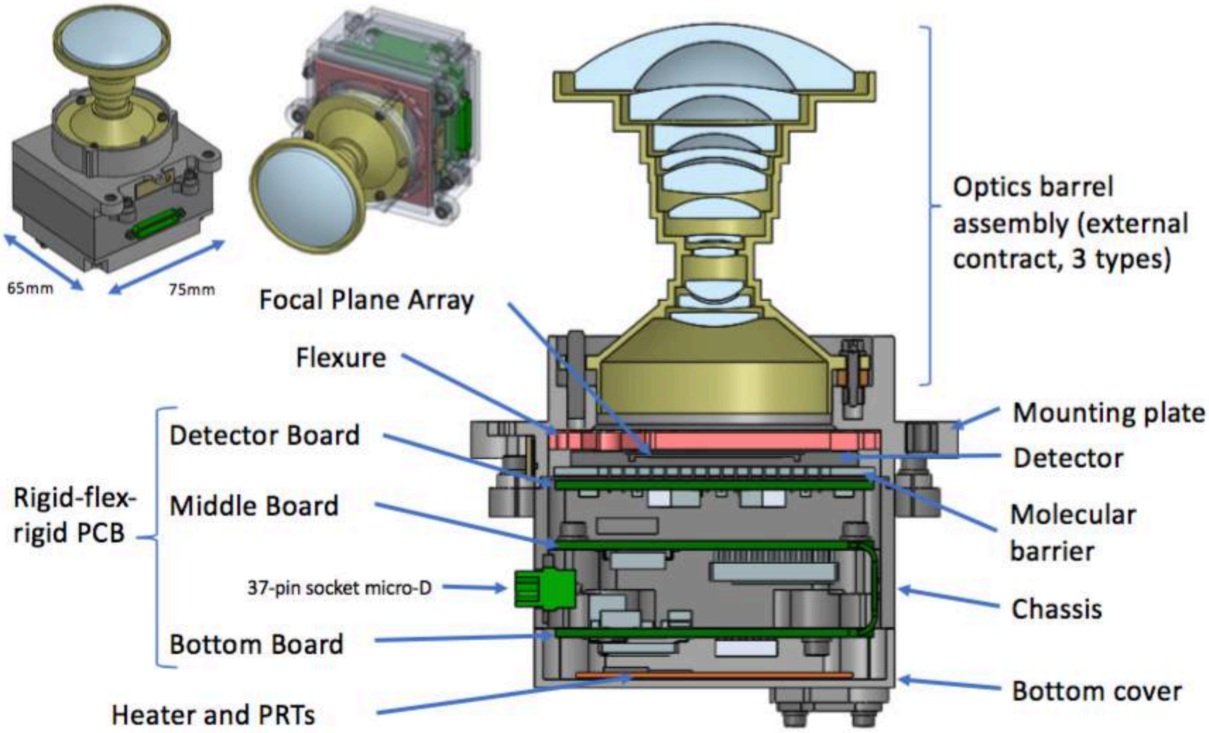


Figure 2-3 – Engineering Camera hardware design

TBD: Insert Image of Camera Detector Readout

This document has been reviewed and determined not to contain export controlled technical data.

2.1.1 Hazard Avoidance Camera (Hazcam)

The Hazard Avoidance Cameras (Hazcams) are two stereo pairs (front) and one stereo pair (rear) of engineering cameras with fish-eye lenses at 25 cm (front) and 93.4 cm (rear) baseline separation mounted at both the front and rear ends of the Warm Electronics Box (WEB).

The Hazcams provide imaging primarily of the near field (< 5 m) both in front of and behind the rover. These cameras will be used to determine safe driving directions for the rover and provide for on-board hazard detection using stereo data to build range maps. They also support science operations for selecting near field target and robotic arm operations.

Hazcam optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-2, with Hazcam locations shown in Figure 2-3 – 2-4 below:

Table 2-2 – Hazcam operational characteristics

Characteristic	Value
Field of View (FOV)	140 x 105 deg
Baseline Stereo Separation	25 cm for front, 93.4 cm for rear
Angular Resolution	0.46 mrad/pixel at center
Pixel Size	6.4 μm x 6.4 μm
Spectral Bandpass	400 - 675 nm
Focal Length	13.8 mm
f/number	12
Depth of Field	0.8 m - infinity
Best Focus	1.0 m

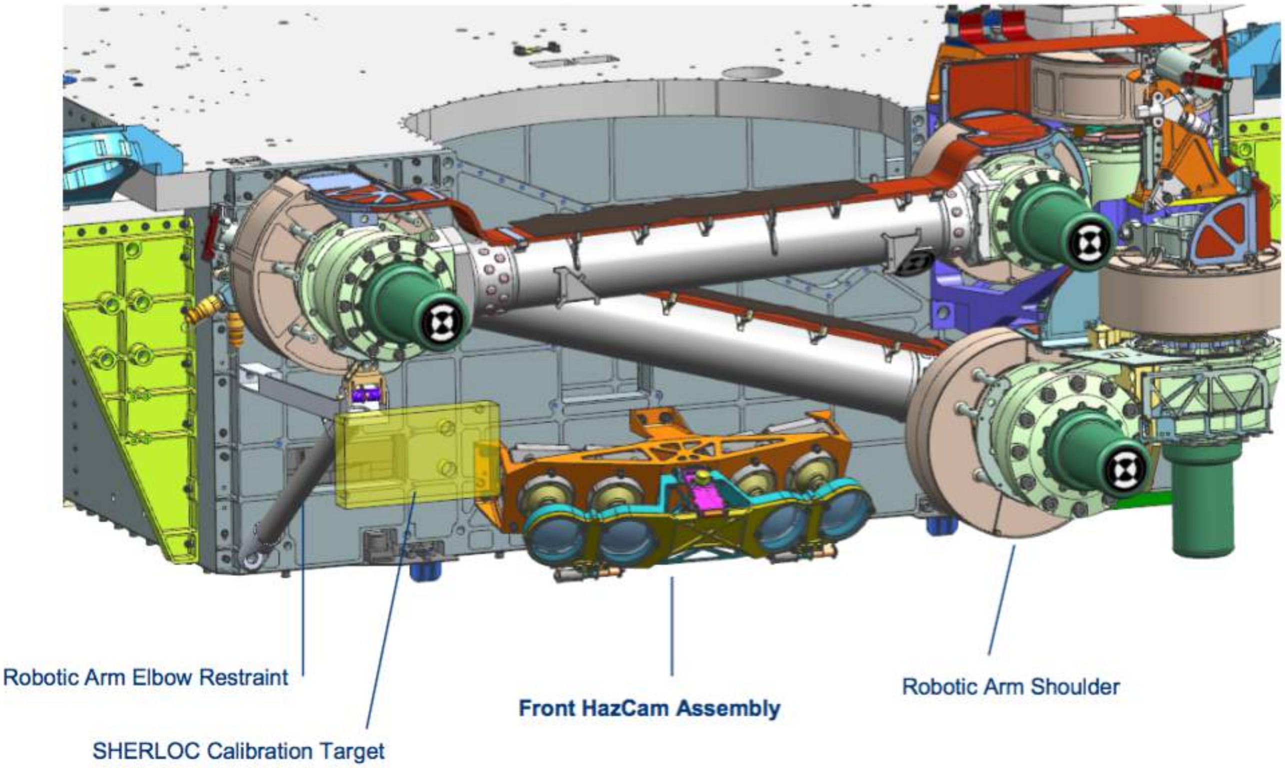


Figure 2-4 – Front Hazcam locations

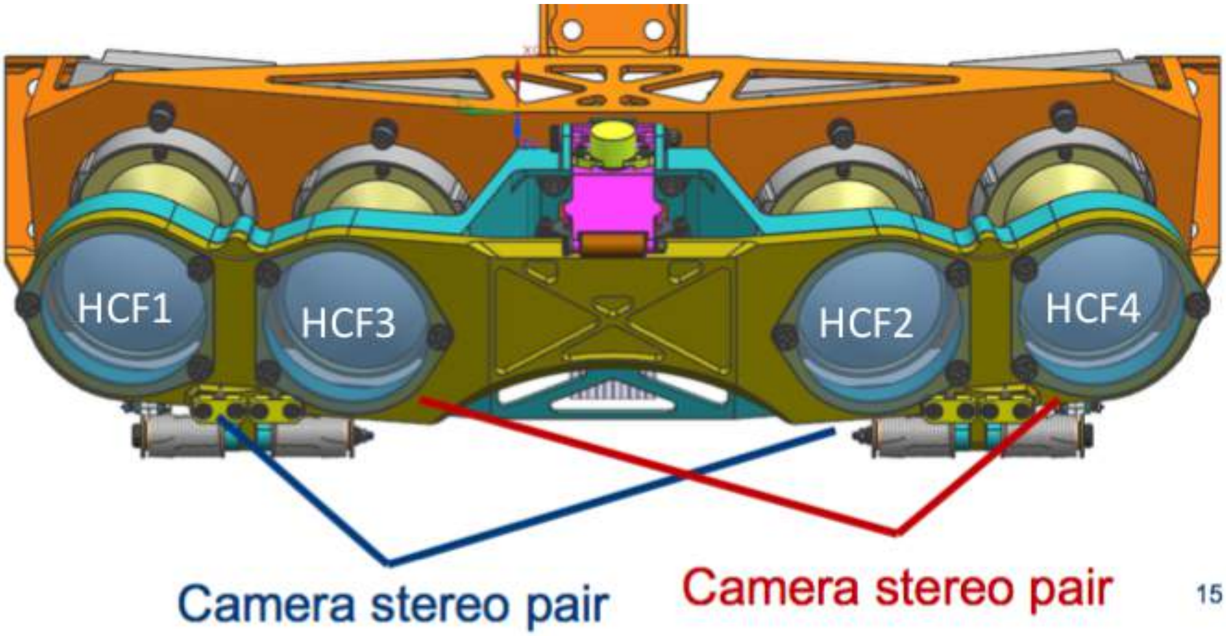


Figure 2-5 – Front Hazcam stereo pair mapping

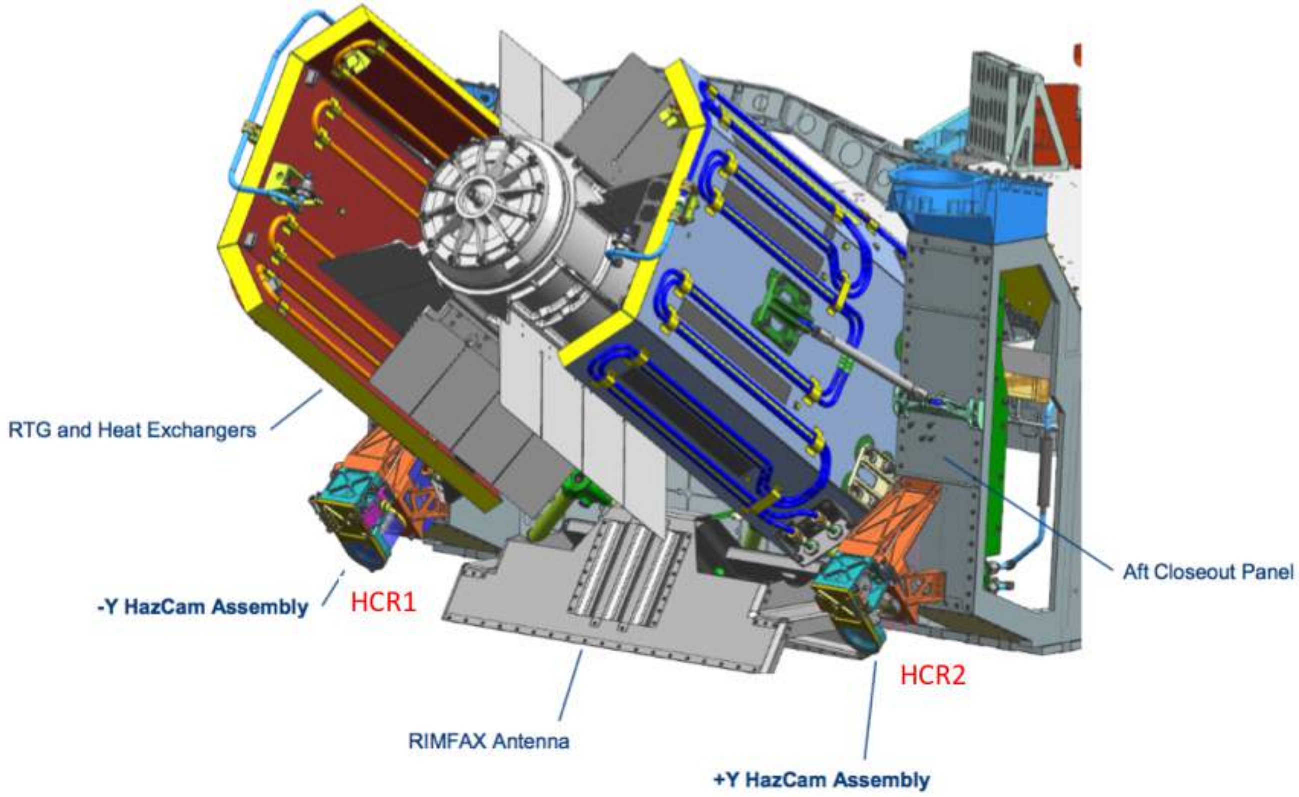


Figure 2-6 – Rear Hazcam location

2.1.2 Navigation Camera (Navcam)

The Navigation Camera (Navcam) is a mast-mounted stereo pair of engineering cameras at 42.4 cm baseline separation with a spectral bandpass at approximately 275 nm. It will primarily be used for navigation purposes and general site characterization (360° panoramic images and targeted images of interest, including terrain not viewable by the Hazcams).

The cameras are boresighted with the Mastcam-Z, and Navcam images will also be used for Science target selection and analysis.

Navcam optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-3, with Navcam location shown in Figure 2-6 below:

Table 2-3 – Navcam Operational Characteristics

Characteristic	Value
Field of View (FOV)	95° x 71°
Baseline Stereo Separation	42.4 cm
Angular Resolution	0.32 mrad/pixel at center
Pixel Size	6.4 µm x 6.4 µm
Spectral Bandpass	400 – 675 nm
Focal Length	19.2 mm

Characteristic	Value
f/number	12
Depth of Field	1.5 m - infinity
Best Focus	3.5 m

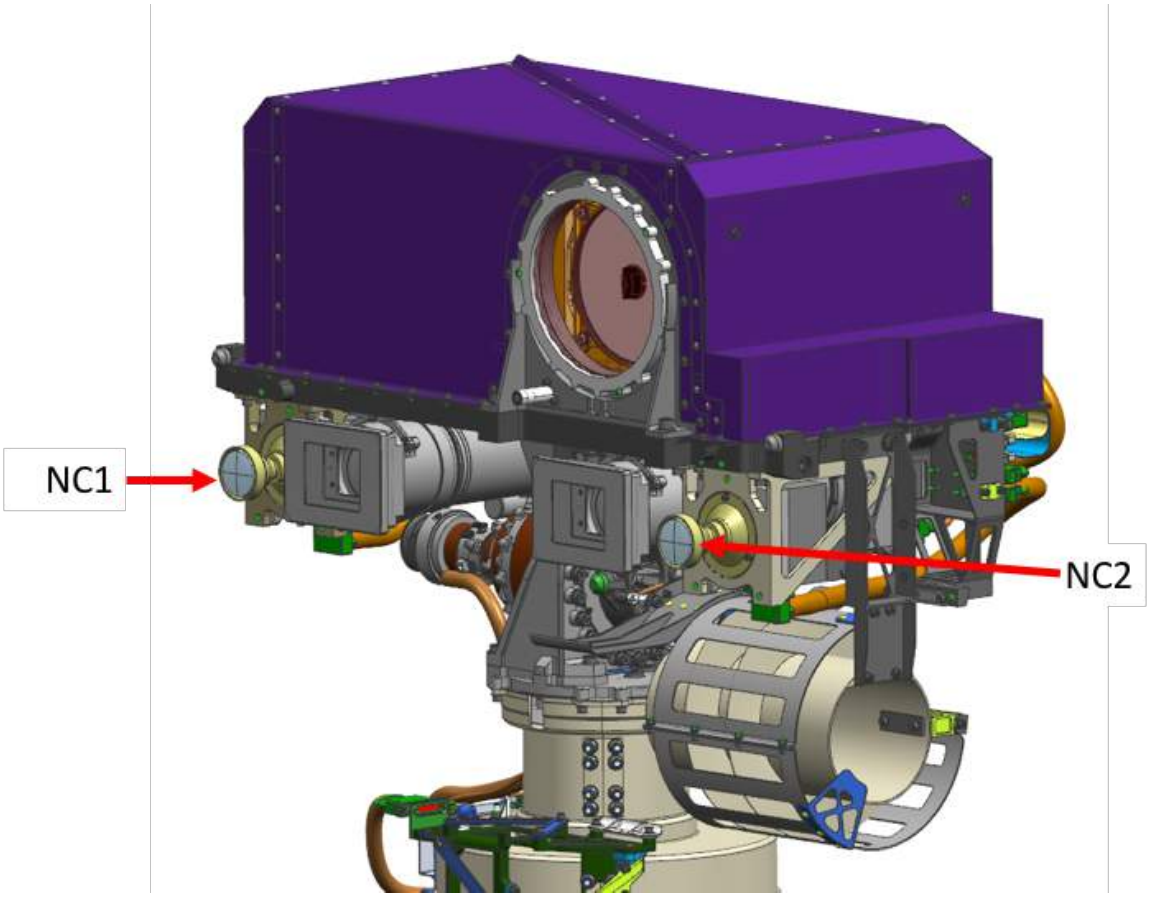


Figure 2-7 – Navcam location

2.1.3 Sample Caching System Camera (CacheCam)

The CacheCam is a single camera mounted inside the body of the rover. It acquires images of the contents of the sample tubes as they are acquired before being sealed. It does not articulate, although the tube carrier can articulate.

Table 2-4 – CacheCam Operational Characteristics

Characteristic	Value
Field of View (FOV) Horizontal	65 mm
Field of View (FOV) Vertical	49 mm

Characteristic	Value
Field of View (FOV) Diagonal	81 mm
Angular Resolution	TBD
Pixel scale (center of FOV)	13 microns/pixel
Spectral Bandpass	TBD
Focal Length	32 mm
Focal ratio	f/8
Depth of Field	+ / - 5 mm
Best Focus	140 mm below illuminator mirror

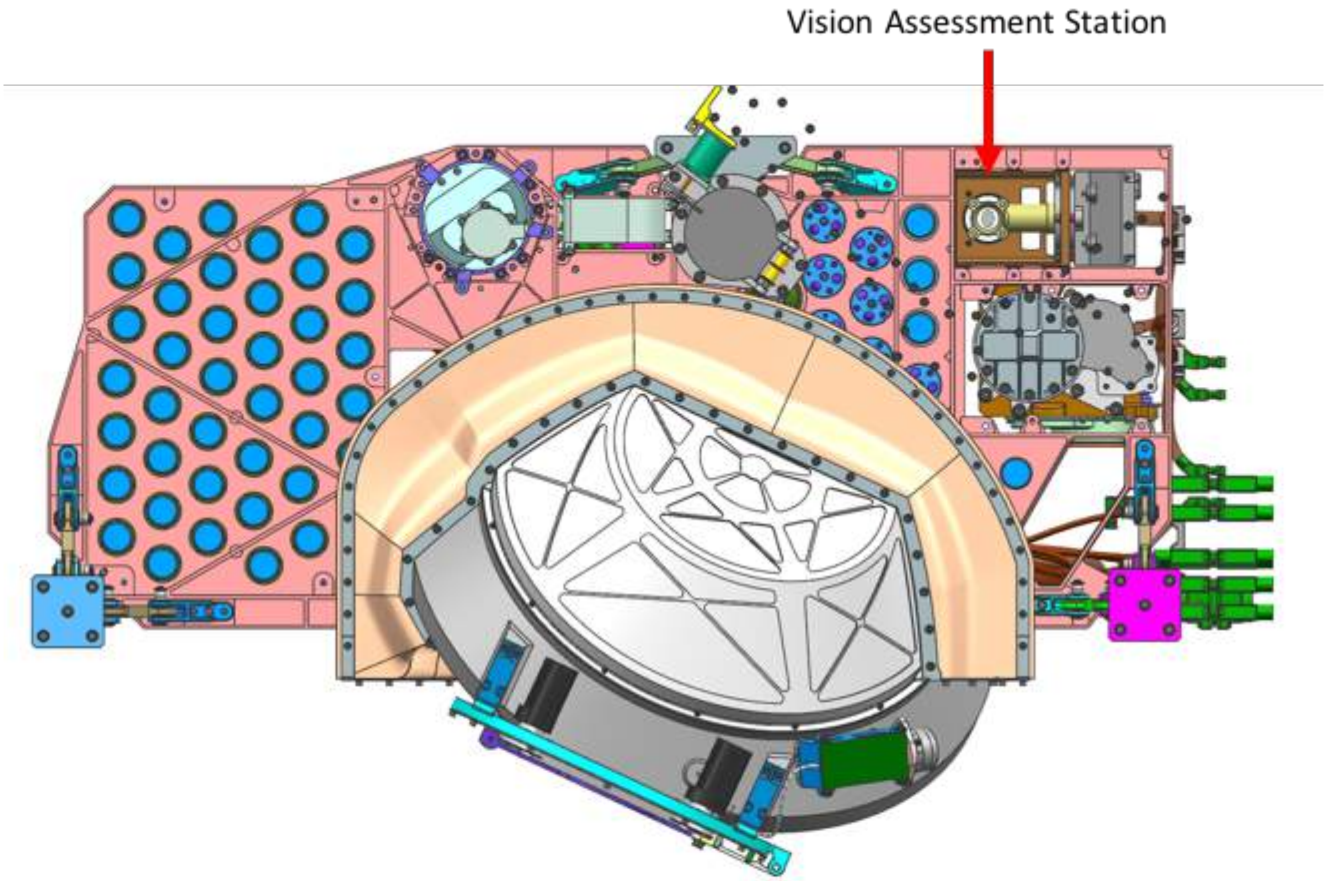


Figure 2-8 – CacheCam location

2.2 SuperCam Remote Micro-Imager (RMI)

The SuperCam RMI camera subsystem is packaged onboard the rover in two instrumentation units: the Mast Unit affixed to the RSM (Figure 2-9), and the Body Unit housed within the rover WEB. The RMI camera provides high-resolution color context images of the LIBS sampling area as well as imaging rock morphologies and distant features. It includes an adjustable focus capability with 2048 x 2048 CMOS color detector (Bayer Filter). Raw RMI images are 13-bit (coded on 16-bits) and are often transformed onboard into four images of 1024 x 1024 pixels each using the red, green1, green2, and blue (RGGB) Bayer cells.

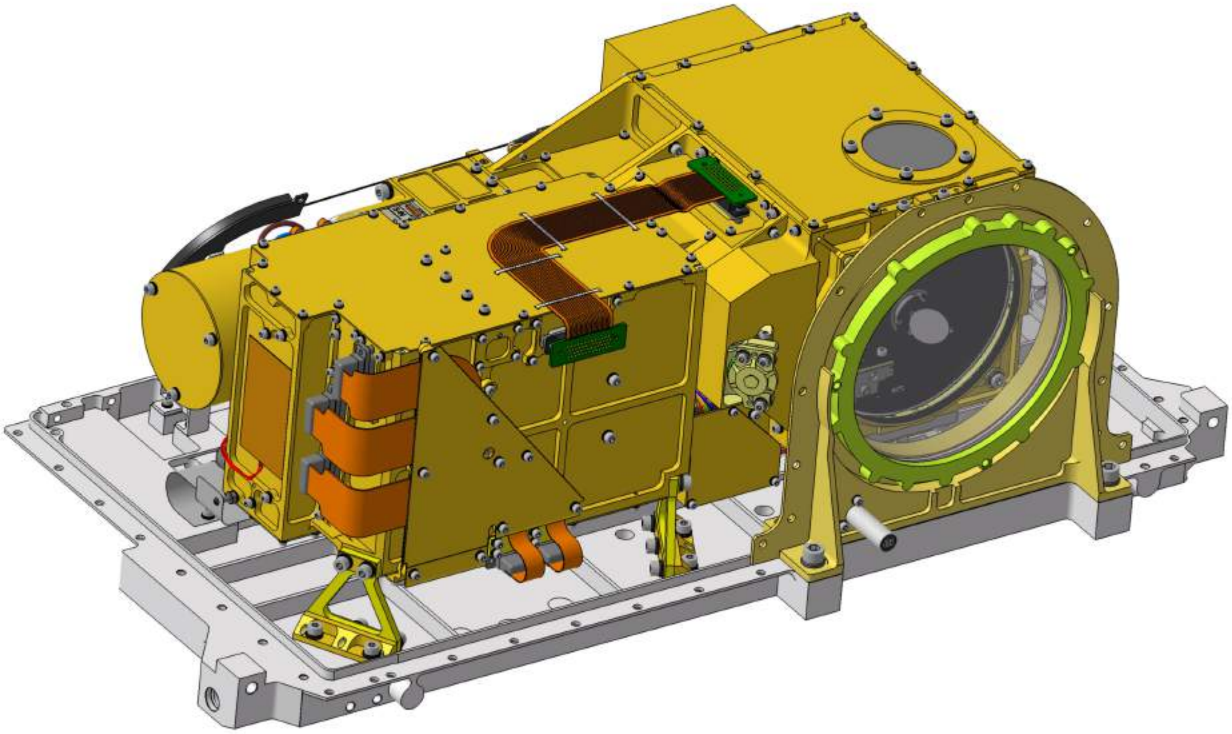


Figure 2-9 – SuperCam Mast Unit

RMI optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-5 below:

Table 2-5 – SCAM RMI Operational Characteristics

Characteristic	Value
Field of View (FOV) useful	18.8 mrad
Field of View (FOV) full frame, including dark corners	20.6 mrad
Spatial Resolution	80 µmrad

Characteristic	Value
with 20% contrast	
Angular Resolution	10 µmrad/pixel
Spectral Bandpass	~370 - ~680 nm
Focus Range	~1.05 m to infinity
Number of Spectral Filters	0

RMI image data return compression modes are listed in Table 2-6 below:

Table 2-6 – RMI Compression Modes

Mode	Description
0	No compression
1	ICER
2	JPEG

The camera acquisition of the scene and subsequent onboard storage and readout of image data is illustrated in Figure 2-X below:

TBD: Insert Image Readout graphic here

2.3 Mastcam-Z Camera Instrument Suite

Mastcam-Z is comprised of a pair of color-capable zoomable, focusable stereo cameras (“eyes”) mounted on the rover’s Remote Sensing Mast (RSM). Each camera has identical zoom/focus capabilities and a different set of spectral filters. Together, they can acquire images of up to 1600 x 1200 pixels and are capable of video. They acquire color via Bayer-pattern filters on the CCD, but also have selectable filters for science/geology applications. For more details on Bayer pattern filters, see Section 2.10.1.2.

Mastcam-Z optics characteristics that are useful in the analysis of EDR and RDR products are described in Table 2-7 below:

Table 2-7 – Mastcam-Z Operational Characteristics

Characteristic	Left/Right Eye
Field of View (FOV)	Wide: 25.6° x 19.2 ° Narrow: 6.18° x 4.63°
Baseline Stereo Separation	24.3 cm
Toe-in	1.25°

Characteristic	Left/Right Eye
Spatial Scale	148-540 mm at 2 m, 7.4-27 cm at 1 km range
Angular Resolution	Wide: 283 μrad Narrow: 67.4 μrad
Focal Length	26.16 – 109.9 mm zoom range
f/number	Wide: f/7 Narrow: f/9
Focus Range	0.5 m to ∞ for 26 mm focal length 1 m to ∞ for 110 mm focal length
Pixel Pitch	7.4 microns
Number of Spectral Filters per Each Eye	8 including Broadband RGB Bayer pattern

Each Mastcam-Z camera has an 8-position filter wheel. One of the positions is a broadband filter for use with the Bayer color capability of the CCD. One other filter per eye has a neutral density coating to provide direct solar imaging capability in two colors. The remaining 12 filters are used for science imaging. The spectral bandwidths [Ref 28] are described in Table 2-8 below:

Table 2-8 – Mastcam-Z Spectral Filters and Bandpass

Filter Position	Left Eye Wavelength (± Bandwidth), λ_{eff} ± HWHM (nm)	Right Eye Wavelength (± Bandwidth), λ_{eff} ± HWHM (nm)
0	630 ± 43 (Bayer filter Red)	631 ± 43 (Bayer filter Red)
	544 ± 41 (Bayer filter Green)	544 ± 42 (Bayer filter Green)
	480 ± 46 (Bayer filter Blue)	480 ± 46 (Bayer filter Blue)
1	800 ± 9	800 ± 9
2	754 ± 10	866 ± 10
3	677 ± 11	910 ± 12
4	605 ± 9	939 ± 12
5	528 ± 11	978 ± 10
6	442 ± 12	1022 ± 19
7	590 ± 88, ND6	880 ± 10, ND5

The Mastcam-Z filter response profiles, including those of the RGB Bayer filters, are plotted below for the left and right camera eyes. Figure 2-10 [Ref 31] shows Spectral Response (rl;k) for filters L0-L6 and R0-R6 in linear space at a detector temperature of -5C. For the purposes of this plot, only the dominant RGB Bayer response is shown any given filter, except L0/R0.

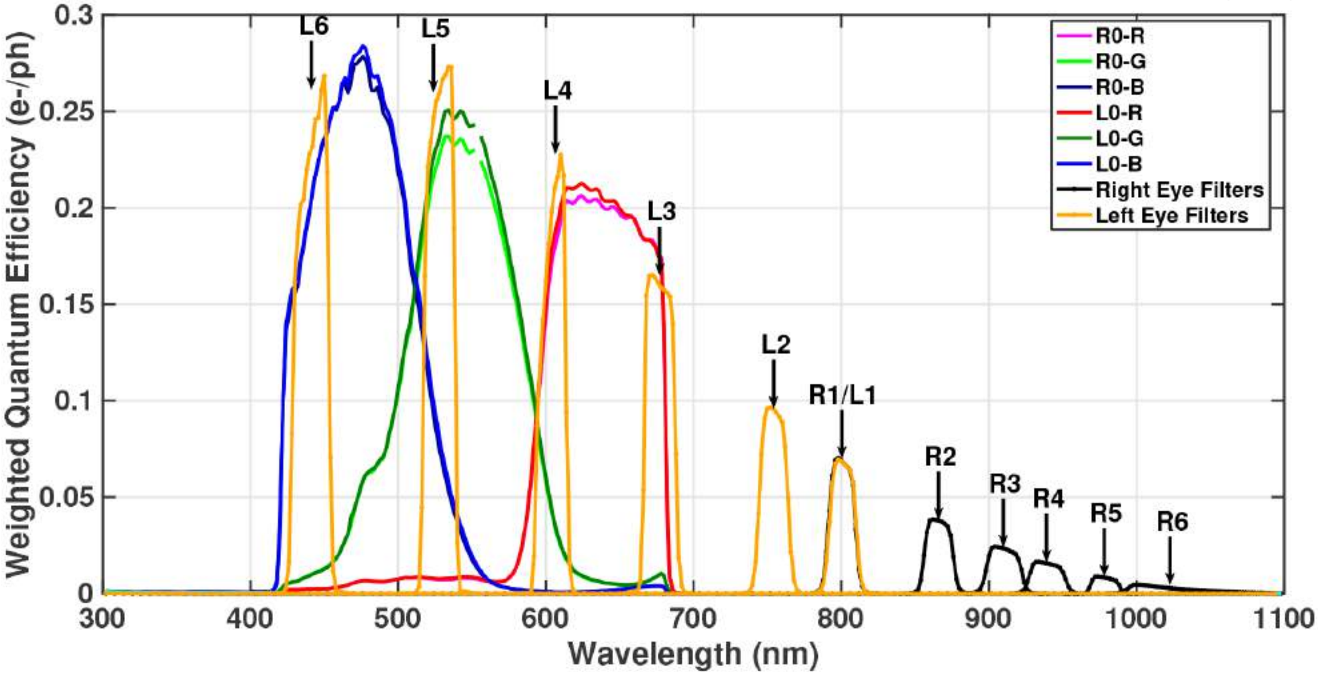


Figure 2-10 – Mastcam-Z Spectral Response

Mastcam-Z Zoom Motor counts and corresponding Focal Length values are listed in Table 2.9 below:

Table 2-9– ZCAM known focal length to Zoom Motor Counts

Zoom Motor Count (z)	Focal Length (f) (mm)
0	26
2448	34
3834	48
5196	63
6720	79
8652	100
9600	110

The relationship of Zoom vs. Focal Length is approximated as a piecewise linear function. Thus for a given Zoom=z, corresponding F = ((f2-f1) / (z2-z1)) * (z – z1) + f1 where z1,z2 are min and max values that z falls into, and f1, f2 are corresponding interval's min and max focal length values.

2.4 SHERLOC Camera Suite

2.4.1 SHERLOC-WATSON

The SHERLOC-WATSON camera is a focusable color camera located on the turret at the end of the M2020 robotic arm and is a build-to-print copy of the MSL MAHLI instrument. The instrument acquires images of up to 1600 by 1200 pixels with a color quality equivalent to that of consumer digital cameras using a Bayer pattern. For details on Bayer pattern filters, see Section 2.10.1.2. It is also capable of video. SHERLOC-WATSON optics characteristics useful in the analysis of EDR and RDR products are described in Table 2-10 below.

Table 2-10 – SHERLOC-WATSON Operational Characteristics

Characteristic	Value
Field of View (FOV)	34.0 - 39.4° deg diagonal
Spatial Resolution	15.9 µm/pixel at 25 mm distance
Angular Resolution	0.3 - 0.34 mrad/pixel
Spectral Wavelength ± Bandwidth (λeff ± HWHM)	590 ± 88 nm (Broadband)
	640 ± 44 nm (Bayer filter Red)
	554 ± 38 nm (Bayer filter Green)
	495 ± 37 nm (Bayer filter Blue)
Focal Length	18.3 - 21.3 mm
f/number	9.8 - 8.5
Depth of Field	1.6 mm - >4800 mm
Focus Range	20.5 mm - infinity
Number of Spectral Filters	1 plus Bayer pattern on CCD

Note that the spatial resolution in Table 2-10 measures the working distance, which is not the same as the distance from the camera model (C) point. Spatial resolution may be calculated by:

Pixel scale (µm/pixel) = 6.9001 + [3.5201 * Working Distance (cm)]
TBD: Describe Working Distance

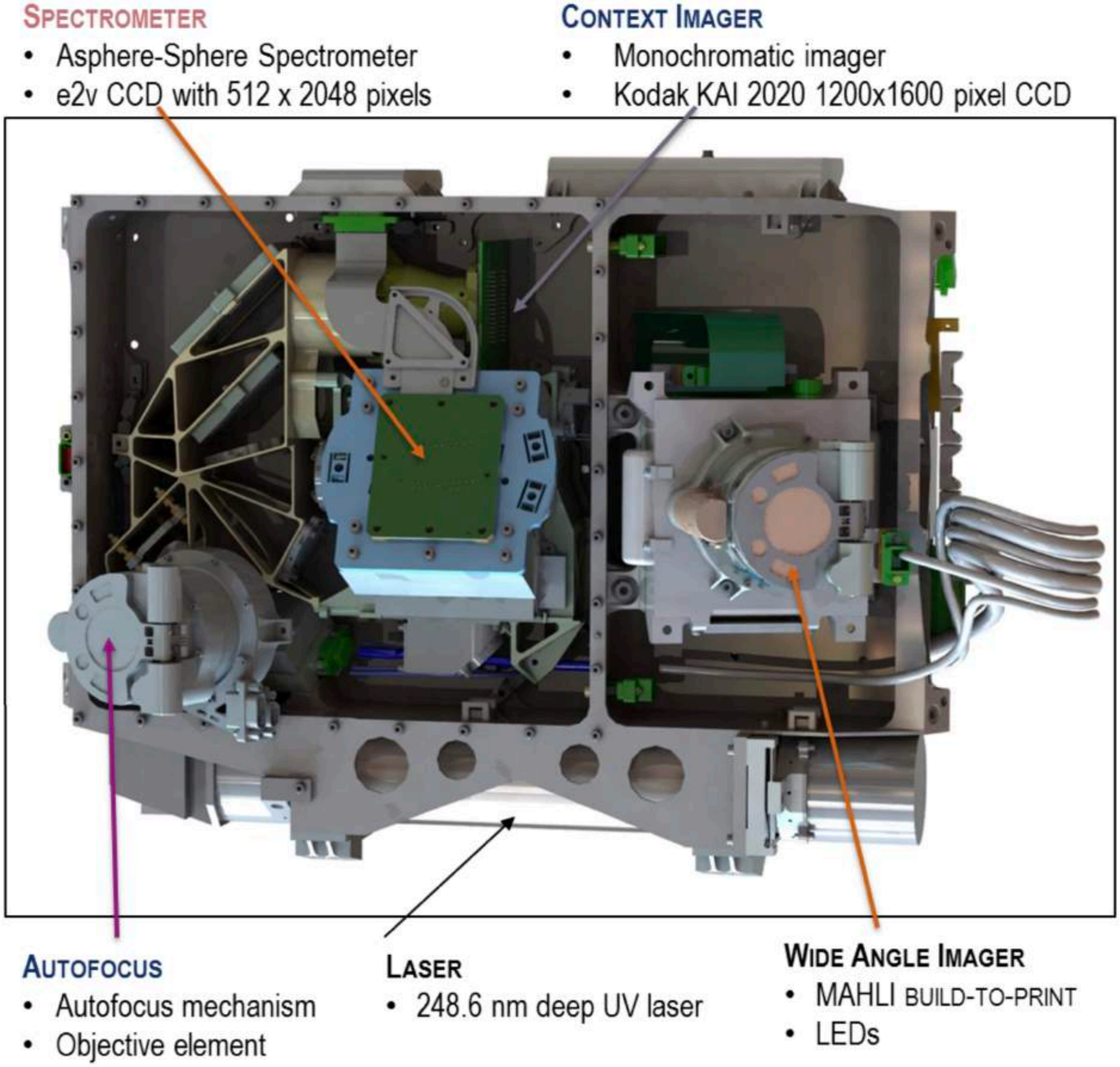


Figure 2-11– WATSON and ACI Cameras of SHERLOC instrument suite.

2.4.2 Autofocus and Context Imager (ACI)

The SHERLOC-ACI is co-boresighted with the laser illumination and spectral collection path. Light from the target (indirect sunlight or LED illumination) is collected by the autofocus objective and directed to the context imager by the foreoptics.

Table 2-11 – SHERLOC-ACI Operational Characteristics

Characteristic	Value
Field of View (FOV)	34.0 – 39.4° diagonal

Characteristic	Value
Spatial Resolution	15 µm/pixel at 25 mm distance
Angular Resolution	0.3 - 0.34 mrad/pixel
Spectral Wavelength	500 ± 50 nm (Broadband)
Depth of Field	1.6 mm - >4800 mm
Number of Spectral Filters	0 (monochrome detector)

2.5 PIXL Micro Context Camera (MCC)

The PIXL Micro Context Camera (MCC) consists of a Data Processing Unit (DPU), a Camera Head Unit (CHU) with Structured Light Illuminators (SLIE), Flood Light Illuminator Electronics, and Flood Light Illuminator (FLI). The DPU is mounted within the rover body and all of the optical components are mounted on the turret of the robotic arm.

The CCD is 7.95mm x 6.45mm, with 752x580 pixels, and each pixel is 8.6 x 8.3 um. The camera has a 12mm optical lens. FLI consists of 4 color channels R, G, B and UV. Each channel has 6 LEDs and each color can be controlled individually. Green LEDs can be individually controlled.

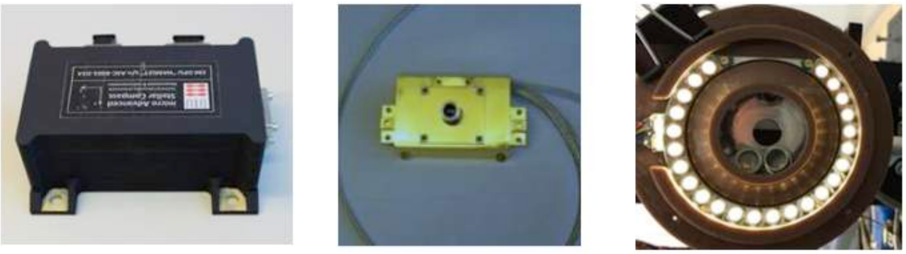
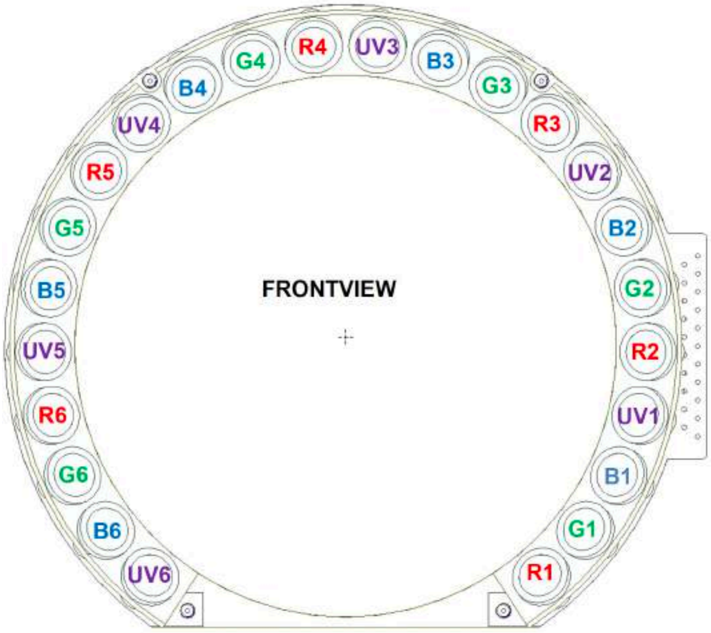


Figure 2-12 – DPU, CHU, FLI and two SLIs mounted with CHU

Control of whether to use specific color channel, is achieved by setting the current level for each LED string. To disable/enable a specific string, the current is set to 0mA/500mA. Figure 2-11 provides the mapping of LED IDs on the FLI and an overview of the arrangement of the LED relative to the output strings from FLIE A and FLIE B.



Color	String	LED ID
Red	FLIE A 1	R1, R3, R5
	FLIE B 1	R2, R4, R6
	ALL (6xLEDs)	ALL R
Green	FLIE A 3	G1
	FLIE B 3	G2
	FLIE B 4	G3
	FLIE A 4	G4
	FLIE B 5	G5
	FLIE B 6	G6
	ALL (6xLEDs)	ALL G
Blue	FLIE A 5	B1, B4
	FLIE A 7	B2, B5
	FLIE B 7	B3, B6
	ALL (6xLEDs)	ALL B
UV	FLIE A 6	UV1, UV4
	FLIE A 8	UV2, UV5
	FLIE B 8	UV3, UV6
	ALL (6xLEDs)	ALL UV
White	ALL	ALL R, G, B, UV

Figure 2-13 – LED distribution on the FLI and corresponding LED ID

2.6 MEDA Sky Camera

The MEDA Skycam is a build-to-print copy of the MER and MSL Engineering Cameras. The camera is mounted in the center of the Radiation and Dust Sensor (RDS) and also includes a 120º FOV Baffle (ring on lens) that will be used to block sunlight at specific times during the Martian Sol.

Table 2-12 – SkyCam Operational Characteristics

Characteristic	Value
Field of View (FOV)	124 x 124 deg
Angular Resolution	2.1 mrad/pixel at center
Spectral Bandpass	600 - 800 nm
Focal Length	5.58 mm
f/number	15
Depth of Field	0.1 m - infinity
Best Focus	0.5 m

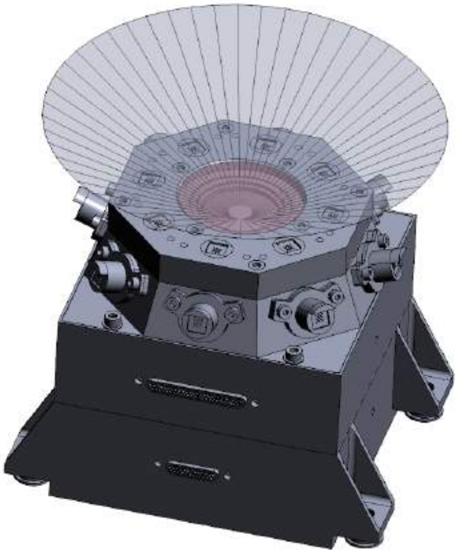


Figure 2-14 – RDS-Skycam with FOV

2.7 EDL Camera Suite

The EDL camera suite consists of 7 COTS FLIR (formerly Point Grey) cameras mounted to the backshell, descent stage structure, and the rover body.

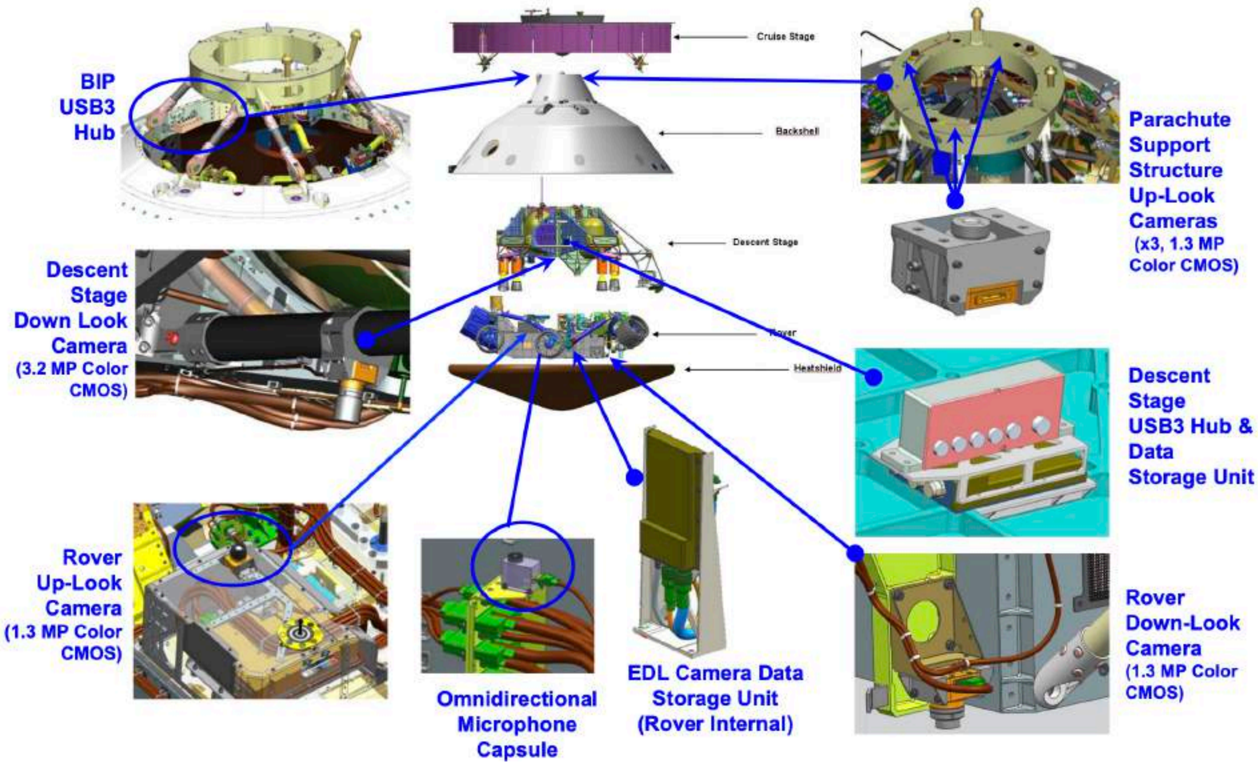

BIP USB3 Hub
Cruise Stage
Backshell
Descent Stage
Rover
Heatshield
Parachute Support Structure Up-Look Cameras (x3, 1.3 MP Color CMOS)
Descent Stage Down Look Camera (3.2 MP Color CMOS)
Descent Stage USB3 Hub & Data Storage Unit
Rover Up-Look Camera (1.3 MP Color CMOS)
Omnidirectional Microphone Capsule
EDL Camera Data Storage Unit (Rover Internal)
Rover Down-Look Camera (1.3 MP Color CMOS)

Figure 2-15 – EDL Cameras overview

2.7.1 Parachute Uplook Cameras (PUC)

The 3 Parachute Uplook Cameras (PUC) are FLIR Chamleon 3 (model CM3-U3-13Y3C-CS). The cameras are mounted on the top of the Backshell. Each camera has identical capabilities and can acquire color (Bayer pattern CCD) images of up to 1280 x 1024 pixels and are capable of video at 149 fps with global shutter readout. The PUCs start to acquire image data immediately before parachute deployment at a frame rate of 75 fps (frames per second) and continue acquiring data at that rate for 30 seconds. After 30 seconds the frame rate drops to 30 fps until backshell separation, expected to occur approximately 98 seconds later. The total number of expected PUC images is ~ 5,190 images per PUC, or 15,570 total images.

2.7.2 Descent Stage Downlook Camera (DDC)

This camera is located at the bottom of the Descent Stage structure and will look downward at the Rover to capture imagery of the following events: rover dynamics during bridle descent and mobility deployment and Descent Stage main engine plume interactions with the Martian surface through touch down. The DDC will start acquiring image data just before heatshield separation. The DDC acquires data at 12 fps for ~ 75 seconds and is expected to acquire approximately 900 images.

2.7.3 Rover Uplook Camera (RUC)

This camera is located on the top of the Rover and will look upward at the Descent Stage Record descent stage dynamics from sky crane through touchdown and capture flyaway event. The RUC will start acquiring data just before heatshield separation. The RUC acquires data at 30 fps for approximately 130 seconds and is expected to generate approximately 4,000 images.

2.7.4 Rover Downlook Camera (RDC)

This camera will be located on the side of the Rover and will look downward at the Martian surface during EDL starting at heatshield release through touchdown on the surface. The RDC will start acquiring data just before heatshield separation. The RDC acquires data at 30 fps for approximately 260 seconds and is expected to acquire a total of approximately 7,500 images.

2.7.5 EDL Microphone

The EDL camera package also includes a microphone, that will be used to record audio during EDL (Entry, Descent, and Landing), and possibly other times throughout the mission.

Table 2-13 – EDL Cameras Operational Characteristics

Characteristic	PUC/RUC/RDC	DDC
Field of View (FOV) (horizontal)	35 deg ± 3	48 deg ± 3
Field of View (FOV) (vertical)	30 deg ± 3	37 deg ± 3
Pixel scale	~0.5 mrad/pixel	~0.4 mrad/pixel
Spectral Bandpass	TBD	TBD
Focal Length	9.5 mm	8 mm
f/number	f/7	f/5.6
Depth of Field	TBD	TBD
Best Focus	44 m	TBD

TBD: add section on format of the Microphone data (.WAV file)

2.8 Lander Visusal System Camera (LCAM)

The LVS Camera (LCAM) is intended to be used only during EDL to acquire images under high attitude rates and velocities. It is located on the outside of the rover body. The camera has a global shutter with a low exposure time and has a wide field of view to be able to capture terrain features even when camera is pointed up to 45˚ off-nadir.

Table 2-14 – LCAM Operational Characteristics

Characteristic	Value
Detector Type	Global shutter and grayscale
Number of Pixels	1024 x 1024
Field of View	90˚x90˚
Focal Ratio	f/2.7
Focal Length	5.8mm
Best Focus	2m to infinity
Signal-to-Noise Ratio	80 at half full well depth
Exposure Time	~1ms
Latency	~100ms

2.9 Helicopter imaging suite

The Mars 2020 rover carries a helicopter as a technology demonstration. The helicopter will fly up to 5 times over a ~30 sol period early in the mission. The helicopter carries two cameras, one for navigation and one for capturing aerial views for science evaluation. The images are stored onboard during flight, and then are transmitted to the rover afterwards for relay to Earth.

2.9.1 Helicopter Navigation Camera

The helicopter navigation camera is a 640x480 pixel, 8-bit, grayscale camera. It points straight down from the belly of the helicopter. It is used by the onboard navigation software to determine helicopter position and attitude, and to help navigate to the desired destination. Selected images may be returned to Earth for analysis.
TBD: Characteristics table

2.9.2 Helicopter Return to Earth Camera

The helicopter “Return to Earth” camera is a 4224x3120 pixel, 8-bit, color camera. It is mounted on the side of the helicopter and faces down at a 45 degree angle [TBC] so it can see both nadir and horizon.

TBD: Characteristics table

2.10 Camera Flight Software Processing

For certain camera instruments the Rover flight-software provides some specific processing to prepare the data for downlink. For ECAM, SuperCam RMI, and MEDA Skycam instruments the Rover FSW handles the compression of data products. There are three special cases that require specific processing steps: JPEG compression for ECAM and SuperCam, Color ICER compression for Supercam, and ICER/LOCO compression for MEDA. Compression of Mastcam-Z and SHERLOC are done by the camera itself (although similar companding and Bayer de-mosaicking steps may be performed by the cameras).

2.10.1 JPEG for ECAM and SuperCam

When ECAM or SuperCam request sending data in JPEG compression mode, the rover flight-software performs 2 processing steps on the data before compression: companding and bayer de-mosaicking.

2.10.1.1 Companding

2.10.1.1.1 Engineering Cameras

Flight software converts the data from 12-bits per pixel to 8 bits per pixel. A square root encoding that preserves detail in the dark areas while allowing a high dynamic range in the bright areas via a look-up table (LUT) or simple bit-shifting is used. LUTs are provided by the ECAM instrument team.

2.10.1.1.2 SuperCam RMI

Flight software converts the data from 13-, 12-, and 10-bit pixels to 8-bits per pixel. A square root encoding that preserves detail in the dark areas while allowing a high dynamic range in the bright areas via a look-up table (LUT) or simple bit-shifting is used. LUTs are provided by the SuperCam instrument team.

2.10.1.2 Bayer De-mosaicking

The Bayer pattern (RGGB) is removed by the flight-software using the Malvar de-mosaicking algorithm (Malvar 2004, [28]). The Malvar algorithm is one of several de-mosaicking algorithms in widespread use today. Malvar was also used on MSL for the Mastcam, MAHLI, and MARDI instruments (on-board and ground), and is used by the Mastcam-Z and SHERLOC-Watson cameras.

The Malvar algorithm produces three images, one for each color band. The missing color bands at each pixel are created using the assumption that chrominance (color) varies slowly, while luminance (intensity) varies quickly over the scene. The chrominance is determined by examining neighboring pixels; variations from this are assumed to be luminance changes. Thus the reconstructed image retains the full spatial resolution of the raw image. 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.

2.10.2 Color ICER for SuperCam

When SuperCam requests sending data in ICER compression mode, the SuperCam Instrument FSW extracts the four Red, Green-1, Green-2, and Blue color arrays and sends each array separately to the ICER compressor with the same compression parameters. Rover FSW then packs the 4 separate color channel "images" into a single downlink product. This is re-converted on the ground into the original Bayer-pattern image (which then requires de-Bayering in the ground system).

2.10.3 Compression for MEDA

The MEDA Skycam images stored in instrument flash memory have been compacted into image segments that store 2 pixels of data in 3 bytes of memory. When the instrument transfers the data to the RCE for downlink, the Rover FSW iteratively receives each segement of the image and decompacts the segments to create the full-size image. The full-size image is then compressed per the compression parameters and stored on the RCE for downlink.

3. GENERAL DATA PRODUCT OVERVIEW

3.1 Data Processing Levels

This documentation generally avoids the PDS4 processing level terminology, instead using the EDR/FDR/RDR terminology adopted by the Mars 2020 operations team. Table 3-1 shows the general mapping between the two. EDRs are defined as products that reconstruct as closely as possible the data acquired by the camera, given transmission constraints. So they include the "Raw" products derived straight from telemetry, as well as the "Partially Processed" products relating to decompanding and de-Bayering, which simply reconstruct the data acquired by the sensor without any form of calibration or processing. FDRs are the "Fundamental Data Record"; this is a consistently-formatted product at the end of the EDR chain that is used as the basis for all downstream RDR processing (it also includes tile reassembly for the Engineering Cameras, as well as some label updates on certain instruments). RDR (Reduced Data Records) are then all downstream products, encompassing the PDS levels "Partially Processed", "Calibrated", and "Derived".

Table 3-1 lists the PDS4 processing levels and the general categories they apply to, while Table 6-5 shows the PDS4 processing level for each specific type of image.

Table 3-1 – PDS4 Processing Levels for Instrument Experiment Data Sets

Processing Level for PDS4 Archive	Operations Data Product Name	Description
Telemetry	n/a	An encoded byte stream used to transfer data from one or more instruments to temporary storage where the raw instrument data will be extracted. PDS does not archive telemetry data.
Raw	EDR (Experiment Data Record, heritage term based on MSL mission)	Original data from an instrument. If compression, reformatting, packetization, or other translation has been applied to facilitate data transmission or storage, those processes will be reversed so that the archived data are in a PDS approved archive format. For Mars 2020, these are the ECM/ECV/ECZ/EDM/ECR original products.
Partially Processed	EDR FDR (Fundamental Data Record) RDR (Reduced Data Record)	Data that have been processed beyond the raw stage, but which have not yet reached calibrated status. For Mars 2020, these are the decompanded and debayered EDRs, the FDRs, and a few RDRs.
Calibrated	RDR	Data converted to physical units, which makes values independent of the instrument.
Derived	RDR	Results that have been distilled from one or more calibrated data products (for example, maps, gravity or magnetic fields, or ring particle size distributions). 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.

3.2 **Product Label and Header Descriptions**

3.2.1 Overview of Labels

Mars 2020 image products consist of at least two files: the image data file in VICAR/ODL format, and a detached PDS4 label file in XML format. Some products also incorporate the original JPEG stream from the spacecraft (EJP type) as a third, supplemental file. The IMAGE file is in a dual VICAR/ODL format, the same as was used on MER and MSL. It consists of three major parts: the ODL label, the VICAR label, and the image area. Thus, each image actually has three labels: the attached ODL and VICAR labels, which are embedded within the image data and are used during operations, and the detached PDS4 label, which is used for archiving the data at PDS. The VICAR and ODL labels are very similar (see below). All three labels contain the same semantic content and can be used interchangeably. For products produced by IDS, the VICAR label is produced natively from the tools; the ODL and PDS4 labels are then created from that label, which ensures they have the same semantic content.The VICAR label is used internally by the Mars 2020 project's GDS and also the MIPL software. The ODL label is used by many science teams. Both are included for maximum compatibility, so that operations tools can still be used on archive data.

3.2.2 ODL Label

As implied in the previous section, the image EDRs and RDRs described in this document have an attached ODL label and a detached PDS4 label. Each institution is responsible for converting PDS-formatted products to be compatible with their own software systems (such as VICAR, IDL, ISIS, etc.).

The attached ODL label starts with the keyword assignment:

ODL_VERSION_ID = ODL3

An ODL label is object-oriented and describes the objects in the data file. The ODL label contains keywords for product identification, along with the data object definition containing descriptive information needed to interpret or process the data in the file.

ODL label statements have the form of "keyword = value". Each label statement is terminated with a carriage return character (ASCII 13) and a line feed character (ASCII 10) sequence to allow the label to be read by many operating systems. Pointer statements with the following format are used to indicate the location of data objects in the file:

^object = location

where the carat character (^, also called a pointer) is followed by the name of the specific data object. The location is the 1-based starting record number for the data object within the file. Alternatively, it could be the 1-based byte location within the file if it includes a <bytes> unit tag. Pointers are used to define the locations of the binary instrument data itself (^IMAGE for image data), and the VICAR header in the case of images (^IMAGE_HEADER).

3.2.2.1 Keyword Length Limits

All ODL keywords are limited to 31 characters in length. Therefore, software that reads M2020 ODL labels must be able to ingest keywords up to 31 characters in length.

For image RDR-producing institutions wishing to accommodate the VICAR mapping of ODL keywords that use a <*unit*> tag after the value, such keywords must be limited to 25 characters in length to accommodate the "__UNIT" suffix. Otherwise, those keywords will not be transcoded from the ODL label into a VICAR label.

3.2.2.2 Data Type Restrictions

In order to accommodate VICAR dual-labeled files, 16-bit data must be stored as signed data. Unsigned 16-bit data are not supported. 12-bit unsigned data from the cameras are stored in a 16-bit signed value. 8-bit data are unsigned.

3.2.2.3 Interpretation of N/A, UNK, and NULL

During the completion of data product labels or catalog files, one or more values may not be available for some set of required data elements. In this case the literals "N/A", "UNK", and "NULL" are used, each of which is appropriate under different circumstances. As a note, if any one of these three symbolic literals are used in place of a keyword value that is normally followed by a Unit Tag(s) (e.g., "<*value*>"), the Unit Tag(s) is <u>removed</u> from the label.

- "N/A" ("Not Applicable") indicates that the values within the domain of this data element are not applicable in this instance. For example, a data set catalog file describing NAIF SPK kernels would contain the statement:

  ```
  INSTRUMENT_ID = "N/A"
  ```

  because this data set is not associated with a particular instrument.

  "N/A" may be used as needed for data elements of any type (e.g., text, date, numeric, etc.).

- "UNK" ("Unknown") indicates that the value for the data element is not known and never will be. For example, in a data set comprising a series of images, each taken with a different filter, one of the labels might contain the statement:

  ```
  FILTER_NAME = "UNK"
  ```

  if the observing log recording the filter name was lost or destroyed and the name of the filter is not otherwise recoverable.

  "UNK" may be used as needed for data elements of any type.

- "NULL" is used to flag values that are *temporarily* unknown. It indicates that the data preparer recognizes that a specific value should be applied, but that the true value was not readily available. "NULL" is a placeholder. For example, the statement:

  ```
  DATA_SET_RELEASE_DATE = "NULL"
  ```

  might be used in a data set catalog file during the development and review process to indicate that the release date has not yet been determined.

  "NULL" may be used as needed for data elements of any type.

  Note that all "NULL" indicators should be replaced by their actual values prior to final archiving of the associated data.

Unlike earlier missions, some effort has been expended to reduce the number of UNK, N/A, and NULL values appearing in the label, since they can cause difficulties with the Velocity technology used to create PDS4 labels. Therefore, while these values are possible, they should be rare.

3.2.2.4 ODL Label Constructs "Class", "Object" and "Group"

For the EDRs and RDRs described in this document, the ODL label includes the following constructs:

- *Class* - The Class construct resides in a ODL label as a grouping of keywords that are thematically tied together. Classes are usually preceded by a label comment, although it is not required. ODL label comments are character strings bounded by "/* */" characters.

  In the M2020 Camera ODL label a Class of keywords will be preceded by a comment string as follows:

  ```
  /*  comment string  */
  keyword      =   keyword value
  keyword      =   keyword value
  ```

- *Object* - The Object construct is a set of standard keywords used for a particular data product. In the M2020 Camera ODL label an Object's set of keywords is specified as follows:

  ```
  OBJECT         =   Object identifier
      keyword    =   keyword value
      keyword    =   keyword value
  END_OBJECT     =   Object identifier
  ```

- *Group* - The Group construct is a grouping of keywords that are not components of a larger Object. Group keywords may reside in more than one Group within the label. In the M2020 Camera ODL label, a Group's set of keywords is specified as follows:

  ```
  GROUP          =   Group identifier
      keyword    =   keyword value
      keyword    =   keyword value
  END_GROUP      =   Group identifier
  ```

3.2.2.5 ODL Image Object

An IMAGE object is a two-dimensional array of values, all of the same type, each of which is referred to as a *sample*. IMAGE objects are normally processed with special display tools to produce a visual representation of the samples by assigning brightness levels or display colors to the values. An IMAGE consists of a series of lines, each containing the same number of samples.

The required IMAGE keywords define the parameters for simple IMAGE objects:

- LINES is the number of lines in the image.
- LINE_SAMPLES is the number of samples in each line.
- SAMPLE_BITS is the number of bits in each individual sample.
- SAMPLE_TYPE defines the sample data type.

The IMAGE object can have a number of keywords relating to image statistics. These keywords are generally not used on Mars 2020. If they are present, they must be updated to reflect the current statistics of the image. Note that the VICAR label never contains these keywords. The statistics keywords are:

- MEAN
- MEDIAN
- MAXIMUM
- MINIMUM
- STANDARD_DEVIATION
- CHECKSUM

Many variations on the basic IMAGE object are possible with the addition of optional keywords and/or objects. The “^IMAGE” keyword identifies the start of the image.

3.2.3 PDS4 Label

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 core, discipline, 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 “.IMG”), the label removes the 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”. Appendix B contains tables mapping between VICAR/ODL keywords and their corresponding PDS4 construct.

3.2.4 VICAR Label

For all image EDR data products and MIPL produced image RDR data products, an embedded VICAR label follows the ODL label and is pointed to by the ODL pointer “^IMAGE_HEADER”. The VICAR label is also organized in an ASCII, “keyword = value” format, although there are only spaces between keywords (no carriage return/line feeds as in ODL). The information in the VICAR label is an exact copy of the information in the PDS label as defined in the next section. The reader is referred to the VICAR File Format document for details of the format, which is available at the URL http://www-mipl.jpl.nasa.gov/external/VICAR_file_fmt.pdf. The following text is an excerpt which describes the basic structure:

> A VICAR file consists of two major parts: the labels, which describe what the file is, and the image area, which contains the actual image. The labels are potentially split into two parts, one at the beginning of the file, and one at the end. Normally, only the labels at the front of the file will be present. However, if the EOL keyword in the system label (described below) is equal to 1, then the EOL labels (End Of file Labels) are present. This happens if the labels expand beyond the space allocated for them. The VICAR file is treated as a series of fixed-length records, of size RECSIZE (see below). The image area always starts at a record boundary, so there may be unused space at the end of the label, before the actual image data starts.
>
> The 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. The first keyword

is always LBLSIZE, which specifies the size of the label area in bytes. LBLSIZE is always a multiple of RECSIZE, even if the labels don't fill up the record. If the labels end before LBLSIZE is reached (the normal case), then a 0 byte terminates the label string. If the labels are exactly LBLSIZE 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 LBLSIZE bytes, whichever is smaller. If the system keyword EOL has the value 1, then End-Of-file Labels exist at the end of the image area (see above). The EOL labels, if present, start with another LBLSIZE keyword, which is treated exactly the same as the main LBLSIZE keyword. The length of the EOL labels is the smaller of the length to the first 0 byte or the EOL's LBLSIZE. Note that the main LBLSIZE does not include the size of the EOL labels. In order to read in the full label string, simply read in the EOL labels, strip off the LBLSIZE keyword, and append the rest to the end of the main label string.

Note that the EOL labels should not appear in archive products.

A binary header may appear in between the VICAR label and the image, containing arbitrary binary data that are not interpreted by VICAR. The number of records in this header is defined by the VICAR system keyword NLB. Binary headers are not used on Mars 2020.

3.2.5 Mapping of ODL and VICAR Labels

In the cases of the attached ODL and VICAR labels, information content is identical, by definition. ODL and VICAR labels may be used interchangeably, for any purpose in the mission. Any software that modifies one label must also modify the other, or remove them entirely. This is often most easily accomplished by stripping off one of the headers, processing the remaining label as desired locally, and then running a conversion tool to re-create the missing header. Such tools will be provided by MIPL.

The mapping between ODL and VICAR keywords is straightforward, and keyword names are usually the same. However, there are some keyword name differences. The mapping rules are as follows:

- For mapped ODL and VICAR (if applicable) keywords, values are identical in all cases with the exception of differences mandated by the file format itself, such as quoting rules. See the respective ODL and VICAR documents for details, but in general, ODL uses double quotes (") while VICAR uses single quotes (').

- For ODL and VICAR label keywords, with the exception of those defining the file format itself (described below), names are identical in both cases.

- Any ODL label group maps 1-to-1 to a VICAR property set with the same name (Group) name == property set name). All contained keywords are identical in both cases. The GROUP and END-GROUP keywords are omitted from the VICAR label; PROPERTY keywords are used instead (as per the VICAR file format definition).

- ODL keywords not in a group is identified by an introductory comment (e.g. /* IDENTIFICATION DATA ELEMENTS */). Such classes map 1-to-1 to a VICAR property set. The name of the VICAR property set and the name of the ODL introductory comment map as follows:

Table 3-2 – ODL Class to VICAR Property Set Mappings

ODL Class Comment	VICAR Property Set Name
/* FILE DATA ELEMENTS */	special case, see below
/* POINTERS TO DATA OBJECTS */	special case, see below
/* IDENTIFICATION DATA ELEMENTS */	IDENTIFICATION
/* TELEMETRY DATA ELEMENTS */	TELEMETRY
/* HISTORY DATA ELEMENTS */	PDS_HISTORY
/* COMPRESSION RESULTS */	COMPRESSION_PARMS

- For VICAR labels, ODL comments (i.e., /* *string* */) are stored in a keyword named "PDS_COMMENT". This keyword appears in the VICAR property containing the elements immediately following the comment. When converting from VICAR to ODL, the comment is placed immediately before the group or class. Blank lines should surround the comment. Note that with IDS-generated EDR and RDR data products, multiple comment lines in a Group are not supported.

- The ODL objects IMAGE_HEADER and IMAGE, as well as the keywords in /* FILE DATA ELEMENTS */ and the ^IMAGE and ^IMAGE_HEADER pointers (in /* POINTERS TO DATA OBJECTS */) in the table above, do not map directly to VICAR. They all describe the layout of the file and the image data. The VICAR equivalent for all of these items is the VICAR System label. Information maps between these in a straightforward way. It should be trivial to construct a VICAR system label and the above-referenced ODL entities after referring to the respective file-format-definition documents. Note that the /* FILE DATA ELEMENTS */ and /* POINTERS TO DATA OBJECTS */ comments are constant and so are not mapped to PDS_COMMENT keywords in the VICAR label. They are inserted automatically as part of the system label conversion process.

- The statistics-related keywords in the ODL IMAGE object are MEAN, MEDIAN, MAXIMUM, MINIMUM, STANDARD_DEVIATION, and CHECKSUM. These keywords are never transferred to the VICAR label. For VICAR -> ODL conversion, they can be computed from the image, or simply omitted from the ODL image.

- A few remaining items in the ODL_IMAGE object are treated specially. The FIRST_LINE, FIRST_LINE_SAMPLE, INVALID_CONSTANT, and MISSING_CONSTANT keywords are transferred to the VICAR IMAGE_DATA property set.

- Any ODL keyword with a <*unit*> tag after the value is transferred to the VICAR label without the unit tag. A VICAR keyword with the same name, but with "__UNIT" (two underscores) appended to the end, is added with the value of the unit. So for example, the ODL keyword "EXPOSURE_TIME = 1.5 <s>" would translate to two VICAR keywords: "EXPOSURE_TIME = 1.5" and “EXPOSURE_TIME__UNIT = s". Note that because of this, any ODL keyword that can support a unit is limited to 25 characters. If there is more than one value (an array), a unit is associated with each. In this case, the “__UNIT” VICAR keyword becomes multi-valued also, with each unit copied in sequence. If one of the elements does not have a unit (but others do), the corresponding entry is "N/A" (which is not copied to the ODL label). So for example, ODL "CONTRIVED_ANGLE = (1.2 <rad>, 22.0, 54.1 <deg>)" would map to VICAR "CONTRIVED_ANGLE = (1.2, 22.0, 54.1)" and "CONTRIVED_ANGLE__UNIT = (rad, N/A, deg)".

- The VICAR history label is omitted from the ODL header.

3.2.6 Mapping of PDS4 and VICAR Labels

The information contained in the PDS4 and VICAR/ODL labels are semantically equivalent. MIPL uses the VICAR label exclusively for its processing; the PDS4 label is then generated from the VICAR label. This ensures consistency of label contents. In the event that additional VICAR/ODL 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/ODL 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. Please refer to Appendix B for the mapping between VICAR/ODL keywords and PDS4 attributes.

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. Cruicially, it has the ability to embed macros, which are used to extract VICAR label information. The Velocity templates are included with the PDS delivery.

3.2.7 SuperCam FITS files

SuperCam additionally stores EDR images in FITS format. These images have the same image content as the VICAR/ODL files, with different metadata. The metadata is stored in HDU (Header Data Unit) structures within the FITS file. It consists of instrument State of Health (SOH), housekeeping, autofocus, and similar information, in binary format as received in the telemetry stream. A copy of the VICAR label is also stored in an HDU.

TBD: better description of the specifics.
TBD: point at documentation from the SuperCam team? Non-imaging SIS, perhaps?

These FITS files are included in the PDS archive, but are not described further in this document.

3.3 Binary Data Storage Conventions

EDR and RDR data products for M2020 camera image 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 10-bit to 13-bit integers stored in signed 16-bit integers. The PDS and VICAR labels are stored as ASCII text.

3.3.1 Bit and Byte Ordering

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

For non-byte instrument data, which includes 8-bit unsigned shorts, 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 EDR/RDR product, the instrument data can have only one ordering, but it is dependent on the host platform where the data was processed. This follows both the ODL and VICAR file format conventions.

For all image data, the ODL label carries keyword SAMPLE_TYPE in the IMAGE Object to define which ordering is used in the file. The VICAR label carries keywords INTFMT and REALFMT in the System portion of the label to define the ordering. Both of these file formats specify that bit 0 is the least significant bit of a byte.

Table 3-3 – M2020 Image EDR/RDR Bit 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. EDR PRODUCT SPECIFICATION

M2020 instrument EDRs described in this document will be generated by JPL's Multimission Instrument Processing Laboratory (MIPL) under the IDS subsystem of the M2020 GDS element.

The data packaged in the image data products will be decoded, decompressed, and decompanded camera image data in single frame form as an Experiment Data Record (EDR). The EDR structures defined in this document may vary depending on instrument, with all containing attached metadata labels.

For M2020, the camera image EDR are the fundamental instrument data archive product. They will be generated as "raw" uncalibrated data within an automated pipeline process managed by MIPL under IDS at JPL. The size of an image EDR data product varies per instrument and is dependent on several factors (sub-framing, downsample, compression, etc.).

EDRs are versioned on M2020. When updated telemetery is received a new version of the EDR will be generated. Updated telemetry is possible when missing packets are retransmitted and/or telemetry is reflowed through the Ground Data System (GDS).

4.1 EDR General Processing

The EDR processing begins with the reconstruction of packetized telemetry data resident on the TDS by the Mission data Processing and Control Subsystem (MPCS) into a binary ".dat" data product and associated ".emd" Earth metadata file. The data product and metadata are written by MPCS to the Operations Cloud Store (OCS) and messages are generated on a Java Message Server (JMS) bus, where they are ingested by MIPL's EDR generation software "m2020edrgen" and processed with SPICE kernels provided by NAIF. The EDR will be generated after the notification describing the OCS location of the respective the binary data product and associated Earth metadata file has been received by the IDS pipeline system. The data flow from instrument to IDS is illustrated in Figure 4-1. The data flow within the IDS pipeline to EDR generation is illustrated in Figure 4-2 and is elaborated subsequently in this section.

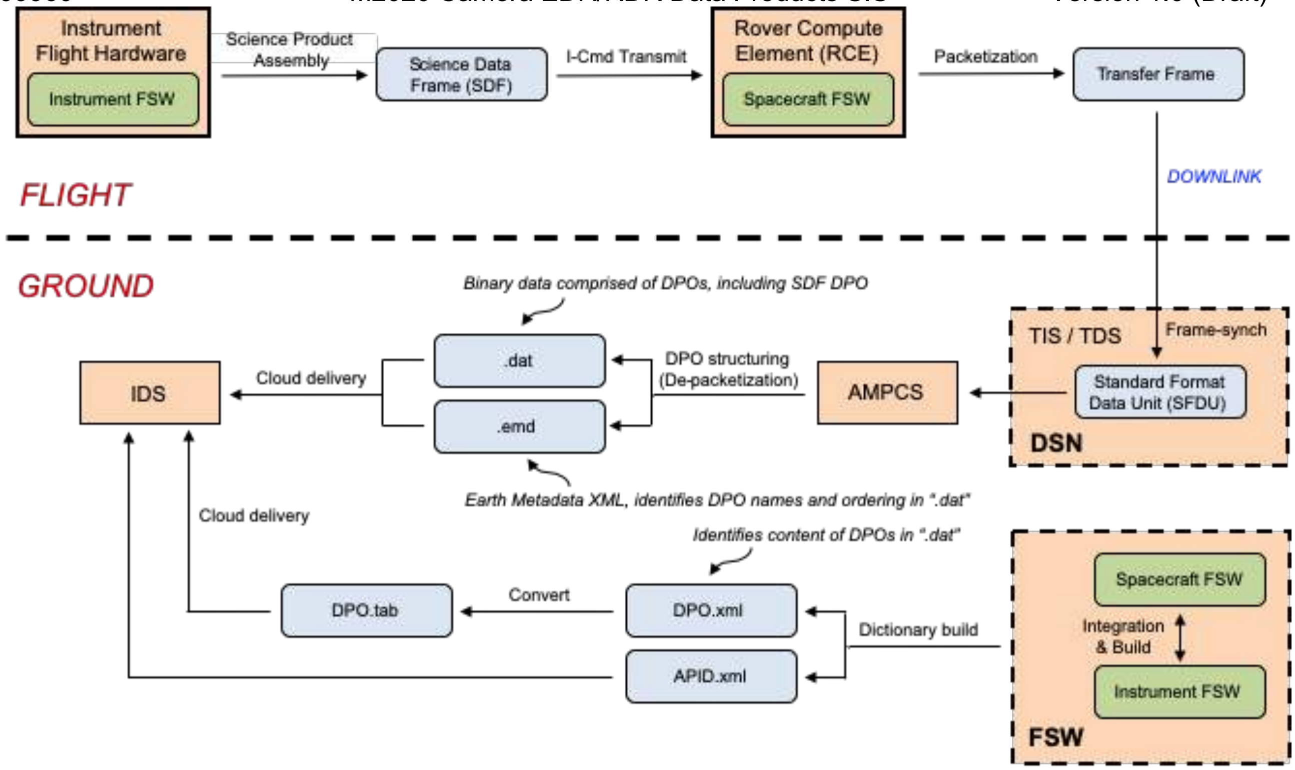


Figure 4-1 – Instrument Data Flow to IDS

This document has been reviewed and determined not to contain export controlled technical data.

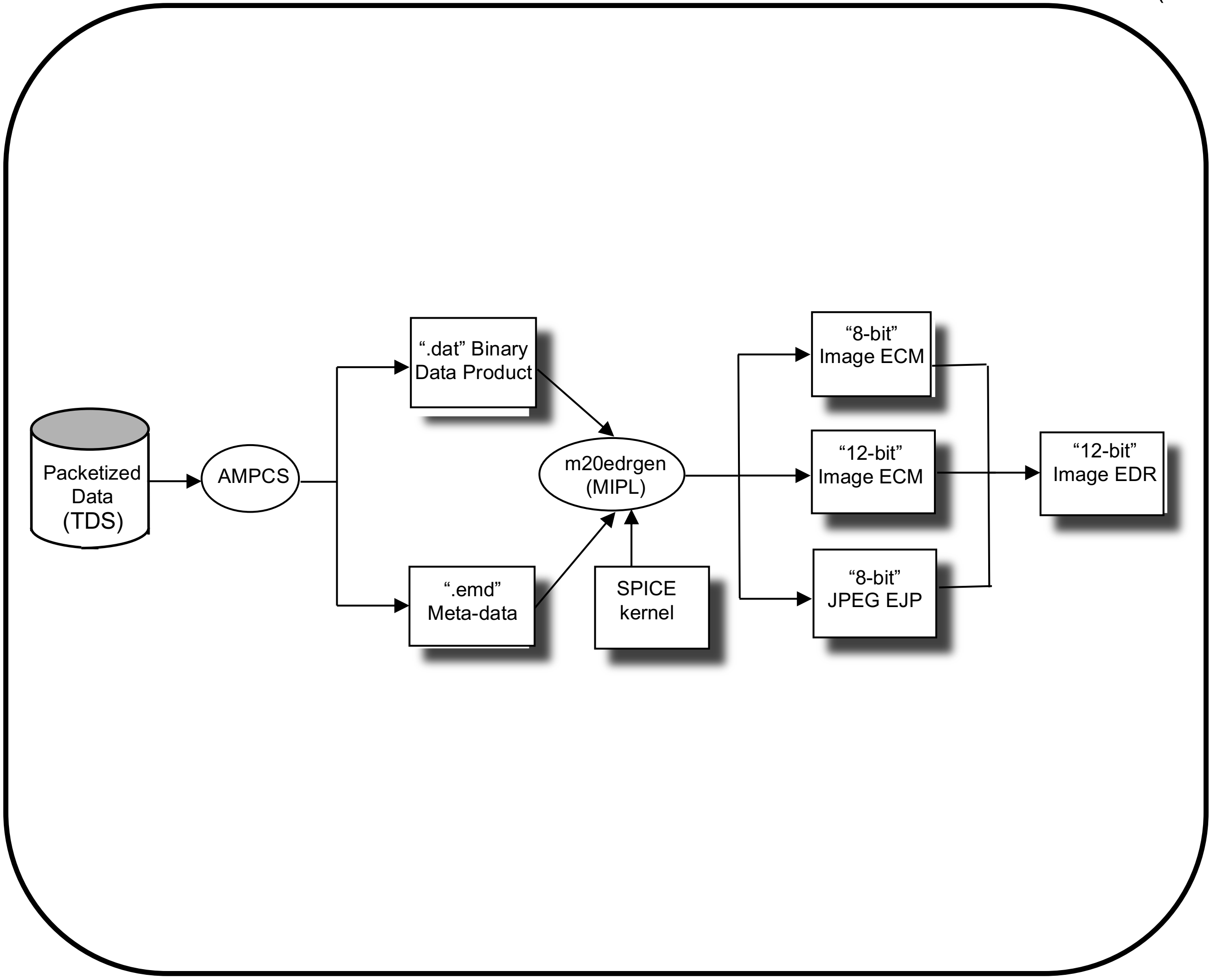


Figure 4-2 – IDS EDR generation data flow

In all EDR cases, missing packets will be identified and reported for retransmission to the ground as "partial datasets". Prior to retransmission, the missing EDR data will be filled with zeros. The EDR data will be reprocessed only after all "partial datasets" are retransmitted and received on the ground. In these cases, the EDR version will be incremented so as not to overwrite any previous EDR versions.

4.2 EDR Product Types

Descriptions for the various EDR product types are provided in this section. They are broken down into twp groupings: a) Image and b) Image Support. Refer also to Table 4-1 for a mapping between the source M2020 instrument and the EDR product type. Also refer to Tables 6-4 and 6-5 for expected priority of use and PDS processing levels.

4.2.1 Image EDR

Image EDRs are the closes representation possible to the data that came off of the imaging detector (CCD or CMOS). This means that compression used for transport has been removed (i.e. the products are uncompressed raster images), decompanding has been performed, and any irreversible operations done onboard (such as de-Bayering) are also part of the EDR. There are several flavors of EDRs for M2020, described in the following sections.

4.2.1.1 Standard EDR (ECM, EDR)

All of the imaging instruments create standard image ECM files. These are the raster-format, raw products as originally received, after being decompressed. Many image sensors are 10-13 bits in depth. In order to save downlink bits, these images are often “companded”, using a lookup table to convert the files to 8-bits. The ECM product thus may contain companded products, or the original products if companding was not done.

ECM files for PIXL MCC, MEDA Skycam, SHERLOC-ACI, LVS, and Heli-Nav are grayscale (single band). Those for ECAM, SuperCam RMI, ZCAM, SHERLOC-WATSON, the EDL cameras, and Heli-RTE may be either Bayer-encoded (1 band) or color (3 bands).

The ECAM Standard EDR is a special case because the detector can only be readout in tiles with a maximum size of 976 lines by 1296 samples. This results in an indiviual “Tile” EDR that can be “reconstructed” into the full image that was acquired on the detector if all tiles are readout and downlinked to Earth. See below.

4.2.1.2 De-companded EDR (EDR)

The “EDR” product (specific product name, not to be confused with the category) is the ECM file after being decompanded. It thus contains the full bit depth of the sensor, stored as 16-bit (signed) integers. Note that some cameras are natively 8-bit; they remain so in the EDR product. Note that the EDR is produced even if the ECM was not companded.

4.2.1.3 De-Bayered EDR (EBY)

If an image is Bayer-encoded, then an EBY file is produced with the de-Bayered result. EBY is thus a color image. EBY is not produced if the image is either not Bayer-encoded (grayscale imager or certain Mastcam-Z filters), or if de-Bayering has already been done onboard.

Note that unlike ECM/EDR, EBY does not distinguish the alternate file types (Z-stack, video, etc).

4.2.1.4 JPEG EDR (EJP)

EJP files contain the original JPEG stream as downlinked from the spacecraft; it has not been uncompressed or recompressed. This type is not produced unless JPEG compression was used onboard. It may be a color (three band) or grayscale (single band) JPEG. This EDR type applies to ECAM, SuperCam, ZCAM, SHERLOC, PIXL, EDL, and both Heli cameras. (LVS TBD)

4.2.1.5 Z-stack EDR (ECZ, EZS)

Z-stack images are created onboard by combining images taken at different focus settings. The result is a best-focus image with a much greater depth of field. Each pixel contains data from the source image(s) that are in best focus (potentially interpolated between neighboring images). This image may be color or a single band. This EDR type applies to SuperCam, ZCAM, and SHERLOC.

ECZ is the potentially companded original image (analogous to ECM) while EZS is the decompanded result (analogous to EDR).

NOTE: The metadata regarding vehicle state for Z-stack images reflects the state of the vehicle at the time the Z-stack product itself was created. This is generally not the same as the vehicle state at the time the imagery was acquired. Most importantly, this means the arm state and mast state, and thus camera model, for these images do not properly describe the image. The metadata must be obtained from the thumbnail or full frame of one of the images that went into making the Z-stack. Determining the proper image to use is beyond the scope of this document, but can often be inferred by inspection of the available data.

4.2.1.6 Depth Map EDR (EDM)

As part of Z-stack processing, a Depth Map image may also be produced. This 8-bit, single-band EDR indicates which image the best-focus data came from. Pixels with a digital number (DN) of 0 indicate the first image in the Z-stack, while a DN of 255 indicates the last, with linear scaling between. For example, with a 5-image stack and expected DNs of 0, 64, 128, 192, and 255, a DN of 96 would indicate the depth is halfway between images 2 and 3. This EDR type applies to SuperCam, ZCAM, and SHERLOC.

Depth maps are not decompanded or de-Bayered.

The metadata regarding vehicle state for Depth Map images reflects the state of the vehicle at the time the Depth Map product itself was created. This is generally not the same as the vehicle state at the time the imagery was acquired. Most importantly, this means the arm state and mast state, and thus camera model, for these images do not properly describe the image. The metadata must be obtained from the thumbnail or full frame of one of t images that went into making the Depth Map. Determining the proper image to use is beyond the scope of this document, but can often be inferred by inspection of the available data.

4.2.1.7 Video EDR (ECV, EVD)

The Video EDR is identical in format to the standard image EDR. It represents a single frame of a video sequence. The EDR type is separate in order to better distinguish video frames from still frames. This EDR type applies to ZCAM and SHERLOC.

ECV is the potentially companded original image (analogous to ECM) while EVD is the decompanded result (analogous to EDR).

4.2.1.8 Recovered EDR (ECR, ERD)

An Recovered EDR is an EDR for which the metadata is unavailable. It existed in camera memory but for whatever reason the onboard data product describing it was lost or deleted. Such products have extremely limited metadata. This EDR type applies to ZCAM and SHERLOC. Note that attempts are made in the pipeline to reconstruct the metadata using the appropriate thumbnail image. If successful, an ECR and a “normal” ECM may both appear for the same image.

ECR is the potentially companded original image (analogous to ECM) while ERD is the decompanded result (analogous to EDR).

4.2.1.9 Image Sampling Types

Except where otherwise noted, in general, the image types described above can be sent in one of three pixel sampling types: 1) Full-frame, 2) Subframe, or 3) Downsampled. They may also be sent as a Thumbnail, in addition to (or instead of) the three aforementioned formats.

4.2.1.9.1 Full Frame EDR

Full-frame EDRs contain the entire contents of the imaging detector at full resolution.

4.2.1.9.2 Subframe EDR

Sub-frame EDRs are a subset of rows and columns of the full-frame image. They can be thought of as a window on the detector, with the same resolution but smaller coverage area.

4.2.1.9.3 Downsampled EDR

Downsampled EDRs contain a smaller version of the image, resulting in reduced resolution of the same coverage area. Downsampling can be done via one of three methods: 1) nearest neighbor pixel averaging, 2) pixel averaging with outlier rejection, or 3) computing the median pixel value. Note that downsampling can be applied to subframes.

4.2.1.9.4 Thumbnail EDR

Thumbnails are a reduced-resolution version of the original image, sent in addition to, or instead of, the original image. They apply to all image EDRs. The main purpose of a Thumbnail EDR is to provide an image summary using a very low data volume compared to the original image. Decisions about downlinking the original image can be made using the Thumbnail. For the engineering camers and RMI, Thumbnails are produced relative to the full-frame image, even if the product is downsampled or subframed. For Mastcam-Z and SHERLOC camera images, thumbnails are of the subframe region only, and not the full-frame.

4.2.2 Image Support EDRs

4.2.2.1 Row Summation EDR (ERS)

A row summation EDR is the summing of the rows of a full-frame or subframed image and returning the results. The EDR is a nx1 array of 32-bit integers (whose length is equal to the image height) where the DN value of the ith element is the value of the sum of all the pixels in the ith row. Applicable to the Engineering Camera instrument suite.

4.2.2.2 Column Summation EDR (ECS)

A column summation EDR is the summing of the columns of a full-frame or sub-framed image and returning the results. The EDR is a 1xn array of 32-bit integers (whose length is equal to the image width) where the DN value of the ith element is the value of the sum of all the pixels in the ith column. Applicable to the Engineering Camera instrument suite.

4.2.2.3 Reference Pixel EDR (ERP)

The onboard detector array has "Reference" dark pixels located before and after the image data. For the SkyCam, there are 12 “pre-Reference” lines before the image lines and 12 “post-Reference” lines after the image lines. Also, there are 16 “pre-Reference” pixels before the image data in each row. Applicable to the SkyCam only.

4.2.2.4 Histogram EDR (EHG)

The histogram EDR is a 32-bit integer array storing the histogram of the image. A 1x256 or 1x4096 array will be returned. Applicable to the Engineering Camera instrument suite.

4.2.3 Engineering EDRs

These EDR types supplement the information available for various instruments. For more information, see the instrument-specific SIS for each instrument.

4.2.3.1 State of Health EDR (SOH)

The SOH EDR is comprised of binary metadata describing the health and safety of SuperCam RMI. They are stored in 16-bit signed integer format.

4.2.3.2 Non-Volitile Memory Telemetry Status Report EDR (E48)

The NVM telelemetry status report EDR contains the MCC system parameters as read from the NVM copy of the parameters in the MCC System memory.

4.2.3.3 RAM Telemetry Status Report EDR (E49)

The RAM telemetry status report EDR contains the MCC System Parameters as read from the RAM (volatile) copy of the parameters in the MCC System memory.

4.2.3.4 Translation Relative Navigation EDR (ESO)

This EDR contains the Translation Relative Navigation(TRN) data from the MCC.

4.2.3.5 Structured Light Illuminator Estimate EDR (ESF)

The PIXL SLI estimate EDR contains the distance and plane solutions derirved from SLI measurements.

4.2.3.6 Centroid Compression EDR (ESC)

The PIXL centroids image EDR contains the centroid positions (X,Y) and intensity of centroid on the CCD.

4.2.3.7 Circle Centroid Compression EDR (ESA)

The PIXL circle centroid EDR contains the estimated circle center and radius of the centroid.

4.2.3.8 Region of Interest EDR (ESR)

The PIXL ROI EDR contains a region around each centroid in uncompressed format. The ROI of centroids are coded sequentially.

4.2.3.9 Debug Info (ED1)

Sherloc debug info EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.10 Directory Dump (ED2)

Sherloc directory dump EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.11 Image Status (ED3)

Sherloc image status EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.12 Zstack List (ED4)

Sherloc z-stack list EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.13 Memory Dump (ED6)

Sherloc memory dump EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.14 Built in Self-Test (ED7)

Sherloc BIST EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.15 Util Test (EUT)

Sherloc util-test EDR is simply a pass-through of raw telemetry in binary format.

4.2.3.16 NavMap EDR

The NavMap products are engineering products used to assess the performance of the rover's on-board navigation software. All 3-char product identifiers starting with “N” (form: Nxx) are NavMap products.

The list of NavMap products and their short descriptions are provided in Table x. Most, but not all, are images of some sort in the standard VICAR/ODL/PDS format with varying data types, sizes, and numbers of bands. Because these are detailed engineering products, no further description of them is provided in this document. They will not be archived.

4.2.4 Other EDRs

4.2.4.1 Movie EDRs

The EDL cameras can produce movie files in MPEG format. For PDS, the MPEG file is delivered as a supplemental product, with the "science" data being extracted individual frames. The number and frequency of these individual frame extractions is TBD.

4.2.4.2 EDL Microphone

The EDL camera suite includes a microphone. The audio data is delivered in an uncompressed WAV file format. Specifics are TBD.

4.3 EDR Product Format

Description of EDR product formats in this section will be by instrument suite: a) Engineering Cameras, b) SuperCam RMI, c) Mastcam-Z, d) SHERLOC cameras, e) PIXL MCC, f) MEDA SkyCam, g) EDL cameras, h) LVS, i) Helicopter.

The EDR will be formatted according to this SIS, following the general terms of labeling and bit ordering previously discussed in Sections 3.2 and 3.3, respectively. This section details the specifics of a variety of formats across all images. The various EDR formats and their data sizes, across all instrument suites, are listed in Table 4-1 and discussed subsequently in this section:

Table 4-1 – List of EDR Types and Formats

Type	Product Identifier	Format (bits)	Instruments	Description
Original Image Product	ECM	8-bit or 16-bit unsigned	All cameras	Contains companded data either as originally downlinked or JPEG decompressed. It may be a color or grayscale.
JPEG	EJP	8-bit unsigned	ECAM, ZCAM, SHERLOC-WATSON, PIXL MCC, SkyCam, EDL, HELI	Contains JPEG-compressed data as originally downlinked. It may be a color or grayscale (single band) JPEG.
Video frame	ECV	8-bit or 16-bit unsigned	ZCAM and SHERLOC-WATSON	Identical in format to the Original Image Product. It represents a single frame of a video sequence.
Z-stack	ECZ	8-bit or 16-bit unsigned	ZCAM, SHERLOC-WATSON, RMI	A best-focus Z-stack image, from a combination of images at different focus settings, with a much greater depth of field. Typically SHERLOC-WATSON data. It may be color or a single band.
Depth map	EDM	8-bit unsigned	ZCAM and SHERLOC-WATSON	Part Z-stack processing. This single-band image indicates which image the best-focus data came from. DN of 0 indicates first image in the Z-stack, 255 is the last, with linear scaling between.
Recovered Product	ECR	8-bit or 16-bit unsigned	ZCAM and SHERLOC-WATSON	A Product for which the metadata is unavailable.
Thumbnail	ALL	16-bit signed integer	ECAM, SCAM-RMI, ZCAM, SHERLOC-	This data product is a spatially sized down version of an existing Full Frame, so is less than full size and less than full resolution.

Type	Product Identifier	Format (bits)	Instruments	Description
			WATSON, PIXL MCC, SkyCam, EDL,HELI	The bit scaling rules described for the Full Frame case above also apply here.
Original Image Product (De-companded)	EDR	16-bit signed integer	ECAM, SCAM-RMI, ZCAM, SHERLOC-WATSON, PIXL MCC, MEDA Skycam	Nominal data product. If “12 to 8-bit” scaling was perfomed, the 12-bit data has been unscaled back to 12 bits, stored as the last 12 bits of a 16-bit integer.
Video frame (De-companded)	EVD	16-bit signed integer	ZCAM and SHERLOC-WATSON	Identical in format to the Original Image Product. It represents a single frame of a video sequence.
Z-stack (De-companded)	EZS	16-bit signed integer	ZCAM, SHERLOC-WATSON, RMI	A best-focus Z-stack image, from a combination of images at different focus settings, with a much greater depth of field. Typically SHERLOC-WATSON data. It may be color or a single band.
Recovered Product (De-companded)	ERD	16-bit signed integer	ZCAM and SHERLOC-WATSON	A Product for which the metadata is unavailable.
De-bayered	EBY	16-bit signed integer	ECAM, ZCAM, SHERLOC-WATSON, PIXL MCC, SkyCam, EDL	Contains de-bayered color image.
Row Summed	ERS	32-bit unsigned	ECAM, SkyCam	nx1 array of 32-bit integers whose length is equal to image height, wherein the DN value for the Jth element equals the sum of all pixels in the Jth row.
Column Summed	ECS	32-bit unsigned	ECAM	1xn array of 32-bit integers whose length is equal to image width, wherein the DN value for the Jth element equals the sum of all pixels in the Jth column.
Reference Pixel	ERP	16-bit unsigned	SkyCam	Dark pixels bookending (pre- and post-) image pixels during serial register readout. There are “pre-” Reference and “post-” Reference pixels.
Histogram	EHG	32-bit unsigned	ECAM	DN histogram computed from image can have either 256 or 4096 bins, each capable of holding count values of up to 4,194,304.
IDPH only	EID	8-bit unsigned or 16-bit signed integer	ECAM	IDPH (Image Data Product Header) only, with no image data. The data is formatted as a 1x1 image with a 0 pixel value. This product is generated when the cameras are commanded to acquire a picture (for example, to pre-point the RSM), but no image data are requested from the camera. These types of products are intended to serve as metadata only and are typically used to identify RSM pre-point activities.
NVM Status RAM Status	E48 E49	32-bit unsigned integer	PIXL	Non-volitile memory or Random Access Memory system parameter dump.
TRN	ESO	CSV	PIXL	Translation Relative Navigation(TRN) data from the MCC
SLI Estimate	ESF	CSV	PIXL	The distance and plane solutions derived from SLI measurements.

Type	Product Identifier	Format (bits)	Instruments	Description
Centroid Image	ESA	32-bit unsigned integer	PIXL	Estimated circle center and radius of the centroid
Circle Centroid	ESC	CSV	PIXL	The centroid positions (X,Y) and intensity of centroid on the CCD.
Image ROI	ESR	32-bit unsigned integer	PIXL	Region around each centroid in uncompressed format. The ROI of centroids are coded sequentially.
State of Health	SOH	Variable depending on the data type of the engineering value	SCAM	Comprised of binary metadata describing the health and safety of SuperCam RMI
Debug Info	ED1		SRLC	Pass-through
Directory Dump	ED2		SRLC	Pass-through
Image Status	ED3		SRLC	Pass-through
Zstack List	ED4		SRLC	Pass-through
Memory Dump	ED6		SRLC	Pass-through
BIST	ED7		SRLC	Pass-through
Util-Test	EUT		SRLC	Pass-through
Movie	TBD	TBD	EDL	Movie files
Microphone	TBD	TBD	EDL	Audio files
Navigation Map	TBD	TBD	TBD	Multiple types TBD

4.3.1 Engineering Camera Instrument Suite

The binary content of the Engineering Camera image EDR data product is a copy of the scene that had been projected onto the camera instrument’s CMOS detector and readout from the camera memory buffer. The image data will be decoded and decompressed in single-frame form as the image EDR. Although the ECAM detectors and on-camera memory allow acquisition of images 3840x5120 pixels in size, hardware readout limitations require reading out “tiles” from the detector. This results in the full-frame EDR with a maximum dimension of 976 lines by 1296 samples.

The data from the Engineering Cameras can come down in one of five different ways: color or grayscale JPEG, 12-bit scaled to 8-bit, ICER compressed, losslessly-compressed (LOCO), or uncompressed. These methods apply across all of the image EDR types.

In the uncompressed case, 12-bit data will be stored “as is” in the 12 lowest bits of the signed 16-bit integer. In the 12-bit scaled case, 12-bit data scaled onboard to 8-bit via a “12 to 8-bit” Lookup Table (LUT) or, by bit shifting, will be downlinked as 8-bit data and stored that way, and also unscaled (decompanded) back to 12-bits and stored as the 12 lowest bits of the signed 16-bit integer. Data returned as compressed are JPEG, ICER, or LOCO encoded and will be decompressed as part of the EDR processing.

4.3.1.1 Bayer Pattern

The Engineering Cameras create images of 3840x5120 pixels. Each pixel on the CMOS detector has an individual red (R), green (G), or blue (B) filter arranged in a Bayer pattern, which consists of repeated squares of pixels:

(R,G)
(G,B)

This allows an RGB color picture to be taken in a single exposure.

4.3.1.2 ECAM Tiling

The engineering cameras for Mars 2020 are a significant upgrade compared to the MSL cameras. They are 5120x3840 pixels (vs. 1024x1024 on MSL), and have color Bayer-pattern filters.

However, due to limitations in the MSL-heritage flight software (FSW), images can be no larger than 1280x960 (actually 1296x976 to accommodate overlap). This means that images must be “tiled”.

Tiling means that only a portion of the image is read in at a time. The entire sensor plane is exposed at once and the results are stored onboard the camera. Separate image reads then pull out the pieces, or “tiles”, of interest. It takes 16 tiles to read out the full-resolution image, or 4 tiles at 2x downsampling. At 4x or 8x, the entire image can be read at once. Only one image can be stored on the camera at a time, so once the next exposure is performed, no more tiles can be read from the previous one.

In the surface FSW, used for most of the mission, this tiling is very flexible. Any arbitrary location can be read from the sensor, at any resolution and any color. Ground software reassembles these tiles (see Section 0) into the full image.

However, the cruise/EDL FSW is much more limited. This controls the vehicle through the critical EDL phase and the first few days after landing. In order to minimize changes to the cruise FSW, a mode was implemented to use “DC offset” to control which tile to read.

In general, DC offset specifies the analog value that is subtracted from the video signal prior to the analog/digital conversion. However, the cruise FSW co-opts this to specify which tile to read out. Note that only one tile can be read from any given exposure; thus building up a full-size, full-resolution image would require 16 separate exposures. The DC offset value corresponds to a lookup table that defines a specific readout mode and portion on the detector. The table has a range of 0-35 as defined below. Each tile is 960 lines by 1280 samples.

Table 4-2 – DC Offset specification for the Engineering Cameras

Table ID	FPGA Mode	Bayer Channel	Binning	Start Row	Start Sample
0	0	Bayer	None	0	0
1	0	Bayer	None	0	1280
2	0	Bayer	None	0	2560
3	0	Bayer	None	0	3840
4	0	Bayer	None	960	0
5	0	Bayer	None	960	1280
6	0	Bayer	None	960	2560
7	0	Bayer	None	960	3840
8	0	Bayer	None	1920	0
9	0	Bayer	None	1920	1280
10	0	Bayer	None	1920	2560
11	0	Bayer	None	1920	3840
12	0	Bayer	None	2880	0
13	0	Bayer	None	2880	1280
14	0	Bayer	None	2880	2560

15	0	Bayer	None	2880	3840
16	2	Red	2x2	0	0
17	2	Red	2x2	0	2560
18	2	Red	2x2	1920	0
19	2	Red	2x2	1920	2560
20	1	Green	2x2	0	0
21	1	Green	2x2	0	2560
22	1	Green	2x2	1920	0
23	1	Green	2x2	1920	2560
24	3	Blue	2x2	0	0
25	3	Blue	2x2	0	2560
26	3	Blue	2x2	1920	0
27	3	Blue	2x2	1920	2560
28	4	Panchro	2x2	0	0
29	4	Panchro	2x2	0	2560
30	4	Panchro	2x2	1920	0
31	4	Panchro	2x2	1920	2560
32	5	Red	4x4	0	0
33	6	Green	4x4	0	0
34	7	Blue	4x4	0	0
35	8	Panchro	4x4	0	0

4.3.2 SuperCam RMI EDR

The binary content of the RMI image EDR product is a copy of the scene that had been projected onto the camera instrument's CMOS detector onboard and read out from the camery memory buffer. Depending on the size of the image, the RMI science data could require up to 3 transfers from the instrument image buffer to the flight system. The flight system will concatenate the data together into one data product to be sent to Earth. The image data, telemetered on a variable number of bits (up to 13-bits uncompressed), will be decoded in single frame form and stored in the lowest 10 to 13-bits of a 16-bit integer as the image EDR. Data returned as compressed are ICER (color) or JPEG (color) encoded and will be decompressed as part of the EDR processing.

TBD: discuss the FITS files here? Or In 3.2.7? Or both? At least mention them both places.

4.3.3 Mastcam-Z (ZCAM)

The binary content of the ZCAM image EDR product is a copy of the scene that had been projected onto the camera instrument's CCD detector onboard and read out from the camery memory buffer. The data from the ZCAM cameras can come down in one of four different ways: color or grayscale JPEG, losslessly-compressed, or uncompressed. These methods apply across all of the image EDR types (standard image, Z-stack, depth map, video frame) and geometry types (full-frame, subframe, thumbnail).

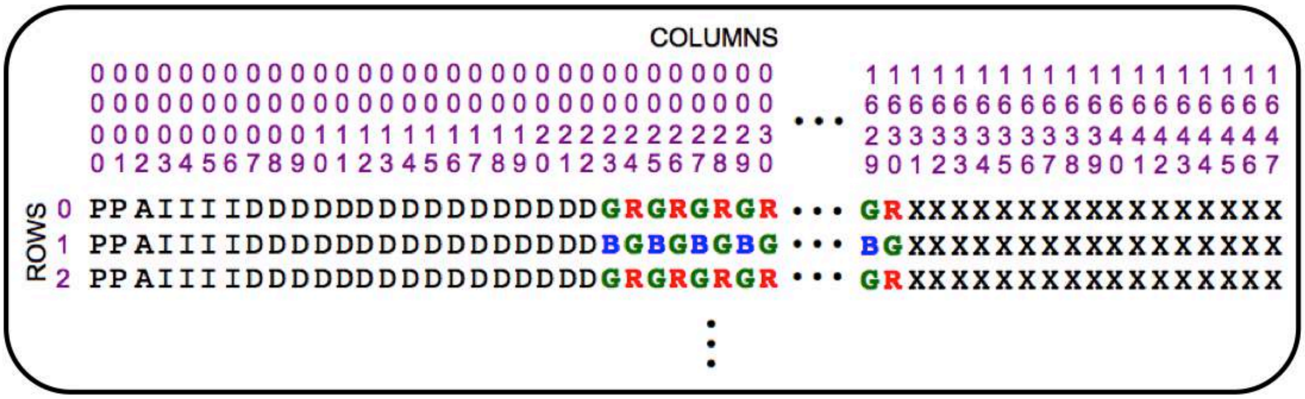
The Mastcam-Z cameras create images of 1648x1200 pixels. Each pixel on the CCD has an individual red (R), green (G), or blue (B) filter arranged in a Bayer pattern, which consists of repeated squares of pixels:

(R,G)
(G,B)

This allows an RGB color picture to be taken in a single exposure. Additional geology filters are available, which interact with the Bayer filters in complex ways. See Section 2.3 for an overview or the ZCAM camera documentation for details.

Of the 1648 pixels per line, only 1608 are photoactive pixels. The line structure is shown in Figure 4.3.3.1 below, and is broken down into the following:

- 2 dark pixels from the end of the **p**revious line (“P” in Figure 4.3.3.1)
- 1 invalid **A**DC pipeline pixel from the interline time (“A” in Figure 4.3.3.1)
- 4 **i**solation pixels (“I” in Figure 4.3.3.1)
- 16 **d**ark pixels (“D” in Figure 4.3.3.1)
- 1608 photoactive RGB pixels (“R”, “G”, “B” in Figure 4.3.3.1)
- 17 dark, isolation, and overscan pixels (“X” in Figure 4.3.3.1)



```
                          COLUMNS
       0000000000000000000000000000000       1111111111111111111
       0000000000000000000000000000000  ...  6666666666666666666
       0000000000111111111122222222223       2333333333344444444
       0123456789012345678901234567890       9012345678901234567
R   0  PPAIIIIDDDDDDDDDDDDDDDDGRGRGRGR  ...  GRXXXXXXXXXXXXXXXXX
O   1  PPAIIIIDDDDDDDDDDDDDDDDBGBGBGBG  ...  BGXXXXXXXXXXXXXXXXX
W   2  PPAIIIIDDDDDDDDDDDDDDDDGRGRGRGR  ...  GRXXXXXXXXXXXXXXXXX
S                                .
                                 .
                                 .
```

Figure 4-3 – ZCAM – RGB in Bayer Pattern Layout on CCD

Because there are an odd number of starting dark pixels, the first valid pixel (23) on even lines will be G (GRGRGR...) while on odd lines it will be B (BGBGBG...).

4.3.4 SHERLOC Camera Instrument Suite

The binary content of the SHERLOC image EDR product is a copy of the scene that had been projected onto the camera instrument’s CCD detector onboard and read out from the camera memory buffer. The data from the SHERLOC cameras can come down in one of four different ways: color or grayscale JPEG, losslessly-compressed, or uncompressed. These methods apply across all of the image EDR types (standard image, Z-stack, depth map, video frame) and geometry types (full-frame, subframe, thumbnail).

4.3.4.1 WATSON

The SHERLOC-WATSON camera creates images similar to Mastcam-Z. See Section 4.3.3for details.

4.3.4.2 ACI

The SHERLOC-ACI camera creates images similar to WATSON, but there is no CFA on the detector and all images will be grayscale.

4.3.5 PIXL MCC

The binary content of the the PIXL MCC image EDR product is a copy of the scene that had been projected on the the camera instrument’s CCD detector on-board and read out from the camera memeory buffer. The image data from the PIXL MCC comes down in one of two different ways: raw (uncompressed) or JPEG compressed.

4.3.6 MEDA Skycam

The binary content of the the MEDA image EDR product is a copy of the scene that had been projected on the the camera instrument's CCD detector on-board and read out from the camera memory buffer. The data from the Skycam come down in one of two different ways: uncompressed or compressed. Data returned as compressed are ICER or LOCO encoded and will be decompressed as part of EDR processing.

4.3.7 EDL Camera Suite

The binary content of the EDL cameras image EDR product is a copy of the scene that had been projected onto the camera instrument's CCD detector onboard and read out from the camera memory buffer. The imaging data from the EDL cameras can come down in one of 2 ways: color or greyscale JPEG and compressed MPEG. For each MPEG file, imaging products are generated by converting every "n-th" frame into an EDR image product where "n" is determined by setting a parameter to the pipeline or EDRgen. The frame extraction is performed using the FFMpeg framework.

4.3.8 LVS

TBD

4.3.9 Helicopter

The data from the helicopter cameras are read into onboard helicopter memory. After the flight is over, the data are transmitted to the rover for relay to Earth. Images may be JPEG compressed or uncompressed. There is no facility to retrieve the raw Bayer pattern from the RTE camera.

The RTE camera is 4224x3120 pixels, but only the first 4208 pixels are active. There is no data in the last 16 columns (4209-4224).

4.4 EDR Product Structure

The description of EDR product structures in this section will be by EDR type: Image EDR and Image Support EDR.

This section details the specifics of a variety of EDR structures, taking into account the concept of product labels and the product's binary content described previously in Sections 3.2 and 3.3, respectively.

Figure 4-3 shows the format of an EDR: an image EDR that has a VICAR label wrapped by an ODL label, Diagram A. Shown with detached ASCII PDS4-compliant labels for archive purposes.

For a description of the ODL labels, see Section 3.2.2, and for a description of the VICAR Label, see Section 3.2.3, and for a mapping between ODL and VICAR, see Section 3.2.4.

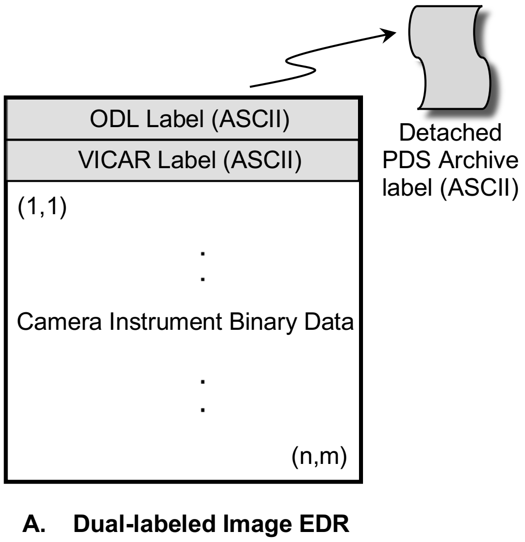

ODL Label (ASCII)
VICAR Label (ASCII)
(1,1)
Camera Instrument Binary Data
(n,m)
Detached PDS Archive label (ASCII)
A. Dual-labeled Image EDR

Figure 4-4 – Dual Label EDR with detached PDS4 label

4.4.1 Image EDR

All image EDRs will have the same structure as decribed in this section. Some of the camera EDR products will be rotated so that the origin (1,1) is not the same as the detector origin. Such rotations preserve the natural concept of what is “up” in the image.

All image EDRs for the M2020 Cameras contain the four components shown in Figure 4.4. They are listed below. The first three components comprise the image file and consist of an integral number of “records”, where the record size is the size needed to contain one line of image data. The fourth attribute is an external (detached) label file for archive purposes.

1) An operations label in ODL (ASCII) format
2) An operations label in VICAR (ASCII) format
3) An n x m block of binary image data with the origin at the upper left pixel in line (row) 1, sample (column) 1.
4) A detached PDS4-compliant label, in a separate file. It may not be available during operations but will be created for the PDS4 archive.

4.4.2 Image Support EDR

The Row Summation, Column Summation, Reference Pixel, and Histogram support EDRs have the same format as the Image EDRs described above in Section 4.4.1.

4.4.3 Engineering EDRs

TBD: Are these all properly part of the non-camera SISs, rather than here? Doesn't hurt to mention them here but if so we need to state that.

4.4.3.1 SuperCam

4.4.3.1.1 State of Health (SOH)

The State of Health (SOH) is comprised of binary metadata describing the health and safety of SuperCam instrumentation. There are five variants: a) State-of-Health, b) Power On, c) Power Off, d) Sun Safe and e) Initialize. Stored in 16-bit unsigned integer format, they are a vector of 9 entities for the ChemCam Body Unit and a vector of 39 entities for the ChemCam Mast Unit. Applicable to the ChemCam instrument suite as a standalone product.

TBD: FITS files…

4.4.3.2 PIXL

All PIXL MCC Engineering EDRs start with 34-byte image header taken directly from the first TM packet from the MCC, followed by variable size PIXL Specific System Parameter list and ended with Meta Data(if any) as provided by MCC.

4.4.3.2.1 MCC Telemetry Status Report - Non-volitile Memory

This Data Product contains the MCC System Parameters as read from the Non-volitile Memory (NVM) copy of the parameters in MCC System memory.

4.4.3.2.2 MCC Telemetry Status Report - RAM

This Data Product contains the MCC System Parameters as read from the RAM (volatile) copy of the parameters in MCC System memory.

4.4.3.2.3 MCC OLM Translation Relative Navigation(TRN) Estimate

This Data Product contains the Translation Relative Navigation(TRN) from the MCC. The OLM compression executes the autonomous terrain relative navigation algorithm by extracting and matching keypoints between successive images. It is stored in CSV format.

4.4.3.2.4 MCC Structured Light Illuminator(SLI) Estimate

The Structured Light Illuminator(SLI) compression executes the SLI algorithm that acquires distances to the observed terrain, using the SLI pattern. The SLI image compression structure contains the distance and plane solution derived from the SLI measurements. It consists of a 26 bytes SLI header followed by Nx24 bytes CenSLI data listed sequentially.

4.4.3.2.5 MCC Centroids Image

This Data Product contains the Centroid compressed image from the MCC in CSV format.
The centroids are listed sequentially. The centroids compression returns only the centroid positions on the CCD. One centroid is 10 bytes long. The bytes 0-3 are X position, bytes 4-7 are Y position both coded as 4 byte unsigned longs. In order to convert to image coordinates, these integers must be divided by 5000000. Bytes 8-9 are the intensity of the centroid with valid data range of 0-320000.

4.4.3.2.6 MCC Circle Centroid

This data product contains the Circle Centroiding(CC) solution as the estimated circle center, radius, and relevant information.

The Circle Centroid(CC) compression executes the autonomous Circle Centroiding algorithm that acquires absolute navigation information, by identifying the location of the circular abraded area. The output TM packets or image data will be different depending on the configuration.

4.4.3.2.7 MCC Image Region of Interest (ROI)

The Region of Interest(ROI) compression returns a region around each centroid in uncompressed format. ROIs are stored as a series of bitmapped images around the detected Centroids. The ROI of the centroids in the image are coded sequentially in the image file. The image data for one centroid is:
Bytes 0-1: The X coordinate of the upper left corner of the centroid.
Bytes 2-3: The Y coordinate of the upper left corner of the centroid.
Bytes 4-5: The width of the ROI (ROI_W)
Bytes 6-7: The Height of the ROI(ROI_H)
Bytes 8-N: The image data of the ROI starting from upper left corner. N=ROI_W * ROI_H

4.4.3.3 Sherloc

All Sherloc Engineering EDR products are pass-through products that have been renamed according to EDR naming convention and generated detached ODL label. The ODL label describes the DPO list in addition to the other label items.

4.4.3.4 Engineering Cameras

4.4.3.4.1 Nav Map

The NavMap products are engineering products used to assess the performance of the rover's onboard navigation software. All 3-char product identifiers starting with “N” (form: Nxx) are NavMap products.

The list of NavMap products and their short descriptions are provided in Table 4.1. Most, but not all, are images of some sort in the standard VICAR/ODL format with varying data types, sizes, and numbers of bands. Because these are detailed engineering products, no further description of them is provided in this document. They will not be archived. TBD

4.5 EDR Product Validation

Validation of the M2020 EDRs will fall into two primary categories: automated and manual. Automated validation will be performed on every EDR product produced for the mission. Manual validation will only be performed on a subset.

Automated validation will be performed as a part of the archiving process and will be done simultaneously with the archive volume validation. 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 PDS syntax of the label, a check of the label values against the database and against the index tables included on the archive volume, and checks for internal consistency of the label items. The latter include such things as verifying that the product creation date is later than the earth received time, and comparing the geometry pointing information with the specified target. 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 through-out the life of the mission, and comprehensive validation of a sub-set 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.

5. FDR PRODUCT SPECIFICATION ***NOT PART OF PEER REVIEW***

Fundamental Data Record (FDR) data products described in this document will be generated by MIPL personnel using the Mars Suite of VICAR image processing software within IDS at JPL. The FDR product type is new for M2020 and represents the "base image" from which all other RDRs derive.

5.1 FDR General Processing

Processing for each FDR may be different for each image or instrument. In general, the FDR product is derived (copied) from either the EDR or the EBY product types. At all times the EDR will be available and will serve as the fundamental archive product. The FDR maintains a consistent starting point for all RDR processing. In the special case of ECAM, the FDR will be a reassembled image from all available fundamental tile data records (TDR) from a single image pointing. The FDR will be upscaled to the highest resolution (downsample) TDR available in the reconstruction. The ECAM FDR Processing is described in detail in section 5.2.

In a few cases (TBD: list) the FDR serves as a place to install metadata that was not available at the time the EDR was created, such as when the information comes down in a separate data product.

<TBD: ADD DIAGRAM to show EDR to FDR processing flow (like MSL)>

5.2 ECAM FDR Processing (Reconstructed Images)

The Engineering Camera Suite on M2020 received a significant upgrade from MSL to a color (Bayer filter) 20 megapixel (MP) CMOS detector. Due to legacy (MSL) hardware restrictions, the M2020 RCE can only process ~1 megapixel at a time. Each individual portion of the detector that is readout is refered to as a "tile" which is described in Section 4.3.1.2 The maximum tile dimensions that can be read from the camera hardware is 976 lines by 1296 samples. This means that to downlink a complete image at full-resolution, the FSW has to read out 16 separate tiles (960 lines x1280 samples, plus an 8-pixel overlapping border on each side) and package them individually for downlink to the ground. Additionally, there are 4 flavors of readout modes from the camera hardware, all relating to the level of downsampling used to acquire the tile: 1x1, 2x2, 4x4, and 8x8. A single exposure can include any or all of the readout modes, which will have to be accounted for on the ground and reassembled (reconstructed). The IDS pipeline will collect all of the TDRs which form a single exposure and reconstruct the image based on the tiles. As more tiles become available on the ground, the reconstructed image is updated with the new information. There are several nuances to the reconstruction, all of which are described by the "camera specific" portion of the filename.

5.2.1 Tile Reconstruction Types

TBD: describe multi-res, and the single-res reconstructions, and use of the recon counter. IDM and IDX files.

5.2.2 ECAM Fundamental Tile Data Record (TDR)

The TDR represents the "base image" from which all other tile RDRs derive. TBD: Add description of TDR and how it forms the FDR and why the ECAM Reconstructed FDR is necessary.

6. RDR PRODUCT SPECIFICATION *** NOT PART OF PEER REVIEW ***

RDR data products described in this document will be generated by MIPL personnel using the Mars Suite of VICAR image processing software within IDS at JPL and the SuperCam Science Team using team-developed software at Los Alamos National Laboratory (LANL) in New Mexico, Institut de Recherche en Astrophysique et Planétologie (IRAP) and Centre National d'Etudes spatiales (CNES) in France. The RDRs produced will be “processed” data. 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 RDR data product will be placed into FEI for distribution.

6.1 RDR General Processing

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.

6.1.1 Image RDRs

Image RDR data products described in this section will be generated by the EDR/RDR data product pipeline operating under the IDS at JPL. Pipeline processing will include

For Engineering Camera and ZCAM camera data, and to a lesser degree SuperCam RMI data, RDRs will be processed using the Mars Suite subset of the VICAR image processing software developed at MIPL. The various image RDR interfaces in the VICAR processing flow are shown in Figure 6-1, and their descriptions are described subsequently in Section 6.2.

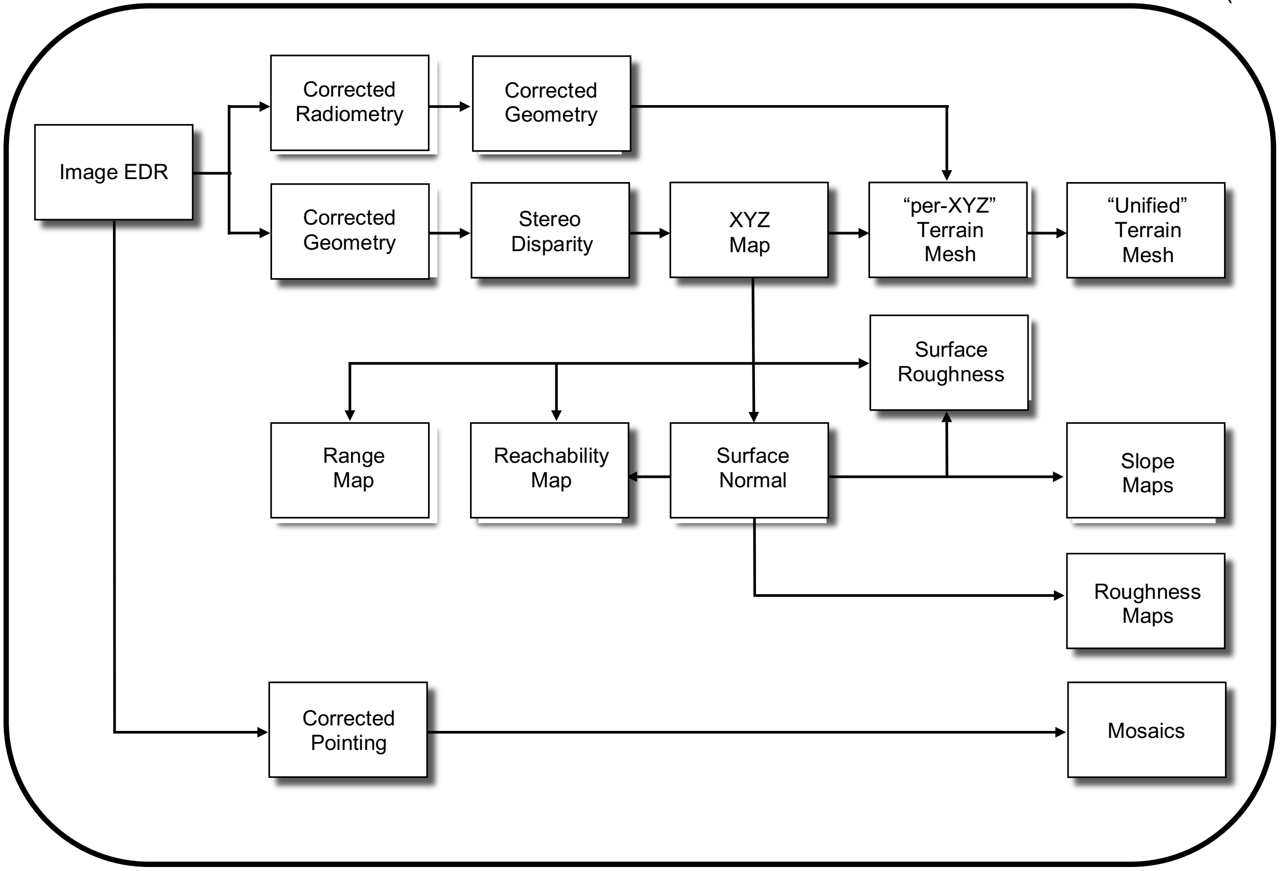


Figure 6-1 – VICAR image processing flow diagram

6.1.1.1 Common Processing

Although the various camera instrument product types are specific in nature, certain types of processing are often common to different RDR base files. These are described below.

6.1.1.1.1 Geometrically Corrected Images (Linearization)

TBD: Rework, given that we are not doing much linearization now.

EDRs and single-frame RDRs are described by a camera model. This model, represented by a set of vectors and numbers, permit a point in space to be traced into the image plane, and vice-versa. Linearization mode is indicated by the "geom" flag in the filename.

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 this imagery. 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.

The Navcam, SuperCam, and ZCAM cameras use a CAHVOR model, while the Hazcams use 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 6-2 for a visual comparison between a Front Hazcam image EDR (left) and the Front Hazcam "linearized" image RDR (right).

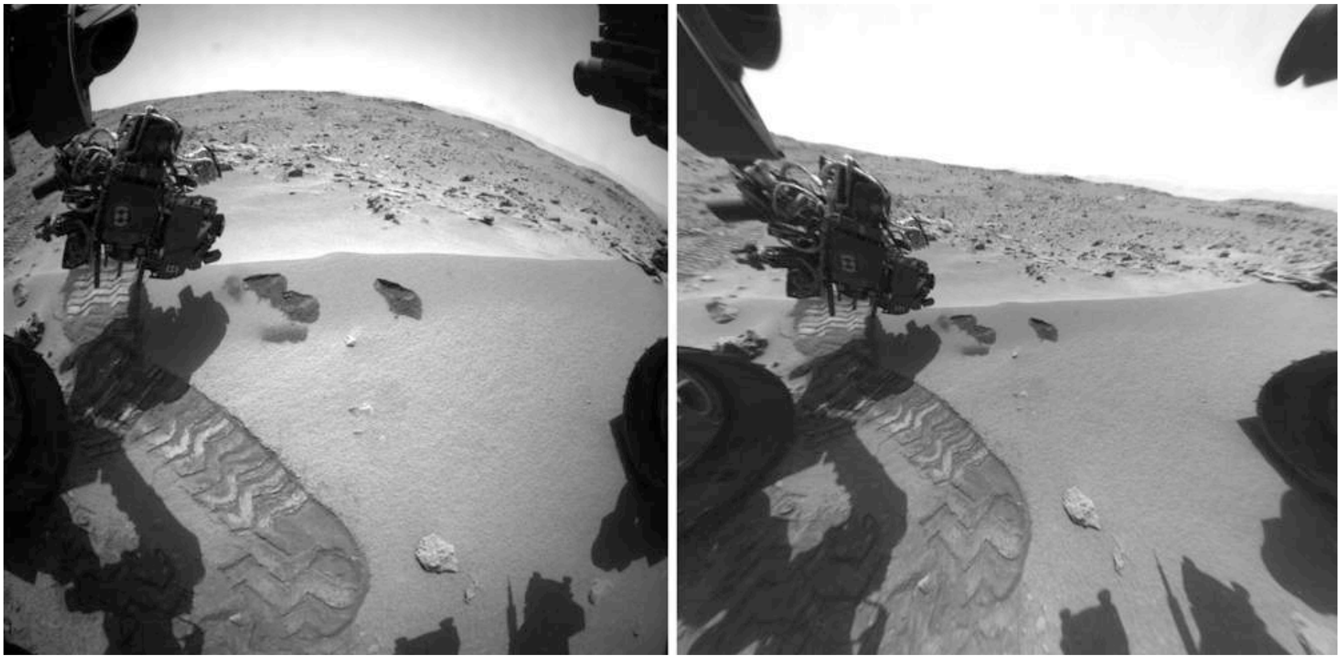


Figure 6-2 – FHAZ EDR (left) vs FHAZ Linearized RDR (right)

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, 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 (see the references).

The linearized CAHV camera model is derived from the EDR's 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 the references.

There are two types of linearization, indicated in the filename. For the nominal case, each image is linearized using a virtual camera model constructed to indicate what the other eye's model would look like at the same pointing. 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, this virtual partner is not always appropriate. In some cases the stereo pair is composed of images acquired at different pointings (often needed for very close-range mast work in order to achieve sufficient overlap), different positions (for an arm-based camera such as SHERLOC-WATSON), different rover locations (long-baseline stereo, used to resolve distances far greater than a normal stereo pair can do), different conditions (such as different focus or zoom on the

Mastcam-Z), or even different instruments. These cases require linearizing with the actual stereo partner.

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 Hazcams and is a 1m radius sphere centered on the camera), and projecting the pixel back through the CAHV model into the output image.

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, the horizontal center
V - A vector pointing roughtly downward in the image; it is a composite of the orientation of the CCD columns, the vertical scale, the vertical center, and A.

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 [Ref 18] through [Ref 27]. 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.

6.1.1.1.2 Image Overlays

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 overlaid on a background of the EDR or other intensity image, using color coding to represent the RDR value. See Figures 5.2.1.4 (XYZ overlay), 5.2.1.5 (Range overlay) and 5.2.1.9.1 (Arm Reachability overlay) for examples. This gives a visualization of the RDR in context of the scene. Overlays are generally indicated by a product type ending with “O” in the filename.

6.1.1.1.3 Filled Images

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 (see 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 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).

6.1.1.1.4 Color Processed Images

The ZCAM cameras routinely produce color images via their Bayer-pattern CCD's. The de-Bayering is typically done onboard but can also be done on the ground if not. In any case, RDRs that contain image data, such as ILT, RAD, RAS, etc., are produced by pulling the three bands apart into separate files, processing each independently, and then combining them into color again. Both the color image and one or more of the individual bands are kept as end products, as specified in the Config flag (3rd character) in the filename.

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 higher resolution image.

Note that not all ZCAM images are color; in particular some of the geology filters for the Mastcam-Z produce single-band images, which are processed like any other single-band image.

6.1.1.1.5 Rover Volume Exclusion Image Masks

For the purposes of Terrain Mesh RDR generation, IDS will create “Rover Volume Exclusion Mask” files that can be applied to several types of RDR (primarily XYZ). They are used to filter out rover features from generated terrain products, as well as the horizon and other undesirable features. 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. These mask files typically have an “M” as the first character of the product type.

6.1.1.1.6 Masked Images

The exclusion masks, or other masks, can be combined with the RDR to create a Masked image. The contents are identical to the source RDR (most often, XYZ) except that where the mask is 255, the value is set to the value specified by MISSING_CONSTANT. Masked files typically have an “M” as the third character of the product type (exception: masked ARM is called ARK).

6.1.1.1.7 EDR-like RDRs

The LIN and BAY RDRs are exactly like EDRs except they have had linearization or de-Bayer processing done to them. This processing can be done to any image and does not change the product type code; the config, geom, and/or samp fields change instead. The LIN and BAY product type codes exist simply to ensure that the “EDR” product type code is not used for any RDRs.

6.2 RDR Product Types

Descriptions for the various RDR product types are provided in this section. Refer also to Tables 5.4.1 and 5.4.2 for a concise list of the RDR product types.

6.2.1 Image RDRs

6.2.1.1 “Inverse LUT” RDR *Deprecated as the EDR is now decompanded

The ILT RDR is produced by IDS to reconstruct the original 12-bit pixels generated by the camera sensors. If the EDR is already in 12-bit format, ILT is simply a copy. However, if the EDR is in “8-bit” format (see Section 4.4) as a result of onboard “12 to 8-bit” scaling using a Lookup Table (LUT) or bit shifting, then an Inverse LUT (ILUT) is to be used to rescale the 8 lowest bits to the 12 lowest bits in the 16-bit signed integer.

The ILC type has had additional processing to remove certain instrument artifacts. At launch for M2020, this type is not used. However, operational experience with MER shows that over time, cameras degrade, with noise appearing due to radiation exposure and other effects. ILC is thus a placeholder for processing such as despiking, which removes this kind of noise, or interpolation of dead pixels.

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

6.2.1.2 “Radiometrically Corrected” RDR

There are 3 different kinds of radiometrically corrected products. “RA*” have been corrected to absolute radiance units of W/m^2/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. Within each kind, the values may be represented as floating-point values, or scaled to integers for ease of manipulaton.

Independent of type, there are multiple methods of performing radiometric correction, distinguished by the RADIOMETRIC_CORRECTION_TYPE keyword. The two described here are CHEMRAD and MIPLRAD.

6.2.1.2.1 Zenith correction

TBD

6.2.1.2.2 CHEMRAD Method

TBD: Verify this is still current (not just MSL copy)

This refers to partial radiometric correction of SuperCam RMI image data performed using an image processing toolkit developed in IDL by SuperCam Science Team members at IRAP and CNES in France. It is deemed “partial” in the sense that the raw image data are not calibrated to physical radiance units. Although the calibration portion of radiometric correction is not performed for the input EDR, other corrections are applied. These corrections are a function of the image geometry, exposure time (milliseconds), sensor temperature (deg Celsius), and target distance (meters). The latter three are extractable from the SOH portion of the EDR file.

The main IDL procedure is RMI_PROCESSING.pro, which can apply successive corrections to RMI EDR image defects by incorporating, in a specific order, several lookup tables (LUTs) constructed during ground calibrations. The processing flow is shown in Figure 6-3.

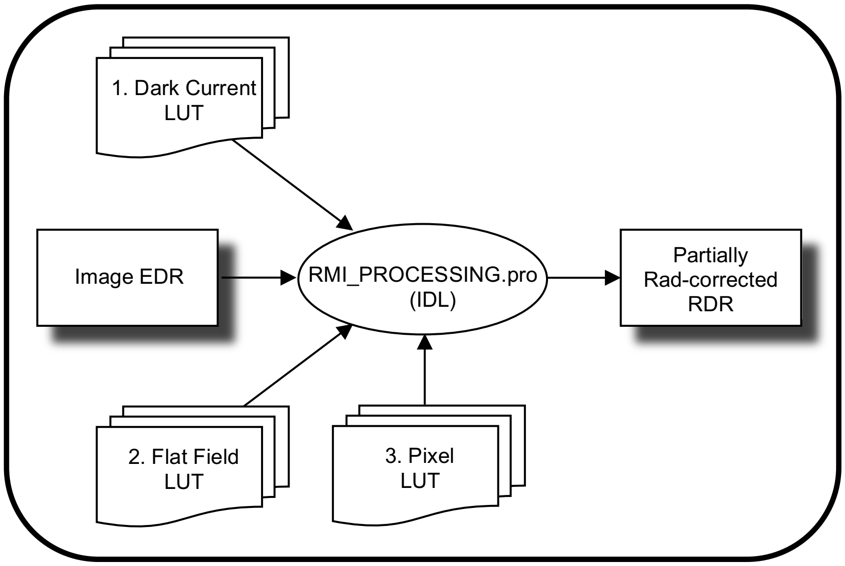


Figure 6-3 – RMI radiometric processing flow diagram

During the ground calibrations, several lookup tables were built with the constant values necessary to compute the corrections; they are included with the RMI_PROCESSING.pro code. These lookup tables will need to be revised with in-flight calibrations at a frequency to be defined.

The code applies corrective image processing steps in the reverse order of the apparition of the defects in the image, since one effect can influence the subsequent ones. In order, the corrections are as follows:

1. Subtract the estimated dark current image (temperature, time). This step incorporates a dark current LUT.
2. Iteratively (row by row) subtract the smearing drag (time) due to the CCD.
3. Iteratively (row by row) subtract the ghost image that is linked to the LIBS telescope and optics.
4. Divide by the estimated flat field that is linked to the LIBS telescope and optics. This step incorporates a flat field LUT.
5. Correct for known bad pixels by replacing them with median of neighboring pixels. This step incorporates a pixel LUT.
6. Apply a mask corresponding to the footprint of the telescope on the useful part of the CCD detector.
7. Enhance the contrast of the corrected image using histogram equalization.

The resulting RDR product is only partially radiometrically corrected and is not calibrated into physical radiance units. Note that the default flat field LUT is derived from a sky observation on Sol 32. Data acquired previous to Sol 32 were reprocessed using the Sol 32 flat field LUT.

Effects not accounted for in the correction process include:

a) Non-linearity - Each pixel of the CCD is a potential well, in which electrons are trapped. The number of trapped electrons is an affine function of the lighting, but above a certain threshold the linearity is broken (saturation). At room temperatures, at gain 12, the non-linearity starts around 700 counts, and saturation is reached around 850 counts, i.e. an integration of about 10 ms. This response function of the detector is a complex function of the temperature and can vary from pixel to pixel. It has not been calibrated since the primary objective of the RMI is to give a qualitative context of the LIBS experiment.

b) Direct Light - Some light from the far field can reach the detector without passing by the telescope mirrors, despite the baffle on the secondary mirror. It can be seen as extra light in the bottom part of some images, and sometimes as the focused picture of the distant scene. The diffuse light is taken into account in the determination of the reference flat field, and thus rather well corrected, but the focused distant scene cannot be corrected. However, this rare and faint, except in the corners that are removed when the telescope mask is applied.

6.2.1.2.3 MIPLRAD Method

This refers to radiometric correction of any camera instrument data systematically performed by MIPL (IDS at JPL) to meet tactical time constraints imposed by rover planners. The exception is SuperCam RMI data, which were processed only using the CHEMRAD method described in the previous section. 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. It then applies the following corrections: flat field, exposure time, temperature-compensated responsivity. The result is calibrated to physical units of W/m^2/nm/sr. The actual algorithm and equations used for MIPLRAD are shown below. Not every step applies to each camera type. Each correction is applied in sequence, to every pixel:

1. For the Engineering Camera instrument suite only, if on-board flat-fielding has been applied, it is backed out according to the parameters in FLAT_FIELD_CORRECTION_PARM, which is described in Appendix F and defines ff(x,y).

   output(x,y) = input(x,y) / ff(x,y)

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

   output(x,y) = input(x,y) / flat_field(x’,y’)

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

   output(x,y) = input(x,y) / exposure_time

4. The temperature responsivity is removed next. The temperature of the CCD is determined from INSTRUMENT_TEMPERATURE using the following rules, where the first valid temperature found (0.0 is ignored as no-reading, and >50C is considered broken) is the one used:

   a) Use the CCD temp of said camera
   b) Use the CCD temp of left/right partner
   c) Use the CCD temp of alternate A/B side same-eye camera
   d) Use the CCD temp of alternate A/B side left/right partner
   e) Use the CCD temp of “similar” camera (other Hazcam, other mast-mount) in a-d order
   f) Use the CCD temp of any camera
   g) Use electronics temp instead of CCD in a-f order

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 F) according to the following formula:

output(x,y) = input(x,y) * (R0 + R1*temp + R2*temp*temp)

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

   output(x,y) = (input(x,y) - RADIANCE_OFFSET) / RADIANCE_SCALING_FACTOR + 0.5

Note that the engineering cameras were not well calibrated radiometrically. Specifically, only flat fields were obtained, not temperature coefficients. Since they are build-to-print copies of the MER engineering cameras, the MER temperature responsivity parameters are used. For ZCAM, no temperature compensation is applied by the MIPLRAD method.

6.2.1.3 Color Correction

TBD… take from InSight

6.2.1.4 “Disparity” RDR

TBD: Potentially rework to better describe the stereo algorithm, LR-RL checking, and masking

A Disparity file contains 2 bands of 32-bit floating point numbers in the Band Sequential order (line, sample). Alternatively, line and sample may be stored in separate single-band files.

The parallax, or difference measured in pixels, between an object 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. 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, the coordinates in the reference image are the same as the *coordinates* in the Disparity image, and the matching coordinates in the stereo partner image are the *values* is the Disparity image. 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. These 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 [Ref 30] and represents the intermediate step between stages. It contains coordinate values, as in the primary disparity.

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 contain 2 bands of floating-point numbers in (line, sample) order using the Band Sequential format. The line and sample components may be stored in separate single-band files.

Raw files contain the results before doing left->right and right->left reconciliation. They should be considered an intermediate result.

Disparity error metric files contain information about the quality of the correlation match. The tools to create them have not yet been developed; they are listed here as a placeholder for future expansion. Thus the format has not yet been determined as of this writing.

Grid overlays 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).

6.2.1.4.1 Stereo Pair Matching Method

Inherent in the designed operation of the stereo cameras is time-sychronization in the acquisition of left and right images intended for stereo processing. So, the SCLK timestamps of the respective left and right image acquired as a stereo pair onboard will be used to automatically identify them as a stereo image pair during nominal ground data processing.

TBD: Rework this to talk about stereo matching flag and off-nominal stereo processing

Occasionally, stereo pairs will need to be processed that were not acquired simultaneously. This could be due to a sequence error, or for special operations such as re-pointing the mast between frames or driving the rover (long-baseline stereo). These off-nominal stereo pairs will be identified manually during ops, and processed as necessary. The special processing flag will be used to identify these. Additionally, most will use the “Actual” linearization method rather than “Nominal”.

6.2.1.5 “XYZ” RDR

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 as a X Component RDR, Y Component RDR and Z Component RDR, respectively. The single component RDRs are implicitly the same as the XYZ file, which is described below. XYZ locations in all coordinate frames for M2020 are expressed in meters unless otherwise noted.

The pixels in an XYZ image are coordinates in 3-D space of the corresponding pixel in the reference image. 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. The 3-D points can be referenced to any of the M2020 coordinate systems (specified by DERIVED_IMAGE_PARMS Group in the PDS label).

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. Realize this list of factors is only representative, and 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 IMAGE object. 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). Note that “0.0,0.0,0.0” is technically a legal value, but could occur at most once in an image and will rarely occur at all. The value is based on legacy software from previous missions and is compatible with current mission software. Additionally, it is extremely unlikely that the value will conflict with actual data since it is between the rover's wheels in Rover Frame (which cannot be imaged in stereo except for heroic SHERLOC-WATSON efforts); even when the Site origin is visible (e.g. the rover moves away), the possibility of sampling an exact value of “0.0,0.0,0.0” is considered extremely low. Also, if the value were to be sampled as actual data, losing a single pixel in the image is not problematic. The file format does not support nulls, so some other sentinel value would have to be chosen instead.

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. 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) product will generally be more accurate. XYE is provided as a convenience and its ellipsoid will always completely contain the RNE ellipsoid.

XYZ files can be filled (XYA). Individual X,Y,Z files can be filled as well. They can have associated rover mask files (MXY) as well as becoming masked (XYM) and saved as an overlay (XYO for XYZ, ZZO for Z-band), as shown in Figure 6-4 below. The rover mask files are discussed in more detail in Section 6.2.1.4.1.

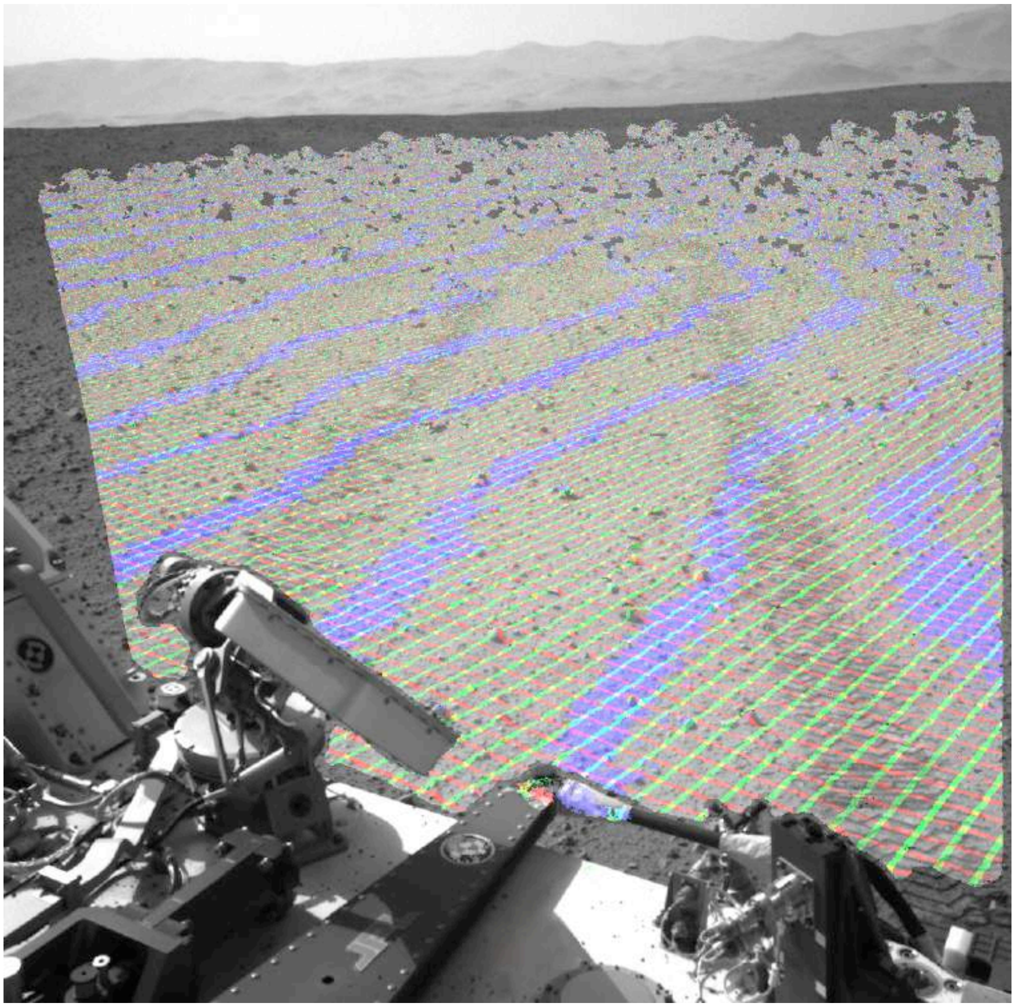


Figure 6-4 – XYZ data masked and overlaid onto image EDR

6.2.1.5.1 Range Filtering

TBD

6.2.1.5.2 Rover Mask RDR

The MXY (mask for XYZ) file is a special kind of mask file, called a rover mask. This file is intended to mask off the rover, so that XYZ points correlated on the rover do not show up in terrains. It is created using a low-fidelity volumetric model of the rover (with some margin around the rover components), which is articulated based on the telemetered joint positions of the arm, wheels, and suspension. This model is then projected into the image to create a mask. The articulation means the mask will reflect the arm and wheel positions, minimizing the amount of good terrain that is masked off. If joint positions are not available, a "swept volume" is used, meaning the mask covers all possible positions of the articulating device. The HGA is always modeled as a swept volume.

Rover masks can also be generated using predicted arm joint positions; this is the basis of the "SuperCam finder" mosaics produced during operations. Such predicted-state masks are marked using the special processing flag field in the filename.

Associated with the MXY mask image is an XML file (MXML extension) with the same base name. This file contains a polygonal representation of the articulated low-resolution volumetric model that is used to make the rover mask (before it is projected into the image). The format of this file is not described completely here, but it is straightforward XML, consisting of a large number of polygons in XYZ space.

6.2.1.6 “Range” RDR

A Range (distance) file contains 1 band of 32-bit floating point numbers.

The pixels in a Range image represent Cartesian distances from a reference point (defined by the RANGE_ORIGIN_VECTOR keyword in the PDS 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. A Range image is derived from an XYZ image and shares the same pixel geometry and XYZ coordinate system. As with XYZ images, range images can contain holes, defined by MISSING_CONSTANT. For M2020, this value is 0.0.

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 (RNA) and overlaid (RNO), as shown in Figure 5.2.1.5 below.

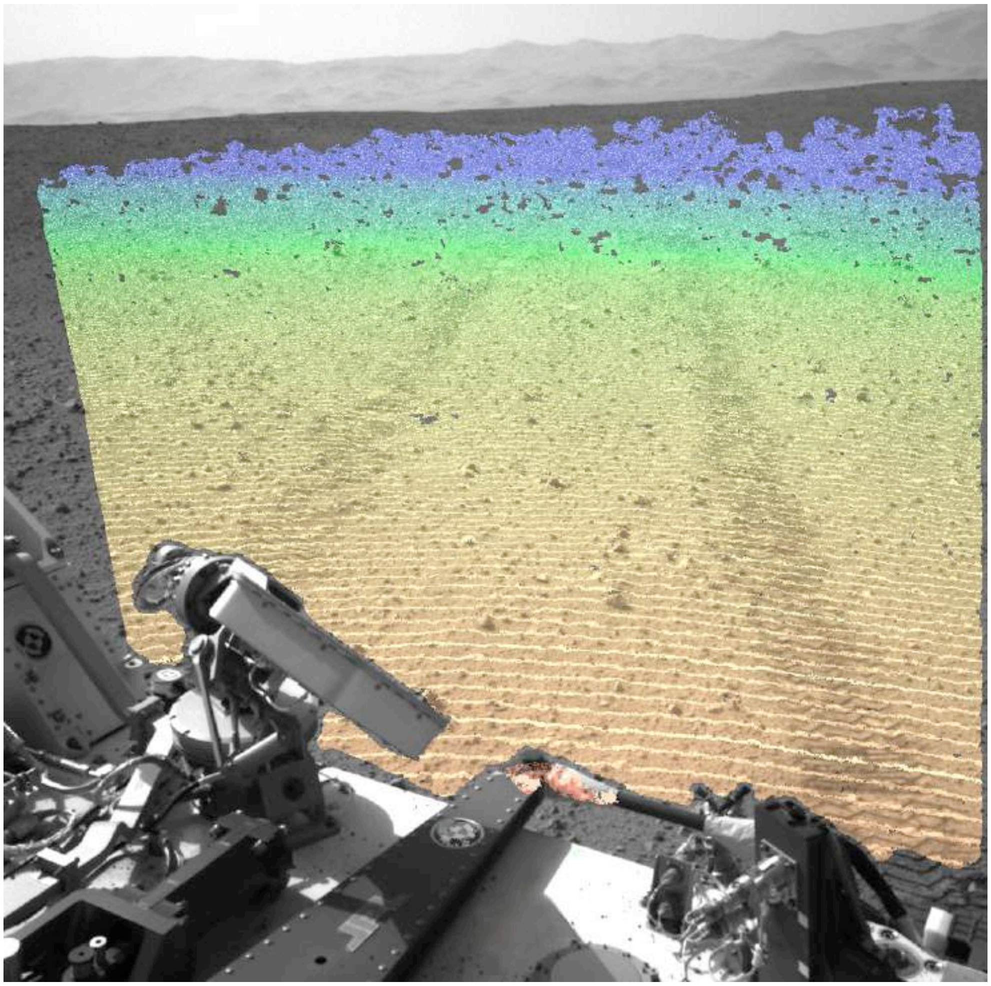


Figure 6-5 – Range (distance) data overlaid onto image EDR

6.2.1.7 "Surface Normal" (UVW) RDR

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 as a U Component RDR, V Component RDR and W Component RDR, respectively. The single component RDRs are implicitly the same as the UVW file, which is described below.

The pixels in a 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). The unit vector can be referenced to any of the M2020 coordinate systems (specified by the DERIVED_IMAGE_PARAMS Group in the PDS label).

Most UVW images will contain "holes", or pixels for which no UVW value exists. These are caused by many factors such as differences in overlap, correlation failures, and insufficient neighbors to compute a surface normal. 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).

Two special kinds of surface normal products are defined based on MER experience. These are not expected to be used during nominal M2020 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.

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 rover), which is appropriate for driving slope determination. Specific patch sizes are operational tuning parameters which have not yet been set, but will be provided in the history portion of the embedded VICAR label.

Surface normals can be filled (UVA) and overlaid (UVO). Separate U,V,W files can also be filled.

6.2.1.8 “Surface Roughness” RDR

The roughness maps, RUD (for the Drill) and RUT (for the Dust Removal Tool (BITCAR)) contain surface roughness estimates at each pixel in the image, along with a "goodness" flag indicating whether the roughness meets certain criteria.

For each pixel, the surface normal defines a reference plane. XYZ pixels in the area of interest are gathered, and the distance to the plane is computed. Outliers are thrown out. For the remainders, the minimum and maximum distances from the plane are found. Roughness is defined as the distance between this min and max (thus, is peak-to-peak variation within the area along the normal vector).

Two roughnesses are potentially computed. The first is an overall measurement containing all points within a radius of the central pixel. This is used for the BITCAR, and for the drill body. The second, used only for the drill, contains points within a ring between two radii. This is used for the drill stabilizer bars, and is not used for the BITCAR.

In each case, the computed roughnesses are compared to thresholds, which determine whether the point is "good" or not. The potential values of this goodness state are:

0.0 = No solution
1.0 = Both ring and overall roughnesses exceed thresholds
2.0 = Overall roughness (only) exceeds its threshold
3.0 = Ring roughness (only) exceeds its threshold
4.0 = Roughnesses within threshold (i.e. point is good)

Values of 1.0 and 3.0 appear only in the RUD product, since there is no ring for RUT.

The files are thus 2-band (RUT) or 3-band (RUD) float images, with the first band being the state, the second band being the overall roughness, and the third band being the ring roughness. For the second and third bands, 0.0 does not indicate a missing value (unlike most other products). Rather, 1.0 is used, as specified in MISSING_CONSTANT.

The default parameters and thresholds for these products are shown in Table 6-1 (all values in meters):

Table 6-1 – Default surface roughness parameters and threshholds

Param / Threshold	Product Type	
	RUD	RUT
Overall radius	0.075 m	0.035 m
Overall threshold	0.04 m	0.005 m
Ring inner radius	0.06 m	n/a
Ring outer radius	0.075 m	n/a
Ring threshold	0.015 m	n/a

6.2.1.9 Slope RDRs

The Slope-related RDRs represent aspects of the slope of the terrain as determined by stereo imaging. The Slope Map is derived from the UVS product, which contains the rover-sized surface normal at every point. 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.

6.2.1.9.1 "Slope" (nominal) RDR

The SLP (SLO) type contains the slope in degrees for each pixel.

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

6.2.1.9.2 "Slope Rover Direction" RDR

The SRD (SRO) type contains the component of the slope (in degrees) that was facing the rover, i.e. if the rover went radially outward from its current position, this indicates the climb or descent. In the formula below, R is the rover's position.

$$\mathbf{V} = \frac{\begin{bmatrix} x - \mathbf{R}_x & y - \mathbf{R}_y \end{bmatrix}}{\sqrt{(x - \mathbf{R}_x)^2 + (y - \mathbf{R}_y)^2}}$$

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

6.2.1.9.3 "Slope Heading" RDR

The SHD (SHO) 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.

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

6.2.1.9.4 "Slope Magnitude" RDR

The SMG (SMO) type contains the magnitude of the normal unit vector projected onto the horizontal plane. It is directly related to sin(slope).

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

6.2.1.9.5 "Slope Northerly Tilt" RDR

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

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

6.2.1.9.6 "Solar Energy" RDR

The SEP (SEO) type is included as a placeholder. It is not used for M2020.

6.2.1.10 Arm Reachability RDRs

The Arm Reachability maps contain information about whether or not the instruments on the Arm can "reach" (contact or image) the object or location represented by each pixel in the scene, and how hard they can push ("preload"). They are derived from the XYZ and Surface Normal (UVW) products.

The geometry of the reachability maps match the reference XYZ, and Surface Normal (UVW) images, in that each pixel in the file directly corresponds to the pixel at the same location in the other products.

For the arm reachability map products, pixels with a DN of 0 denote an area where an arm instrument is unable to make contact (see Tables 6-2 and 6-3). To avoid ambiguity in the operational data store and archive, all arm reachability products are generated and archived, regardless of whether the reachability values are 0 or not. Consequently, there are arm reachability products comprised entirely of pixels with DN 0, that is, with no pixels indicating reachability. These products help document the decision process employed by the M2020 project when identifying contact science targets.

6.2.1.10.1 "Arm Reachability" RDR

The reachability map (ARM) encodes information for each of the 5 arm instruments in each of the 8 possible arm configurations, for a total 40 values per pixel. It is stored as a 5-band image of 16-bit integers in standard Band Sequential order. Each band represents one of the 5 arm instruments in the order defined by INSTRUMENT_BAND_ID. Within each band, the 16-bit integer contains 2 bits of information for each of the 8 configurations (in the order defined by CONFIGURATION_BIT_ID) packed into the 16-bit value. The first mode in CONFIGURATION_BIT_ID is in the high-order 2 bits of the integer. The two bits represent 4 states: not reachable (0), and three levels of reachability, with 3 being the most easily reachable. Reachability is determined by checking if the contact, approach, and overdrive positions are reachable using the computed surface normal and 3 additional normals differing by a "tweak" angle from the computed one. Reachability level 3 indicates all checks pass, while 1 and 2 indicate that some did not and may require extra attention from the arm operator in order for the instrument to safely reach the point.

Table 6-2 – Arm reachability bit assignements per configuration (TBD)

16-bit Integer Bit Order (15=MSB, 0=LSB)							
15 - 14	**13 - 12**	**11 - 10**	**9 - 8**	**7 - 6**	**5 - 4**	**3 - 2**	**1 - 0**
Shoulder Out, Elbow Up, Wrist Up	Shoulder Out, Elbow Up, Wrist Down	Shoulder Out, Elbow Down, Wrist Up	Shoulder Out, Elbow Down, Wrist Down	Shoulder In, Elbow Up, Wrist Up	Shoulder In, Elbow Up, Wrist Down	Shoulder In, Elbow Down, Wrist Up	Shoulder In, Elbow Down, Wrist Down

Table 6-3 – Arm reachability values

Value		Description
Decimal	Binary	
0	00	Not reachable
1	01	Reachability quality 1
2	10	Reachability quality 2
3	11	Reachability quality 3 (best)

Note that reachability maps may include masks (MAR), be masked (ARK), and be overlaid (ARO) as shown in Figure 6-6 below.

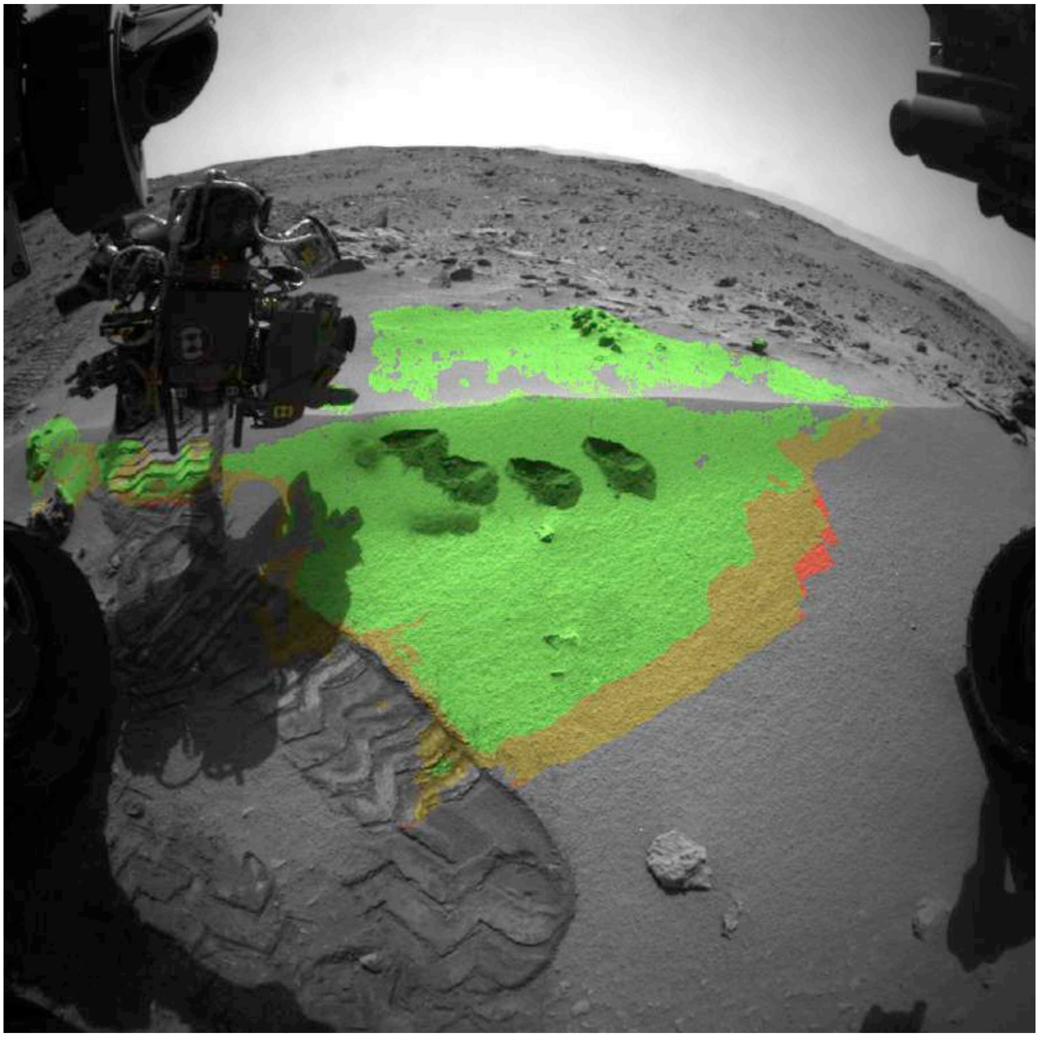


Figure 6-6 – Arm reachablilty qualities overlaid onto image EDR

6.2.1.10.2 "Arm Preload" RDR

The ARP type indicates the minimum and maximum preload values (in Newtons) that can be applied by the Drill instrument at the point represented by the pixel. This is a 2-band 16-bit signed integer products, where the bands represend (minimum, maximum) preloads (also defined by INSTRUMENT_BAND_ID).

6.2.1.11 Color RDRs TBD

TBD: Rework, or fold into the color processing section above

The ECAM, SuperCam RMI, ZCAM, SHERLOC, and EDL cameras contain a Bayer pattern of color filters on the detector. This is a repeating pattern of 4 pixels where each "cell" contains one red, two green, and one blue pixel (see Figure 4.3.3.1). This allows acquisition of color without using the filter wheel. In order to be generally useful, the image must normally be "de-Bayered" in one of several ways. The method of de-Bayering is indicated by the "config" field in the product filename, with additional support from the "samp" field. Any time de-Bayering results in an image size change, the camera model must be adjusted to match.

For color images, the Bayer cells are typically extracted to separate color bands. This is indicated by values of "R", "G", or "B" for the "color" field. If the image is half-size in both dimensions as compared to the original (e.g. the RGB cells are simply extracted), then the "samp" field displays "TBD" to indicate Bayer subsampling. If the pixels are interpolated back to full size, the "samp" field is unchanged. Green pixels, being twice as numerous, present special challenges. For a full-size image, the "color" code is "TBD". For Bayer-subsampled images, several cases are possible. If only the upper or lower green pixels are used from each cell, then the "color" values are "A" or "D". If both are combined, the value is "TBD". If both are present, so the image is twice as big in one dimension than the other, the value is "D". Regardless of the extraction mode, if all three colors are merged back into a single 3-band image, the value is "F".

For non-color images taken using geology filters, there are two additional options. Individual cells can be extracted as described above. They can also be averaged in a 2x2 pattern (potentially taking into account responsivity of the filter) to create one pixel per Bayer cell. This is indicated with a "color" value of "TBD". Or, each pixel can be corrected for the combined responsivity of the Bayer and geology filters; this is indicated with a value of "TBD".

Note that some de-Bayering is done onboard. This is reflected using the "color" code. 2x2 averaging or subsampling simply results in a downsampled or subsampled image, while JPEG creation results in a single 3-band EDR image. Thus for all EDRs, the "color" field represents the Bayer state.

6.2.1.12 "Photometry" RDR

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 (IEA).

6.2.1.13 Tactical Terrain Map RDRs TBD

TBD: Rework to talk about OBJ files and dynamically-adjusted triangle sizes and …

Terrain models are high level products which are derived using XYZ files and the corresponding image files. The XYZ files contain point clouds: sets of vertices in a specific coordinate system. The corresponding image files are used to obtain intensity or color information for each vertex in the point cloud. The terrain models are generated by triangulating point clouds using volume based surface extraction. The original image is used as a texture map to add detail and color to the polygonal surface representation. Terrain models are stored in Open Inventor binary format. Image textures are stored in SGI RGB format. Height maps (i.e., digital elevation maps, or DEMs) are also produced and used by the Rover Sequencing and Visualization Program (RSVP) for tasks which require simple and fast lookup such as rover settling. Height Maps are stored in VICAR format.

6.2.1.13.1 "Per-XYZ" Terrain RDRs TBD

For every XYZ RDR created, the following terrain products are generated and follow the Single-frame RDR filename convention (see Section 6.1.1):

a) (*.tar) - 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 defines terrain in multiple Level of Details (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.

b) (*.iv) - Open Inventor terrain representation of a point cloud defined in the XYZ RDR. It's a single file generated by combining all tiles contained in the tar-file described above and storing it as an Open Inventor binary file. It constitutes the per-XYZ mesh product.

c) (*.rgb) - Image file in SGI RGB format that is used as the Texture Map for the per-XYZ mesh product.

d) (*.ht) - Height Map (DEM) derived from the XYZ RDR, stored as an image file. It's in VICAR format but is not fully compliant to VICAR label specifications. It has 3 bands. Band 1 contains actual height data. Band 2 fills areas for which there is no actual data using interpolation. Band 3 provides metric of how close a pixel value is to the actual data.

Example:

Given the point cloud XYZ RDR named NLA_412403715XYZLF0060000NCAM15000M1.IMG and the corresponding image NLA_412403715RASLF0060000NCAM15000M1.IMG, the following files are created:

- NLA_412403715RASLF0060000NCAM15000M1.tar - Collection of vertices tiles in Open Inventor ASCII format representing spatial subdivision of the point cloud.
- NLA_412403715RASLF0060000NCAM15000M1.iv - Concatenation of all tiles into one Open Inventor file in binary format.
- NLA_412403715RASLF0060000NCAM15000M1.rgb - Texture Map image in SGI image format.
- NLA_412403715RASLF0060000NCAM15000M1.ht - Height Map image, with the following label items defining the spatial extent of the Height Map:

NL = 512	(number of lines)
NS = 442	(number of samples)
X_AXIS_MINIMUM = -39.906654	
Y_AXIS_MINIMUM = -8.1579	
MAP_SCALE = 0.087151	(Resolution at which Height Map has been generated)

6.2.1.13.2 "Unified" Terrain RDRs TBD

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 rover planners during tactical operations. Per-XYZ polygonal surfaces are generated using XYZ RDRs defined in a Site frame. The tool RSVP extracts Site information from the unified mesh product filename to render terrains into the proper locations for rover traverse and arm placement applications, as shown in Figures 6-7 and 6-8.

The terrain products listed below follow the unified mesh product filename convention (see Section 6.1.3), which differs from the Single-frame filenames carried by the per-XYZ terrain products:

a) (*.iv) - Open Inventor file in ASCII format that contains references to all individual binary per-XYZ Open Inventor files

b) (*.mod) - ASCII file that contains references to all corresponding individual per-XYZ ".ht" Height Maps files.

Example:

Assuming the generation of multiple per-XYZ mesh products described and exemplified in Section
6.2.1.12.1, the following unified mesh files are created:

- N_L0168_RASLF_006_0000_AUTOGENM1.iv - Collection of references to Open Inventor files including NLA_412403715RASLF0060000NCAM15000M1.iv

- N_L0168_RASLF_006_0000_AUTOGENM1.mod - Collection of references to Height Maps including N_L0168_RASLF_006_0000_AUTOGENM1.ht

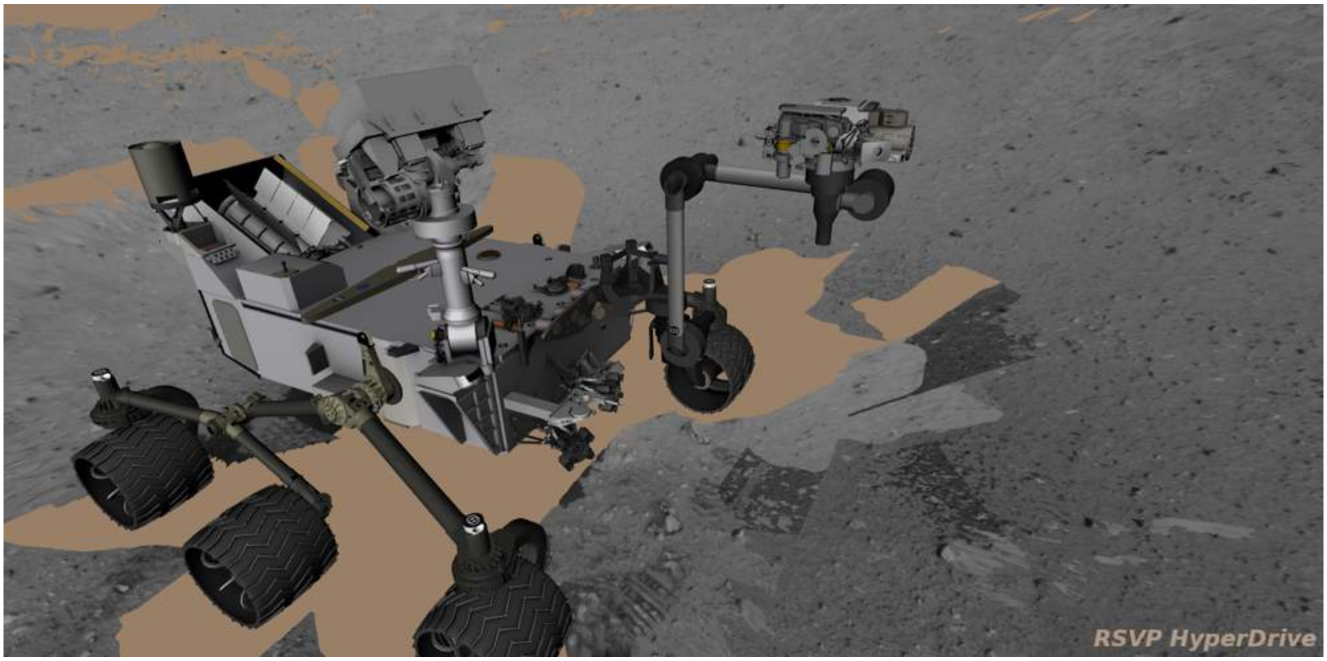


Figure 6-7 – Rover location rendered in a unified terrain mesh

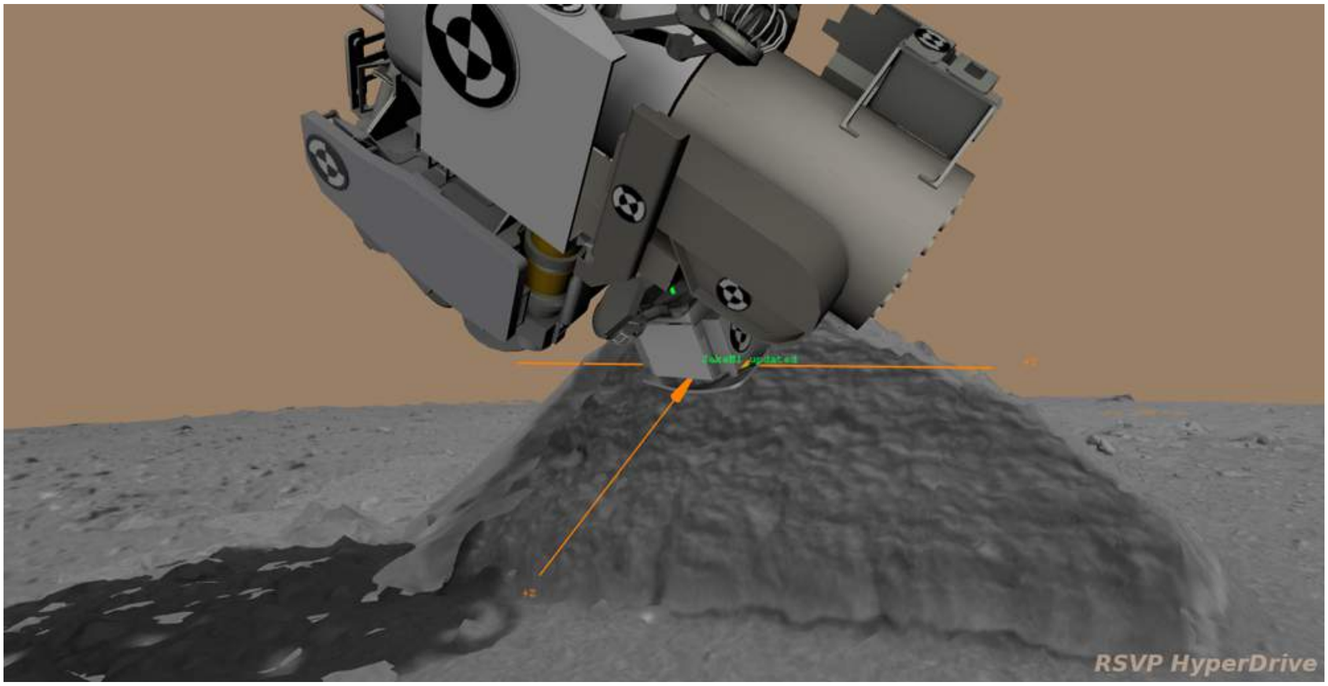


Figure 6-8 – Arm payload location rendered in a unified terrain mesh

6.2.1.14 Mosaic RDRs

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 under IDS. It should be noted that governing methods and software can differ between IDS and other operations subsystems or science instrument teams; the algorithms followed by other teams may not be the same as described here.

6.2.1.14.1 Overview of Mosaics in General

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

Mosaics can be assembled autonomously by tracing a view ray from each mosaic location or pixel into the scene, determining its intersection with a surface model (typically a ground plane), and then querying each input image to determine if that point lies within its field of view. In this fashion mosaics containing perhaps hundreds of images can be assembled for each spectral band. It may be necessary to refine the camera pointing in order to produce accurate mosaics. This requires the determination of the actual azimuth and elevation of each image in order to correct for errors such as gear backlash. One way to do this is to acquire tiepoints between all pairs of overlapping images. Camera pointing parameters are then estimated which cause the camera model to map the tiepoints to their correct locations. In many cases this can be accomplished automatically, but it often requires human intervention to select tiepoints because of small overlap, parallax, or changing lighting.

6.2.1.14.2 Mosaicking Method

The process used by the MIPL software to create mosaics is described below. It consists of several sub-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. ***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.

   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 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.

   Regardless of method, the result is encapsulated in a pointing correction or "nav" file. A more detailed description of this file is provided in Section 5.2.1.13.3, but fundamentally, this file contains, for each image being corrected, the original pointing parameters and the revised pointing parameters.

Pointing parameters are simply those numbers which represent how the camera is pointing in the rover frame, reduced to available degrees of freedom. These are used as inputs to the kinematics procedures which 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. The mast-mounted cameras have a standard model with two parameters: azimuth and elevation actuator angles. However, another model is available with three: azimuth, elevation, and “twist”, which is a rotation around the camera’s A axis (A being one of the CAHV camera model parameters). For SHERLOC-WATSON, 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). The Hazcam and MARDI models have zero parameters, since they are rigidly attached to the rover body. Other pointing models may exist 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.

B. ***Output Projection Determination*** - The output projection is then determined. The parameters describing the projection are listed in Appendix A, and described in detail in Appendix F. 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.

C. ***Surface Determination*** - A surface model is critical for mosaics. 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 F 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.

Note that the parameters (surface normal and ground point, for PLANE) for an appropriate surface model are often determined automatically as part of the pointing correction process, and the surface model can be stored in the “nav” file.

D. ***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 PDS 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.

E. ***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 rover situations, SPHERE1 thus roughly models a convex hill, while SPHERE2 roughly models a concave crater when the rover is outside the sphere.

F. ***Projection from Surface to Input*** - The XYZ location (or direction for the infinity case) is then back-projected into each input camera model 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.

G. ***Interpolation and Storage of the Result*** - Finally, a bilinear interpolation is performed on the input image, 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.

6.2.1.14.3 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. On PDS-released archive volumes, the ancillary files have the same basename as the mosaic to preserve one-to-one matching with the mosaic, albeit with a different extension. On the operational data store (ODS) for operations, they might not necessarily have the same mosaic basename, 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.

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 (ODS), 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: specifed 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 indentifying information. Note that the "static_parameters" file is listed as "M2020:M2020_pma.point" in some nav files; this should be corrected to "M2020:M2020_mast.point" and refers to the calibration file used to find mast kinematics parameters.

   Pointing correction works by applying a set of pointing parameters (e.g. mast azimuth and elevation) to a kinematics algorithm and using that to re-point the camera model. See definition of keyword POINTING_MODEL_NAME in Appendix F 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 (irregardless 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.

   At the end is usually a "surface_model" element describing the surface model determined by the MIPL software program MARSNAV (which creates the nav files). This information is repeated in the SURFACE_MODEL_PARMS group of the mosaic product's label.

c) **Tiepoint files** - With a ".TIE" or ".TPT" extension, tiepoint files are XML files containing image tiepoints used as input to program MARSNAV. 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 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 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. Other types are rarely used and their descriptions are beyond the scope of this document. Full descriptions are in the help documentation for 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 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). 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.

6.2.1.14.4 "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:

azimuth = i / MAP_RESOLUTION + START_AZIMUTH
elevation = (ZERO_ELEVATION_LINE - j) / MAP_RESOLUTION

The view ray emanates from the point PROJECTION_ORIGIN_VECTOR.

Figure 6-9 shows such a mosaic overlaid onto azimuth and elevation grid lines, with individual frame boundaries superimposed and annotated by number. 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 north.

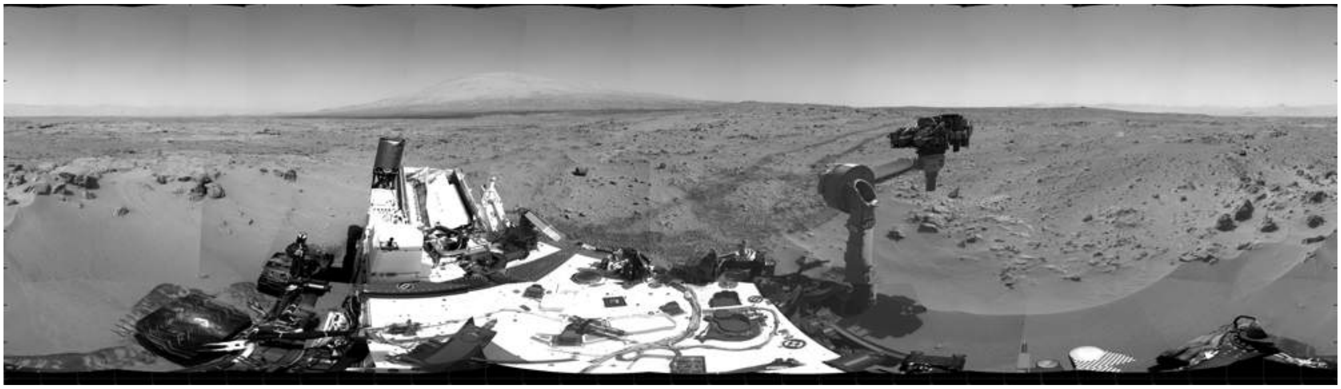


Figure 6-9 – Cylindrical projection mosaic (CYL)

6.2.1.14.5 “Camera Point Perspective Mosaic” RDR

Figure 6-10 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 M2020, this type of mosaic is typically in the Rover Frame and thus may have a tilted horizon if the rover 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 6.4.1 and references [Ref 18] through [Ref 27] for the mathematics.

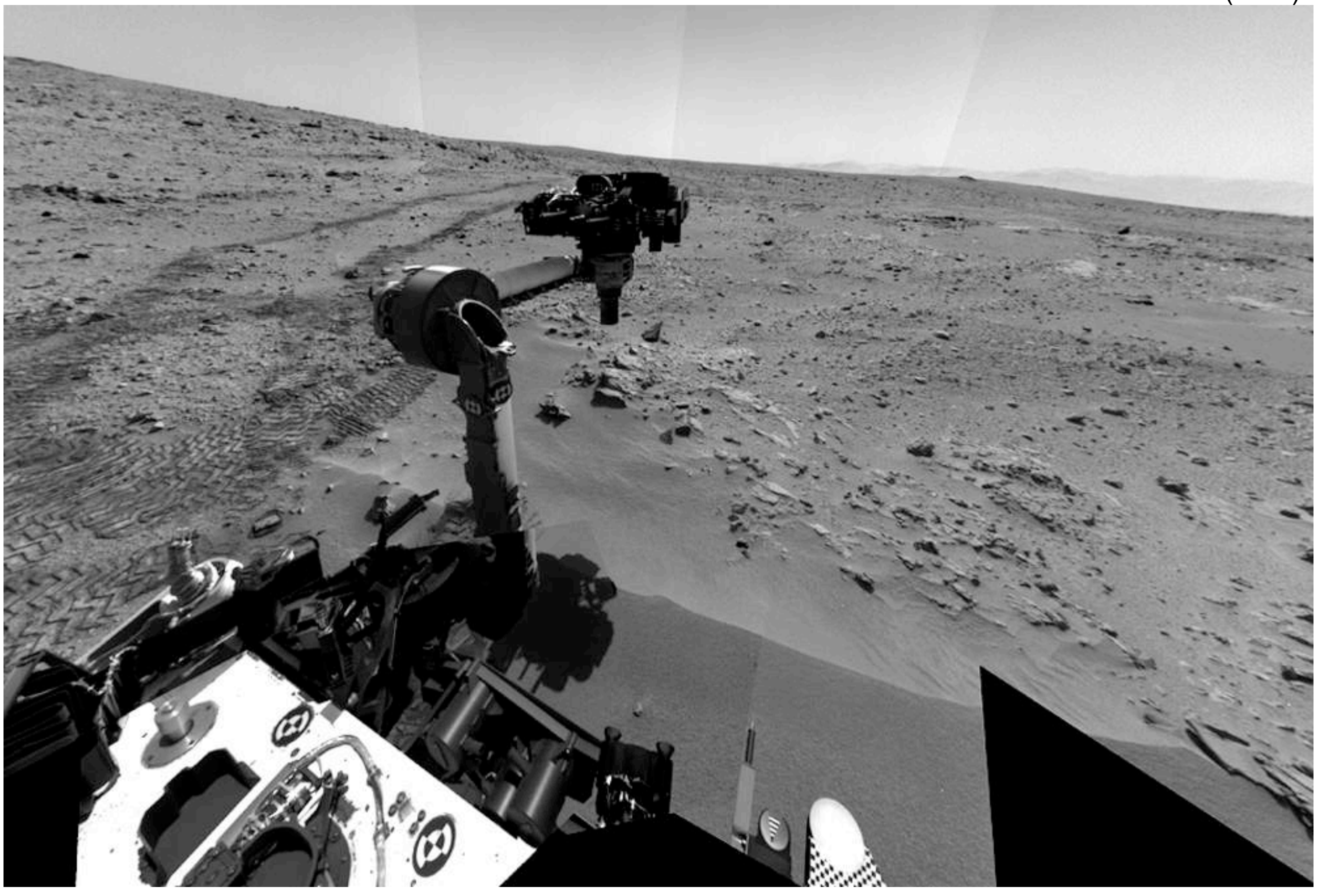


Figure 6-10 – Camera perspective mosaic (PER)

6.2.1.14.6 "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:

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

   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):

ifov = (STOP_AZIMUTH - START_AZIMUTH) / LINE_SAMPLES

3) The azimuth of the column is computed:

   azimuth = START_AZIMUTH + i * 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. 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 Rover 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 rover 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 Rover Nav 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):

$x\_center = \vec{A} \cdot \vec{H}$

$y\_center = \vec{A} \cdot \vec{V}$

samp = x_center

line = y_center + j - PROJECTION_ELEVATION_LINE

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 5.4.1 and references [Ref 18] through [Ref 27] for the mathematics of camera models.

Figure 6-11 shows a Cylindrical-Perspective projection in which a 360 degree view can be viewed in stereo. This is a perspective projection similar to Figure 5.2.1.13.5 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 rover 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 rover 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.

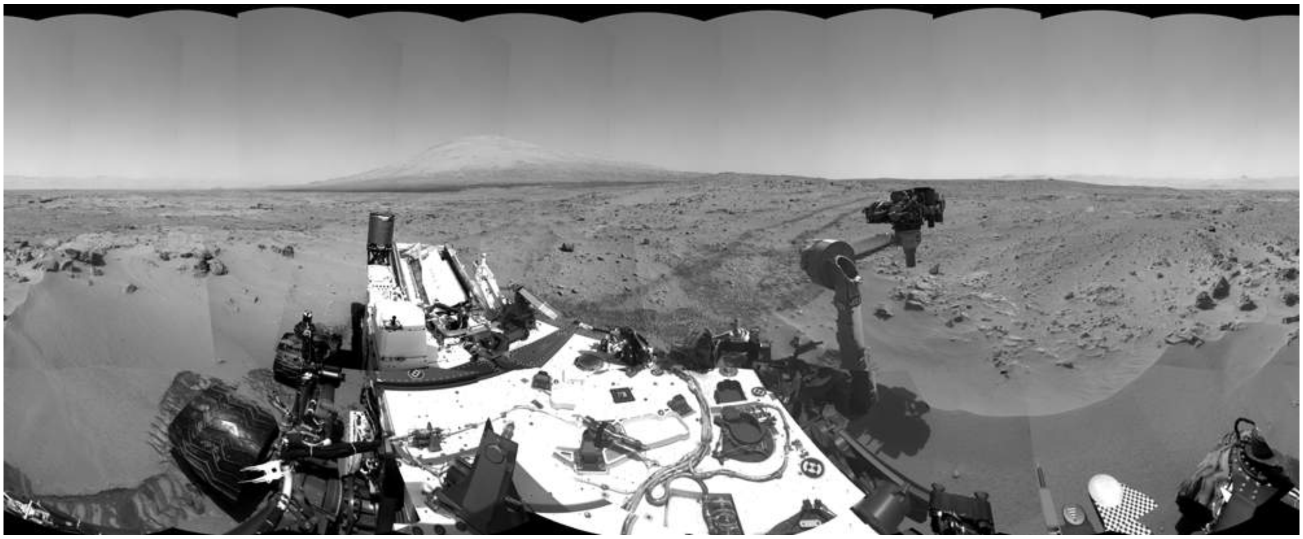


Figure 6-11 – Cylindrical-perspetive projection mosaic (CYP)

6.2.1.14.7 “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 6-12 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.

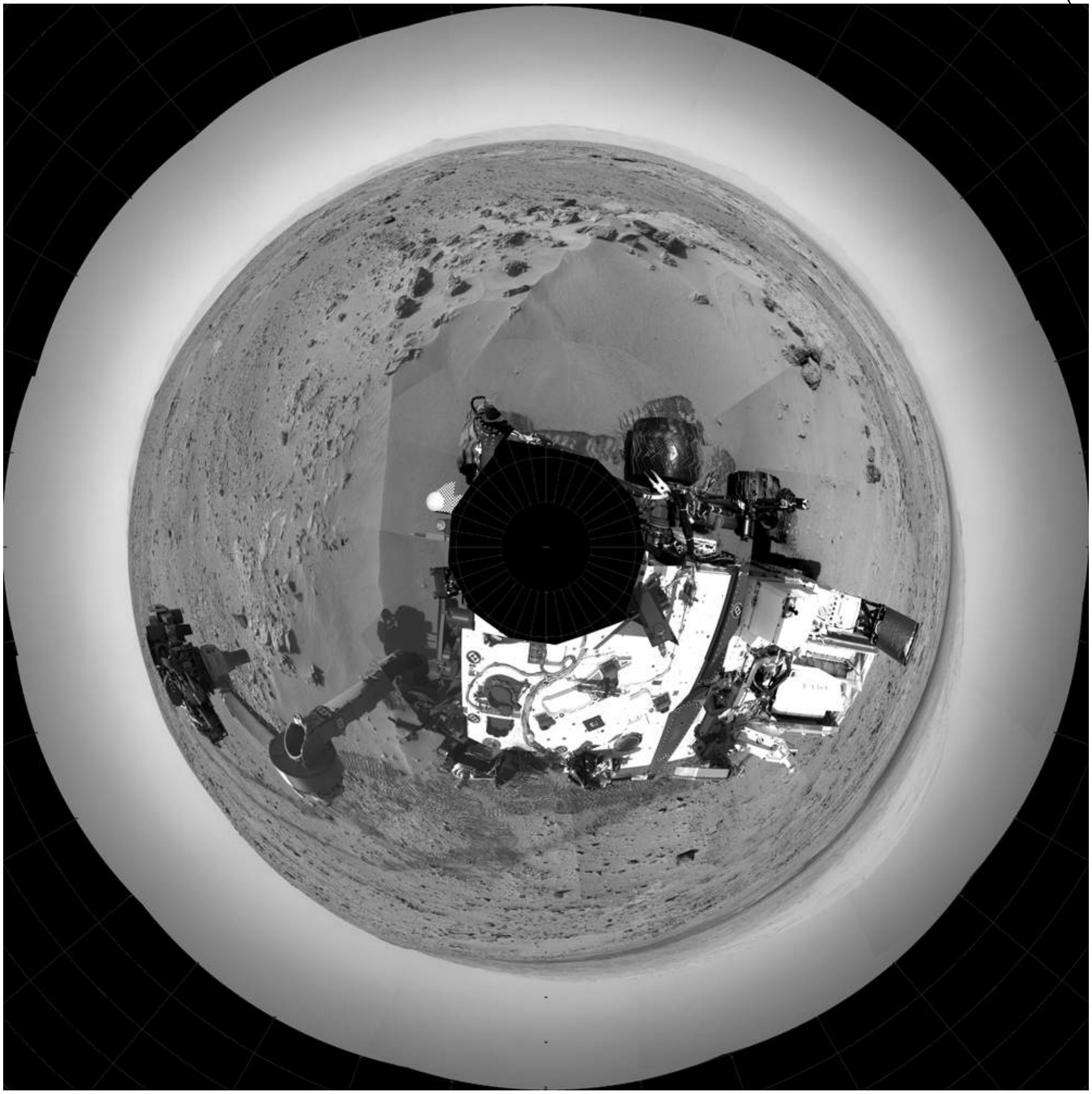


Figure 6-12 – Polar projection mosaic (POL)

6.2.1.14.8 "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 imagery, 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.

MIPL creates the Vertical projection as follows:

nl = number of lines in the mosaic (IMAGE object, LINES)
ns = number of samples in the mosaic (IMAGE object, 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 E of Section 5.3.13.3 above, it is changed to point straight up (0,0,-1).

Figure 6-13 shows 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.

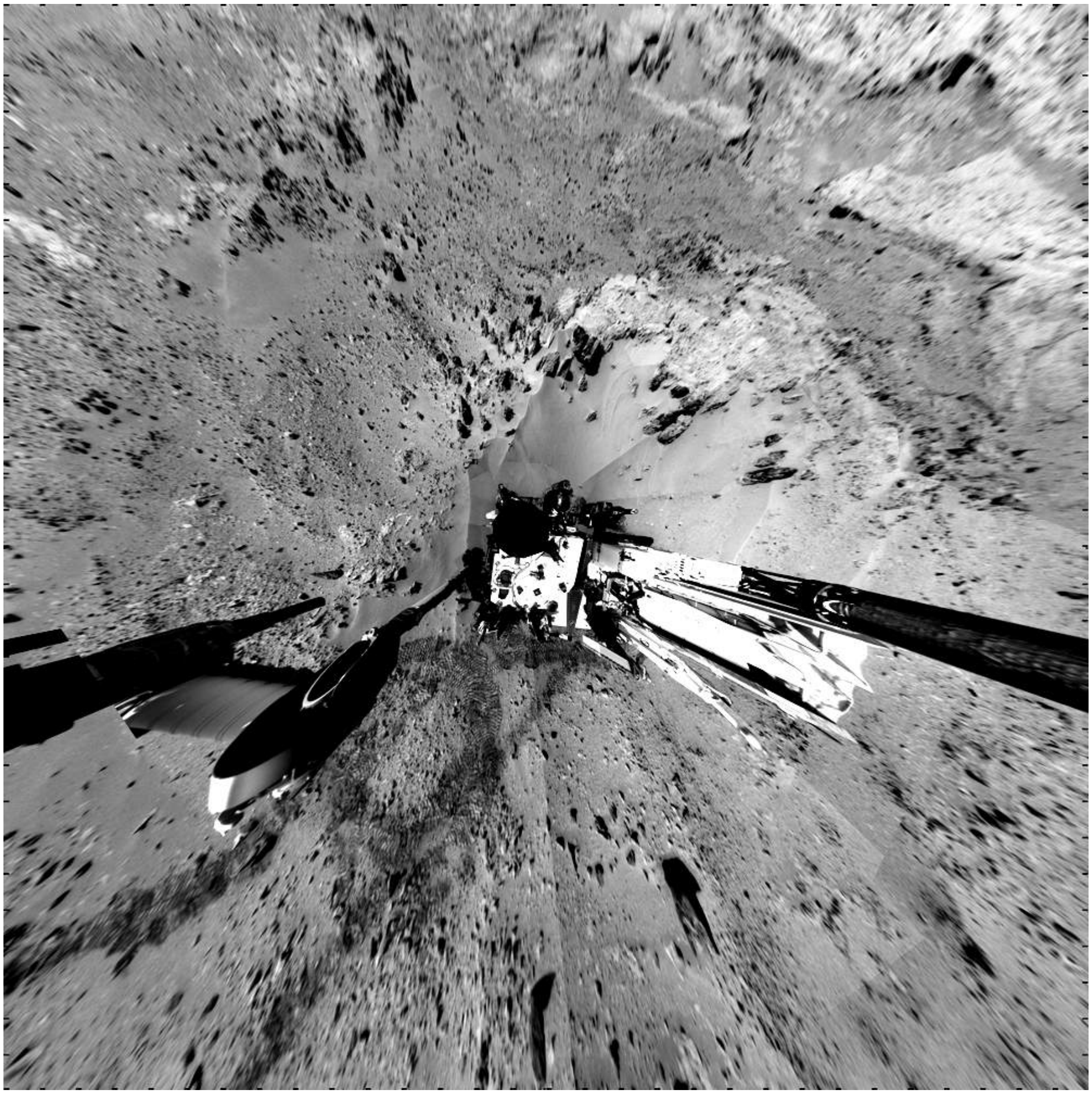


Figure 6-13 – Vertical projection mosaic (VRT)

6.2.1.14.9 "Orthographic Projection Mosaic" RDR

The Orthographic projection is a generalization of the Vertical projection intended primarily for use with SHERLOC-WATSON data. 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 Xhat and Yhat 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:

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

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

where the "•" 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 PROJECTION_X_AXIS_VECTOR = (1,0,0), PROJECTION_Y_AXIS_VECTOR = (0,1,0), and PROJECTION_Z_AXIS_VECTOR = (0,0,1).

6.2.1.14.10 "Orthorectified Projection Mosaic" RDR

Orthorectified mosaics 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 imagery. 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, PROJECTION_X_AXIS_VECTOR = (1,0,0), PROJECTION_Y_AXIS_VECTOR = (0,1,0), and PROJECTION_Z_AXIS_VECTOR = (0,0,1).

The specific algorithm must deal with filling holes in the output mosaic and is still to be determined, as the software for this projection remains under development as of this writing.

Figure 6-14 shows an orthorectified rendering.

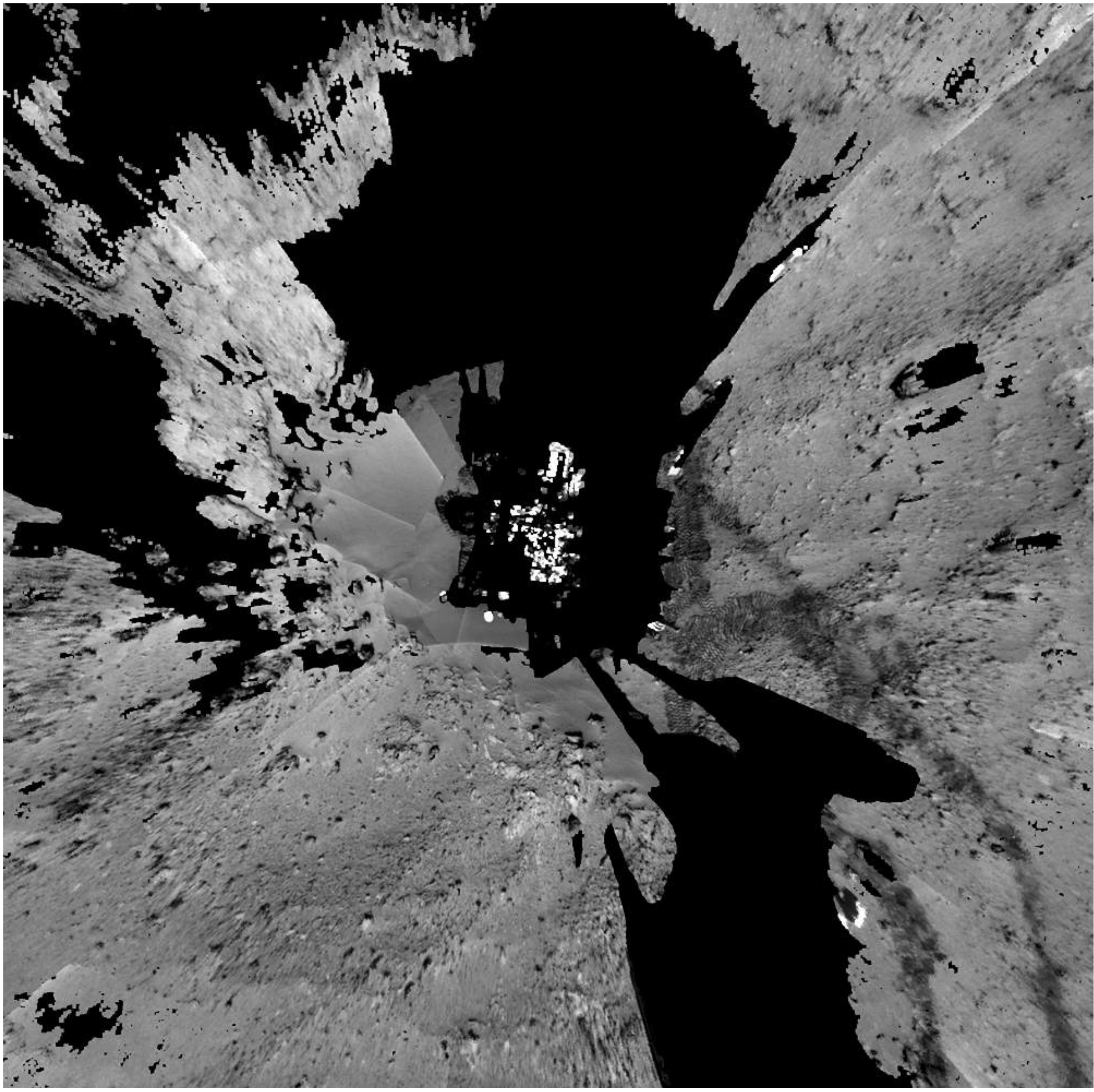


Figure 6-14 – Orthorectified projection mosaic (ORR)

6.2.1.14.11 Non-image Mosaics

Normally mosaics are created using imagery, 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, surface normal, and the various Slope types.

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.

As another example, Slope mosaics are often created and then overlaid on an image mosaic using the same projection parameters to help with rover navigation.

For M2020, a common product is the "SuperCam Finder" mosaic. This consists of a set of three Polar projection mosaics: the imagery, the range, and the rover exclusion mask. The mask is adjusted based on actual or predicted arm locations. The overlay of the mask mosaic over the image aids the SuperCam team in targeting (to avoid the rover and the arm), while the overlay of range helps determine focus distances for commanding.

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.

TBD: Talk about mosaic ICM/IDX files

6.2.1.15 “Anaglyph” RDR

A stereo anaglyph is a method of displaying stereo imagery quickly and conveniently using conventional display technology (no special hardware) and red/blue glasses. This is done by displaying the left eye of the stereo pair in the red channel, and displaying 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).

The Anaglyph method can also apply to multi-frame mosaic products. MIPL-generated mosaic Anaglyphs occasionally required some subtle pixel-shifting of the right eye mosaic data to improve the stereo effects. Mosaic Anaglyph products are distinguishable in the Mosaic RDR filename convention (see Section 6.1.2).

6.3 RDR Product Format

6.3.1 Image RDRs

The image RDR data products covered by this SIS are listed in Tables 5.3.1.1 and 5.3.1.2 below. Products listed as “1-3” bands are generally 1-band products but could have 3 RGB bands for certain ZCAM camera modes.

Table 6-4 – 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 pipline as a special request	🟥

Table 6-5 – M2020 camera image EDR/RDR product binary formats

Data Product	Type	PDS Proc. Level	Derived Image Type
Original Image Product, Possibly Companded (8->12bit)	**ECM**	Raw	N/A
Original JPEG as received from the rover	**EJP**	Raw	N/A
Single frame of a Video Sequence (possibly companded)	**ECV**	Raw	N/A
Z-stack frame (possibly companded)	**ECZ**	Raw	N/A
Depth Map	**EDM**	Raw	N/A
Recovered Product (possibly companded)	**ECR**	Raw	N/A
Reference Pixel	**ERP**	Raw	N/A
Decompanded Image Product	**EDR**	Partial	IMAGE
Single frame of a Video Sequence (de-companded)	**EVD**	Partial	IMAGE
Z-stack frame (de-companded)	**EZS**	Partial	IMAGE
Recovered Product (de-companded)	**ERD**	Partial	IMAGE
De-Bayered (de-mosaicked)	**EBY**	Partial	IMAGE
CAHV-linearized	**LIN**	Partial	IMAGE
Bayer pattern	**BAY**	Partial	IMAGE
Base image from which all RDRs are derived	**FDR**	Partial	IMAGE
Image Mask File	**MSK**	Derived	MASK
Rad-corrected IOF radiance factor, float	**IOF**	Calibrated	IMAGE
Rad-corrected IOF radiance factor, integer	**IOI**	Calibrated	IMAGE
Rad-corrected absolute radiance units, float	**RAF**	Calibrated	IMAGE
Rad-corrected absolute radiance units, 15-bit integer static scale	**RAD**	Calibrated	IMAGE
Rad-corrected absolute radiance units, 12-bit integer static scale	**RAS**	Calibrated	IMAGE
Rad-corrected absolute radiance units, 15-bit int dynamic scale	**RAY**	Calibrated	IMAGE
Rad-corrected radiance (RAD), masked	**RDM**	Calibrated	IMAGE
Rad-corrected for Instrument Effects only, integer DN	**RIE**	Calibrated	IMAGE
Rad-corrected for Instrument Effects only, float	**RIF**	Calibrated	IMAGE
Rad-corrected scaled radiance (RAS), masked	**RSM**	Calibrated	IMAGE
Zenith-scaled radiance, float	**RZF**	Calibrated	IMAGE
Zenith-scaled radiance, 15-bit integer static scale	**RZD**	Calibrated	IMAGE
Zenith-scaled radiance, 12-bit integer static scale	**RZS**	Calibrated	IMAGE
Zenith-scaled radiance, 15-bit integer dynamic scale	**RZY**	Calibrated	IMAGE
Stereo Delta Disparity (2-band, true disparity offset)	**DDD**	Derived	DISP_DELTA_MAP
Stereo Delta Disparity Line (single-band)	**DDL**	Derived	DISP_DELTA_LINE_MAP
Stereo Delta Disparity Sample (single-band)	**DDS**	Derived	DISP_DELTA_SAMPLE_MAP
Stereo Disparity Error Metric	**DSE**	Derived	DISPARITY_ERROR_MAP
Stereo Disparity of Lines (single-band)	**DSL**	Derived	DISPARITY_LINE_MAP
Stereo Disparity Final	**DSP**	Derived	DISPARITY_MAP
Stereo Disparity Raw	**DSR**	Derived	DISPARITY_MAP
Stereo Disparity of Samples (single-band)	**DSS**	Derived	DISPARITY_SAMPLE_MAP

Data Product	Type	PDS Proc. Level	Derived Image Type
Stereo Disparity Mask File	**MDS**	Derived	MASK
Surface Roughness (general, not instrument-specific)	**RUF**	Derived	ROUGHNESS_MAP
Stereo First-stage Disparity Final	**DFF**	Derived	DISPARITY_MAP
Stereo First-stage Disparity Line	**DFL**	Derived	DISPARITY_LINE_MAP
Stereo First-stage Disparity Sample	**DFS**	Derived	DISPARITY_SAMPLE_MAP
XYZ Mask File	**MXY**	Calibrated	MASK
XYZ Error Metric	**XYE**	Calibrated	XYZ_ERROR_MAP
XYZ expressed in Rover Nav frame	**XYR**	Calibrated	XYZ_MAP
XYZ expressed in Site frame	**XYZ**	Calibrated	XYZ_MAP
XYZ expressed in Rover, Masked	**XRM**	Calibrated	XYZ_MAP
XYZ expressed in Site frame, Masked	**XYM**	Calibrated	XYZ_MAP
XYZ Filled	**XYF**	Derived	XYZ_MAP
XYZ with Overlay	**XYO**	Derived	IMAGE
XYZ from disparity raw expressed in Site frame	**XRZ**	Derived	XYZ_MAP
XYZ from disparity raw expressed in Rover Nav frame	**XRR**	Derived	XYZ_MAP
Digital Elevation Model. Like ZZZ but positive up.	**DEM**	Calibrated	ELEVATION_MAP
XYZ X-band	**XXX**	Calibrated	X_MAP
XYZ Y-band	**YYY**	Calibrated	Y_MAP
XYZ Z-band	**ZZZ**	Calibrated	Z_MAP
XYZ X-band Filled	**XXF**	Derived	X_MAP
XYZ Y-band Filled	**YYF**	Derived	Y_MAP
XYZ Z-band Filled	**ZZF**	Derived	Z_MAP
Surface Normal (UVW) Filled	**UVF**	Derived	UVW_MAP
Surface Normal (UVW) with Overlay	**UVO**	Derived	IMAGE
Surface Normal (UVW) Projected onto Plane	**UVP**	Derived	UVW_MAP
Surface Normal (UVW) for Slope computations	**UVS**	Calibrated	UVW_MAP
Surface Normal (UVW) Angle (Theta) between Normal and Plane	**UVT**	Derived	ANGLE_MAP
Surface Normal (UVW)	**UVW**	Calibrated	UVW_MAP
Surface Normal (UVW) U-band	**UUU**	Calibrated	U_MAP
Surface Normal (UVW) V-band	**VVV**	Calibrated	V_MAP
Surface Normal (UVW) W-band	**WWW**	Calibrated	W_MAP
Surface Normal (UVW) U-band Filled	**UUF**	Derived	U_MAP
Surface Normal (UVW) V-band Filled	**VVF**	Derived	V_MAP
Surface Normal (UVW) W-band Filled	**WWF**	Derived	W_MAP
Range Error Metric	**RNE**	Calibrated	RANGE_ERROR_MAP
Range Filled	**RNF**	Derived	RANGE_MAP
Range from Camera	**RNG**	Calibrated	RANGE_MAP
Range from Camera, Masked	**RNM**	Calibrated	RANGE_MAP
Range with Overlay	**RNO**	Derived	IMAGE
Range from Rover Nav frame origin	**RNR**	Calibrated	RANGE_MAP
Arm Reachability, Masked	**ARK**	Derived	REACHABILITY_MAP
Arm Reachability	**ARM**	Derived	REACHABILITY_MAP
Arm Reachability with overlay	**ARO**	Derived	IMAGE
Arm Reachability Mask File	**MAR**	Derived	MASK

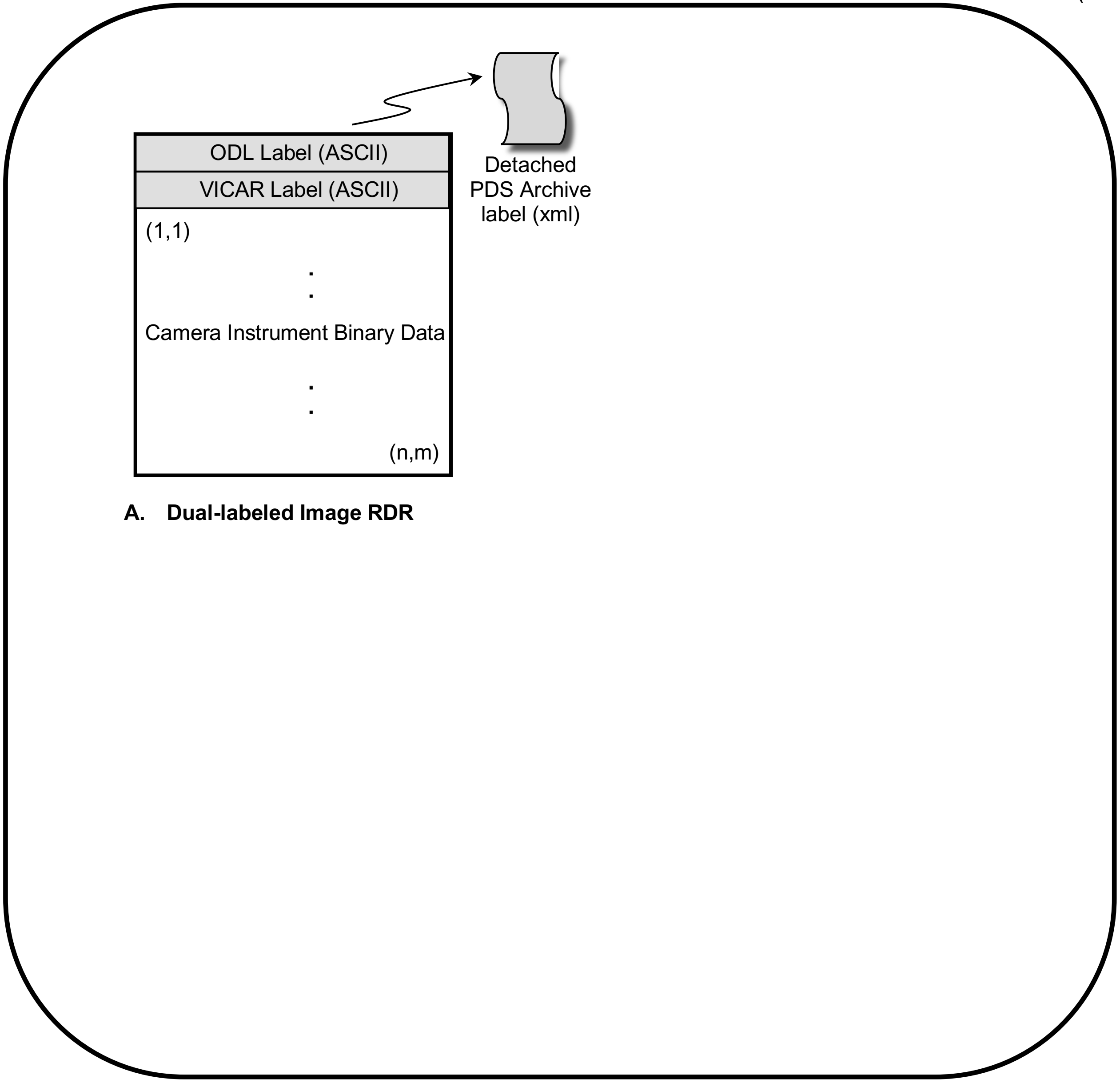
Data Product	Type				PDS Proc. Level	Derived Image Type
Solar Energy	**SEN**				Derived	SOLAR_ENERGY_MAP
Slope Heading	**SHD**				Derived	SLOPE_HEADING_MAP
Slope Heading with Overlay	**SHO**				Derived	IMAGE
Slope with Overlay	**SLO**				Derived	IMAGE
Slope	**SLP**				Derived	SLOPE_MAP
Slope Magnitude	**SMG**				Derived	SLOPE_MAGNITUDE_MAP
Slope Magnitude with Overlay	**SMO**				Derived	IMAGE
Slope Northerly Tilt with Overlay	**SNO**				Derived	IMAGE
Slope Northerly Tilt	**SNT**				Derived	NORTHERLY_TILT_MAP
Slope Radial Direction	**SRD**				Derived	RADIAL_SLOPE_MAP
Incidence, Emission, Phase angles Filled	**IEF**				Derived	IEP_MAP
Incidence, Emission, Phase angles	**IEP**				Derived	IEP_MAP
Terrain Class Confidence	**TEN**				Derived	TERRAIN_CLASSIFICATION_MAP
Terrain Class	**TER**				Derived	TERRAIN_CLASSIFICATION_MAP
Mosaic Input Index File	**IDX**				Derived	IDX_MAP
Mosaic Input Coregistration Map	**ICM**				Derived	ICM_MAP
Exposure Time Compensated Color	**Cxx**					
Raw DN Color		**Dxx**				
Zenith Scaled Color			**Zxx**			
Masked Color				**Mxx**		
Color, No Illuminant, 15-bit scaled integer, linear space	**CNR**	**DNR**	**ZNR**	**MNR**	Derived	IMAGE
Color, No Illuminant, 12-bit scaled integer, linear space	**CNS**	**DNS**	**ZNS**	**MNS**	Derived	IMAGE
Color, No Illuminant, Float, linear space	**CNF**	**DNF**	**ZNF**	**MNF**	Derived	IMAGE
Color, No Illuminant, Byte, linear space	**CNB**	**DNB**	**ZNB**	**MNB**	Derived	IMAGE
Color, No Illuminant, Byte, Gamma-corrected	**CNG**	**DNG**	**ZNG**	**MNG**	Derived	IMAGE
Color, Standard Illuminant, 15-bit scaled integer, linear space	**CSD**	**DSD**	**ZSD**	**MSD**	Derived	IMAGE
Color, Standard Illuminant, 12-bit scaled integer, linear space	**CSS**	**DSS**	**ZSS**	**MSS**	Derived	IMAGE
Color, Standard Illuminant, Float, linear space	**CSF**	**DSF**	**ZSF**	**MSF**	Derived	IMAGE
Color, Standard Illuminant, Byte, linear space	**CSB**	**DSB**	**ZSB**	**MSB**	Derived	IMAGE
Color, Standard Illuminant, Byte, Gamma-corrected	**CSG**	**DSG**	**ZSG**	**MSG**	Derived	IMAGE
Color, White Bal, Mars Illuminant, 15-bit scaled int, linear space	**CPD**	**DPD**	**ZPD**	**MPD**	Derived	IMAGE
Color, White Bal, Mars Illuminant, 12-bit scaled int, linear space	**CPS**	**DPS**	**ZPS**	**MPS**	Derived	IMAGE
Color, White Balanced, Mars Illuminant, Float, linear space	**CPF**	**DPF**	**ZPF**	**MPF**	Derived	IMAGE
Color, White Balanced, Mars Illuminant, Byte, linear space	**CPB**	**DPB**	**ZPB**	**MPB**	Derived	IMAGE
Color, White Balanced, Mars Illuminant, Byte, Gamma-corrected	**CPG**	**DPG**	**ZPG**	**MPG**	Derived	IMAGE
Color, White Balanced, 15-bit scaled integer, linear space	**CWD**	**DWD**	**ZWD**	**MWD**	Derived	IMAGE
Color, White Balanced, 12-bit scaled integer, linear space	**CWS**	**DWS**	**ZWS**	**MWS**	Derived	IMAGE
Color, White Balanced, Float, linear space	**CWF**	**DWF**	**ZWF**	**MWF**	Derived	IMAGE
Color, White Balanced, Byte, linear space	**CWB**	**DWB**	**ZWB**	**MWB**	Derived	IMAGE
Color, White Balanced, Byte, Gamma-corrected	**CWG**	**DWG**	**ZWG**	**MWG**	Derived	IMAGE

Table 6-6 – M2020 camera non-image RDR binary formats

Description	Product ID	# Bands	Data Type	Data Structure
Terrain Mesh	TBD	N/A	TBD	TBD
JPEG compressed	TBD	TBD	TBD	TBD

6.4 RDR Product Structure

RDR products will have three possible structures. RDRs generated by MIPL will have a VICAR label wrapped by an ODL label, see Figure 6.15, Diagram A). This is the same as the EDR format, with the exception that the binary header data are eliminated. RDR products not generated by MIPL may look the same, or they may omit the VICAR label, containing only an ODL label (Figure 5.4, Diagram B). Or, RDR products conforming to a standard other than PDS, such as JPEG compressed or certain Terrain products (Figure 5.4, Diagram C), may contain no additional labels, instead following the other standard's formatting. In any case, RDRs will also have a detached PDS label. For a description of the ODL labels, see Section 3.2.1, and for a description of the VICAR Label, see Section 3.2.2, and for a mapping between ODL and VICAR, see Section 3.2.3.



A. Dual-labeled Image RDR

Figure 6-15 – RDR Product Structures (TBD)

The RDR data product is comprised of processed versions of the raw camera data, in both single and multi-frame (mosaic or mesh) form. Most RDR data products will be in “image” format, having detached PDS labels, as well as attached ODL labels or, if generated by MIPL (IDS), dual attached ODL/VICAR labels. Non-labeled RDRs include JPEG compressed products and the Terrain products.

For Mosaic RDRs, the detached PDS label is shared amongst four types of archiveable files: a) the “.IMG” Mosaic image file, b) the “.NAV” navigation file, c), the “.BRT” brightness correction file, and d) the “.LIS” list file indentifying the mosaic component images. For a description of “b” and “c”, see Section 5.2.1.13.3. To support the four types of Mosaic files, the PDS label contains four FILE Objects in the structure. See Appendix B for an example of a Mosaic RDR detached PDS label.

7. STANDARDS USED IN GENERATING PRODUCTS

7.1 File Naming Standards

The file naming scheme has been relieved of the string length constraint imposed by the Level II 36.3 filename standard approved by PDS in 2009. PDS has migrated to a new standard that is PDS-4, and filenames are essentially limited to a character string length of 255. This is a change from the 36.3 convention that M2020 was constrained to using. Use of three- character extensions, such as ".IMG" for image EDRs and RDRs and ".DAT" for spectrum EDRs and state-of-health EDRs, is consistent with the PDS standard.

There are three file naming schemes adapted for the M2020 image and non-image data products. The first applies to the EDR data product and all Single-frame RDR data products. The second applies to all Mosaic RDR data products. The third applies to Terrain products.

The primary attributes of the filename nomenclature are:

a) Uniqueness - It must be unique unto itself without the file system's directory path. This Protects against product overwrite as files are copied/moved within the file system and external to the file system, if managed correctly.

b) Metadata - It should be comprised of metadata fields that keep file bookkeeping and sorting intuitive to the human user. Even though autonomous file processing will be managed via databases, there will always be human-in-the-loop that puts a premium on filename intuition. Secondly, the metadata fields should be smartly selected based on their value to ground processing tools, as it is less CPU-intensive to extract information from the filename than from the label.

NOTE: Most metadata information in the filename is also in the product label.

The metadata fields have been selected based on MSL, MER, and PHX lessons learned. In general, the metadata fields are arranged to achieve:

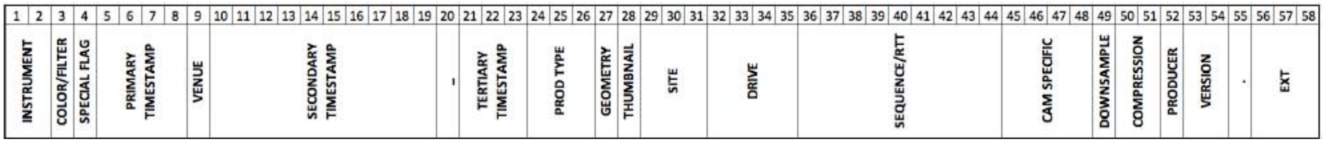
a) Sortability - Near the beginning of the filename resides primary time oriented fields such as Sol and Spacecraft Clock Start Count (SCLK). This allows for sorting of files on the file system by spacecraft data acquisition time as events occurred on Mars.

b) Readability - An effort is made to alternate Integer fields with ASCII character fields to optimize differentiation of field boundaries for the human user.

7.1.1 EDR/RDR Single-Frame Filename

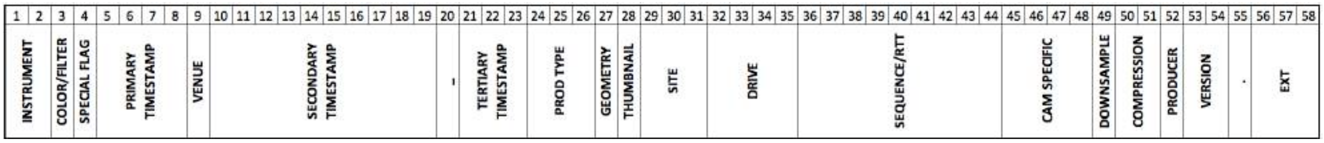
Each M2020 camera instrument EDR/RDR data product can be uniquely identified by incorporating into the product filename at minimum the Instrument ID, SCLK (or UTC), Product Type identifier, and Version number. The convention is illustrated in Table 7-1. below.

Table 7-1 – Camera instrument single-frame fileanaming convention



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	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT		COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP				VENUE	SECONDARY TIMESTAMP										_	TERTIARY TIMESTAMP			PROD TYPE			GEOMETRY	THUMBNAIL	SITE			DRIVE				SEQUENCE/RTT									CAM SPECIFIC				DOWNSAMPLE	COMPRESSION		PRODUCER	VERSION		.	EXT		

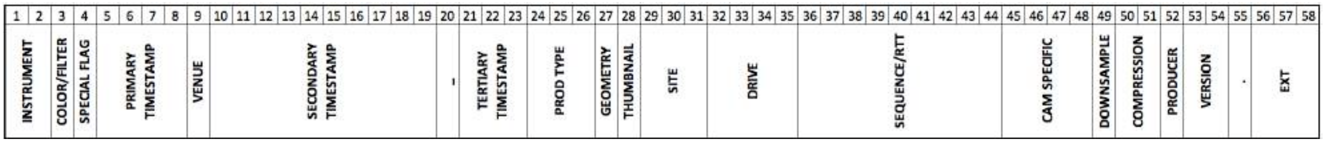
Field	Position (size, type)	Description
Instrument	01 (2, a)	Instrument Identifier: • FL : Front Hazcam Left (RCE-A) • FR : Front Hazcam Right (RCE-A) • FA : Front Hazcam Anaglyph (RCE-A) • FG : Front Hazcam Colorglyph (RCE-A) • BL : Front Hazcam Left (RCE-B) • BR : Front Hazcam Right (RCE-B) • BA : Front Hazcam Anaglyph (RCE-B) • BG : Front Hazcam Colorglyph (RCE-B) • CC : Cache Cam • EA : EDL Parachute Uplook Cam A (PUC-A) • EB : EDL Parachute Uplook Cam B (PUC-B) • EC : EDL Parachute Uplook Cam C (PUC-C) • ED : EDL Rover Downlook Cam (RDC) • EL : EDL Lander Vision System (LVS) • ES : EDL Descent Stage Downlook Cam (DSD) • EU : EDL Rover Uplook Cam (RUC) • HN : Helicopter Navigation Cam • HS : Helicopter Return To Earth Cam • ZL : Mastcam-Z Left • ZR : Mastcam-Z Right • ZA : Mastcam Anaglyph • ZG : Mastcam Colorglyph • WS : MEDA Skycam • NL : Navcam Left • NR : Navcam Right • NA : Navcam Anaglyph • NG : Navcam Colorglyph • PC : PIXL Micro Context Cam (MCC) • RL : Rear Hazcam Left • RR : Rear Hazcam Right • RA : Rear Hazcam Anaglyph • RG : Rear Hazcam Colorglyph • SC : SHERLOC Context Imager (ACI) • SE : SHERLOC Engineering (Imagers) • SI : SHERLOC Imaging (Watson) • SA : SHERLOC Imaging (Watson) Anaglyph



<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<ul>
<li>SG : SHERLOC Imaging (Watson) Colorglyph</li>
<li>SL : Watson treated as left-eye</li>
<li>SR : Watson treated as right-eye</li>
<li>LR : Supercam RMI</li>
</ul>
Non-imaging instrument identifiers (not described in this SIS):
<ul>
<li>WE : MEDA Environment</li>
<li>OX : MOXIE</li>
<li>PE : PIXL Engineering</li>
<li>PS : PIXL Spectrometer</li>
<li>LS : SuperCam Non-Imaging Data</li>
<li>SS : SHERLOC Spectrometer</li>
<li>XM : RIMFAX Mobile</li>
<li>XS : RIMFAX Stationary</li>
</ul>
</td></tr>
<tr><td>Color/Filter</td><td>03 (1, i/a)</td><td>
Color flag (see Section X.X):
<ul>
<li>E : Raw Bayer pattern</li>
<li>M : Grayscale image (Monochrome/Panchromatic)</li>
<li>A : Upper green bayer cells (G1 see Section X.X)</li>
<li>D : Lower green bayer cells (G2 see Section X.X)</li>
<li>O : Other</li>
<li>_ : N/A</li>
</ul>
For all color cameras, the flag is set based on the color space depending on whether the product is a 3-band product, or an individual band.
<table>
<tr><th>Color Type</th><th>3-Band</th><th>Band 1</th><th>Band 2</th><th>Band 3</th></tr>
<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>
</table>
For certain instruments, this flag may take on additional values indicating illumination or filter:

For PIXL, the flag can be set based on the LEDs used to illuminate the target.
<table>
<tr><th>LED Color</th><th>Flag</th></tr>
<tr><td>Red</td><td>R</td></tr>
</table>
</td></tr>
</table>

This document has been reviewed and determined not to contain export controlled technical data.



<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>Green</td><td>G</td></tr>
<tr><td>Blue</td><td>B</td></tr>
<tr><td>Multiple</td><td>W</td></tr>
<tr><td>UV</td><td>U</td></tr>
<tr><td>SLI-A (Dense)</td><td>D</td></tr>
<tr><td>SLI-B (Sparse)</td><td>S</td></tr>
<tr><td>Off</td><td>_</td></tr>
</table>
For ZCAM and SHERLOC Imaging the flag can be filter or cover state.
<table>
<tr><th>Instrument</th><th>Filters</th><th>Description</th></tr>
<tr><td>Mastcam-Z</td><td>0 – 7</td><td>See section X on ZCAM filters</td></tr>
<tr><td>Sherloc</td><td>0 - 3</td><td>Cover and LED state:

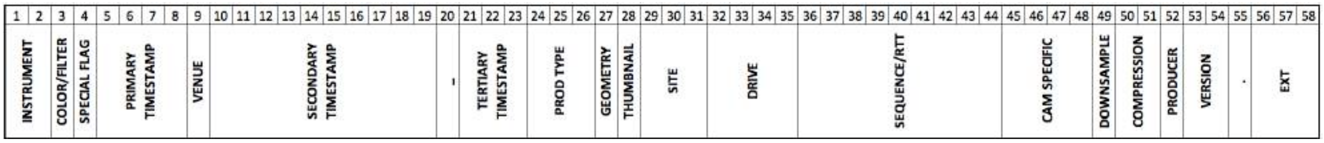
0 = cover closed, LEDs off
1 = cover open, LEDs on
2 = cover closed, LEDs off
3 = cover open, LEDs on</td></tr>
</table>
Other color flags may be defined in the future.

<b>Only “E”, “F”, or “M” or Filter/LED can appear in EDRs.</b>

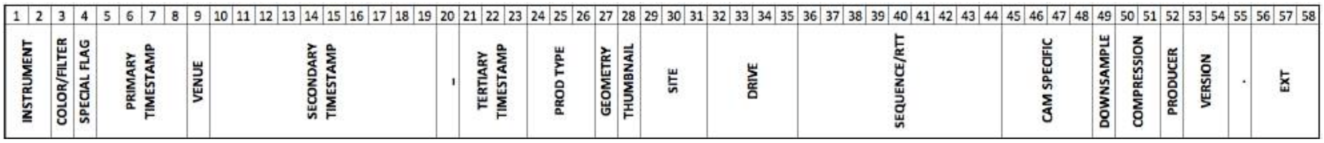
<b>For Bayered images, the filter is replaced by the color flag at the debayering step.</b></td></tr>
<tr><td>Special flag</td><td>04 (1, a)</td><td>Special Processing flag, applicable to RDRs only. EDRs always have "_".

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.

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</td></tr>
</table>

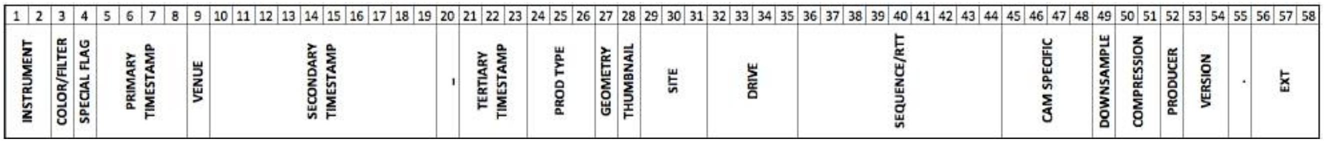


Field	Position (size, type)	Description
		sols. A database will be maintained containing all special processing designators that are used, the sols they relate to, and a description of the special processing that was done. This information will be included in the PDS archive.
Primary timestamp	05 (4, i/a)	Primary Timestamp that is of coarser granularity than the Secondary timestamp (documented later). Value type is based on either of four scenarios: Flight Cruise Year-DOY (4 alphanumeric) - This field stores two metadata items in the order: a) One alpha character in range “A-Z” to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042 b) Three integers in range “001-365” representing Day-of-Year (DOY) Flight Surface Sol (4 integer) - This field stores the 4-integer Sol (Mars solar day) of the first (i.e., lowest Clock time) acquired instrument data. Ground Test in which SCLK in NOT reset When SCLK continuously increments and does NOT repeat, there are two variants: a) Year-DOY (4 alphanumeric) - This field stores two metadata items in the order: 1. One alpha character in range “A-Z” to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042 2. Three integers in range “001-365” representing Day-of-Year (DOY) – OR – b) Sol (4 integer) - This field stores the 4-integer Sol (Mars solar day) of the first (i.e., lowest Clock time) acquired instrument data. Ground Test in which SCLK is reset When SCLK is reset and repeats, we lose time “uniqueness”. So, we have to change from SCLK to using “wall clock” derived from ERT and represent with a UTC-like format: DOY-Year (4 alphanumeric) - This field stores two metadata items in reverse order compared to the previous “Year-DOY” cases, indicating that the Secondary Time field(described later) contains ERT a) Three integers in range “001-365” representing Day-of-Year (DOY) One alpha character in range “A-Z” to designate Earth Year portion of the UTC-like time value, representing Years 2017 to 2042 This field’s value type (Sol or Year/DOY) dictates the value type for the Secondary time field (SCLK or UTC-like).



<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Scenario</th><th>Time Type</th><th>Value Format</th><th>Valid Values</th><th>Time Range</th></tr>
<tr><td rowspan="2">Flight Cruise</td><td rowspan="2">Year-DOY</td><td>A&lt;ddd&gt;,
B<ddd>,
.
.
.
Z&lt;ddd&gt;</td><td>A001, A002, … A365,
B001, B002, … B365,
.
.
.
Z001, Z002, …, Z365</td><td>2017 DOY 1 – 365,
2018 DOY 1 – 365,
.
.
.
2042 DOY 1 – 365</td></tr>
<tr><td>&lt;aaaa&gt;</td><td>“_ _ _ _” (4 underscores)</td><td>Value is out of range</td></tr>
<tr><td rowspan="2">Flight Surface</td><td rowspan="2">Sol</td><td>&lt;nnnn&gt;</td><td>0000, 0001, … 9999</td><td>0 thru 9999</td></tr>
<tr><td>&lt;aaaa&gt;</td><td>“_ _ _ _” (4 underscores)</td><td>Value is out of range</td></tr>
<tr><td rowspan="3">Ground Test SCLK NOT reset</td><td>Year-DOY</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td><td>(same as Flight Cruise)</td></tr>
<tr><td rowspan="2">Sol</td><td>&lt;nnnn&gt;</td><td>“0000”, “0001”, … “9999”</td><td>0 thru 9999</td></tr>
<tr><td>&lt;aaaa&gt;</td><td>“_ _ _ _” (4 underscores)</td><td>Value is out of range</td></tr>
<tr><td rowspan="2">Ground Test SCLK Reset</td><td rowspan="2">DOY-Year</td><td>&lt;ddd&gt;A,
<ddd>B,
.
.
.
&lt;ddd&gt;Z</td><td>001A, 002A, … 365A,
001B, 002B, … 365B,
.
.
.
001Z, 002Z, … 365Z</td><td>DOY 1 – 365 2017,
DOY 1 – 365 2018,
.
.
.
DOY 1 – 365 2042</td></tr>
<tr><td>&lt;aaaa&gt;</td><td>“_ _ _ _” (4 underscores)</td><td>Value is out of range</td></tr>
</table>
</td></tr>
<tr><td>Venue</td><td>9 (1, a)</td><td>Mission venue identifier:
• _ : Flight (surface or cruise)
• A : AVSTB
• F : FSWTB
• M : MSTB
• R : “ROASTT”
• S : “Scarecrow”
• V : VSTB

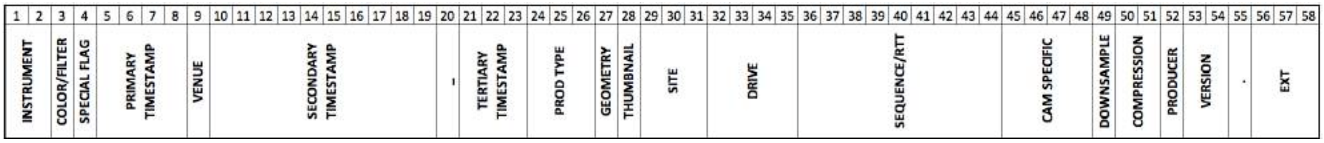
Other venue identifiers may be defined later.</td></tr>
</table>



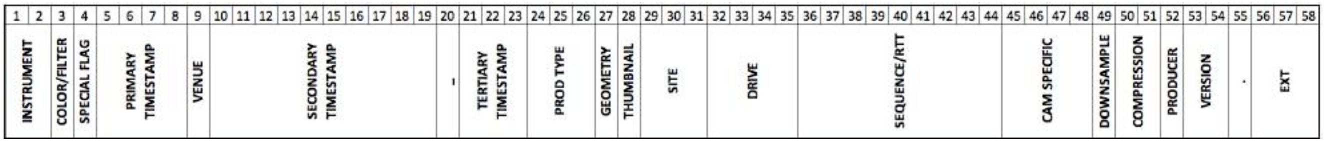
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT		COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP				VENUE	SECONDARY TIMESTAMP										_	TERTIARY TIMESTAMP			PROD TYPE			GEOMETRY	THUMBNAIL	SITE			DRIVE				SEQUENCE/RTT									CAM SPECIFIC				DOWNSAMPLE	COMPRESSION		PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
Secondary timestamp	10 (10, i)	Secondary timestamp that is of finer granularity than the Primary timestamp. Value type is based on either of four scenarios: Flight Cruise SCLK – This field stores the 10-integer SCLK (seconds). Which specific SCLK count (Start or End) is used depends on the instrument, but nominally it is the starting count of the first (i.e., lowest Clock time) acquired instrument data. Flight Surface SCLK – Same as for “Flight Cruise” Ground Test in which SCLK in NOT reset SCLK – Same as for “Flight Cruise” Ground Test in which SCLK is reset ERT - This field stores the ERT time portions Month, Day-of-month, Hour and Seconds as 10 integers in a UTC-like format This field’s value type associates with the value type of the Primary timestamp field.

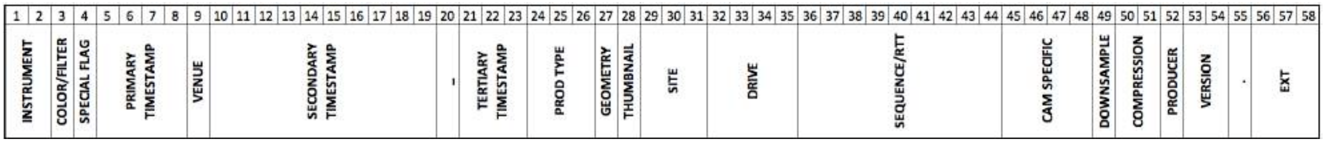
Scenario	Time Type	Value Format	Valid Values	Time Range
Flight Cruise	SCLK	<ssssssssss> (Seconds)	0000000000, 0000000001, • • • 9999999999	0 thru 9999999999
		<aaaaaaaaaa>	“__________” (10 underscores)	Value is out of range
Flight Surface	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)
Ground Test SCLK NOT reset	SCLK	(same as Flight Cruise)	(same as Flight Cruise)	(same as Flight Cruise)
Ground Test SCLK reset	ERT	<MMDDHHmmss> (Month, Day-of-month, Hour, Minute, Second)	“0101010000”, “0101010001”, • • • “1231235959”	January 1, 01:00:00 thru December 31, 23:59:59
		<aaaaaaaaaa>	“__________” (10 underscores)	Value is out of range



Field	Position (size, type)	Description
		ZCAM/SHERLOC Video Sequence Secondary Timestamp + Milliseconds for ith frame is computed as follows: First_Frame_SCLK + framerate * i Frame rate calculation (as in MSSS M2020 Camera Common Command): frametime = 1/sysclk*(400*(2*vflush+ul_y+lines)+(lines*1648*4))+exposure framerate = 1/(frametime+interval) where sysclk is 40e6 for mode 0, 20e6 for mode 9, and 10e6 for mode 1 mode is the camera head mode used to acquire the image vflush is the value of the low- order 12 bits of parameter 2 times 2 ul_y is the upper left Y coordinate of the image (specified in bits) lines is the number of lines in the output image (specified in bits) exposure is the exposure time in seconds 1648 is total number of samples per line interval: the time delay between images in units of 1 millisecond. If data acquired at the fastest possible frame rate, the value is 0. (This equation only applies if the video exposure mode is not being used. For 1280x720 video, video exposure mode increases the frame rate by about 1.75x.)"
_	20 (1, a)	Underscore for readability. Always set to "_".
Milliseconds	21 (3, i)	Milliseconds of either the SCLK or UTC.
Product type	24 (3, a)	Product identifier. EDR: See Table 4.1 for details RDR: See Table 5.3.1.1 for details.
Geometry	27 (1, a)	Linearization flag: • _ : Non-linearized (raw geometry) • L : Product has been linearized with nominal stereo partner • A : Product has been linearized with an actual stereo partner. Note that for the "A" case, an image can have multiple stereo partners and the linearized images will be different for each partner. A user will need to look in the ODL/VICAR label to determine which partner was used for linearization. All EDRs are Raw geometry ("_").
Thumbnail	28 (1, a)	Thumbnail flag: • T : Product is a thumbnail • N : Nominal Product is a non-thumbnail (full-frame, sub-frame, downsample)
Site	29 (3, i/a)	Site identifier (see section X.X on Site frames):



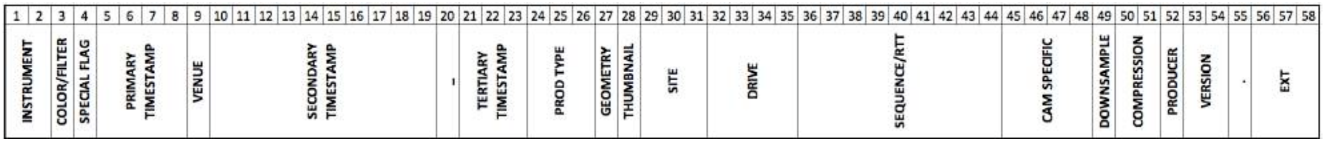
<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>Site location count from the RMC where the data was acquired.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>000, 001, …, 999</td><td>0 thru 999</td></tr>
<tr><td>A00, A01, …, A99</td><td>1000 thru 1099</td></tr>
<tr><td>B00, B01, …, B99</td><td>1100 thru 1199</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z00, Z01, … Z99</td><td>3500 thru 3599</td></tr>
<tr><td>AA0, AA1, …, AA9</td><td>3600 thru 3609</td></tr>
<tr><td>AB0, AB1, …, AB9</td><td>3610 thru 3619</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZ0, ZZ1, …, ZZ9</td><td>10350 thru 10359</td></tr>
<tr><td>AAA, AAB, …, AAZ</td><td>10360 thru 10385</td></tr>
<tr><td>ABA, ABB, …, ABZ</td><td>10386 thru 10411</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZA, ZZB, …, ZZZ</td><td>27910 thru 27935</td></tr>
<tr><td>0AA, 0AB, …, 0AZ</td><td>27936 thru 27961</td></tr>
<tr><td>0BA, 0BB, …, 0BZ</td><td>27962 thru 27987</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>7CA, 7CB, …, 7CZ</td><td>32720 thru 32745</td></tr>
<tr><td>7DA, 7DB, …, 7DV</td><td>32746 thru 32767</td></tr>
<tr><td>_ _ _</td><td>Value out of range</td></tr>
</table></td></tr>
<tr><td>Drive</td><td>32 (4, i/a)</td><td>Drive identifier:
Drive count (position within a Site location) from the RMC where the data was acquired.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>0000, 0001, …, 9999</td><td>0 thru 9999</td></tr>
<tr><td>A000, A001, …, A999</td><td>10000 thru 10999</td></tr>
<tr><td>B000, B001, …, B999</td><td>11000 thru 11999</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z000, Z001, … Z999</td><td>35000 thru 35999</td></tr>
<tr><td>AA00, AA01, …, AA99</td><td>36000 thru 36099</td></tr>
<tr><td>AB00, AB01, …, AB99</td><td>36100 thru 36199</td></tr>
</table></td></tr>
</table>



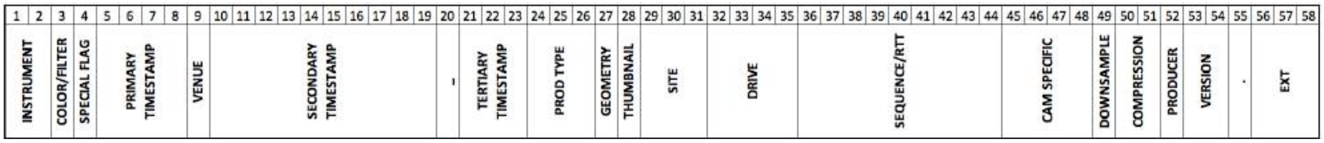
<table>
<tr><th>Field</th><th>Position
(size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>…</td><td>…</td></tr>
<tr><td>AZ00, AZ01, …, AZ99</td><td>38500 thru 38599</td></tr>
<tr><td>BA00, BA01, …, BA99</td><td>38600 thru 38699</td></tr>
<tr><td>BB00, BB01, …, BB99</td><td>38700 thru 38799</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>LJ00, LJ01, …, LJ35</td><td>65500 thru 65535</td></tr>
<tr><td>_ _ _ _</td><td>Value is out of range</td></tr>
</table>
</td></tr>
<tr><td>Seq-ID/RTT</td><td>36 (9, i/a)</td><td>Identifies sequence-ID or Round-Trip Tracking token (RTT).
• Sequence-ID : Identifier indicating the command sequenced the image was acquired from. Specific values will be assigned by the uplink team.
• RTT : Unique identifier used for tracking activities. Used in filename for PIXL ONLY.
Note: All cameras will have an RTT associated with their images but not included in the filename. PIXL is a special case where RTT and PMC are required for filename uniqueness.</td></tr>
<tr><td>Camera specific</td><td>45 (4, i/a)</td><td>Camera specific identifier.

Each instrument has a unique format:
<table>
<tr><th>Instrument</th><th>Format</th></tr>
<tr><td>ECAM
Tile</td><td>STT_</td></tr>
<tr><td>ECAM
Reconstructed</td><td>S0DR</td></tr>
<tr><td>ZCAM</td><td>SZZZ</td></tr>
<tr><td>SCAM</td><td>SNNN</td></tr>
<tr><td>PIXL MCC</td><td>PPPP</td></tr>
<tr><td>SHERLOC</td><td>SENN</td></tr>
<tr><td>MEDA Skycam
EDL Cameras
HELI</td><td>S000</td></tr>
</table>

Format key:</td></tr>
</table>

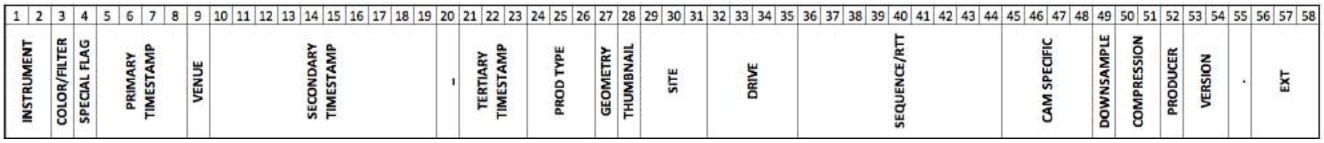


<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<ul>
<li>D : Single-resolution or multi-resolution downsampling (ECAM Reconstructed images only). Multi-resolution value represents the highest resolution tile(s) of the reconstructed image.
<table>
<tr><th>Valid values</th><th>Description</th></tr>
<tr><td>A</td><td>Single resolution (Alone)</td></tr>
<tr><td>M</td><td>Multi-resolution</td></tr>
</table>
</li>
<li>NNN : Point Number (RMI only). SuperCam MU point number that is reported in the ScamAncillaryArgs DPO. This value indicates the point number of a raster activity. Valid values are 000-099.</li>
<li>ENN : E represents the experiment ID and NN represents the total number of images that were collected as part of the experiment. Valid values are 0-9 for E and 0-99 for NN.</li>
<li>PPPP : PIXL motion counter. Valid values TBD.</li>
<li>R : Reconstruction counter (ECAM Reconstructed only). When a new tile is added that changes either the geometry or color of the reconstructed image, the counter increments by one. Valid values are 1-9, then A-Z.</li>
<li>S : Stereo partner counter. Details are TBD.
<p>All EDRs are “_”.</p>
<table>
<tr><th>Valid values</th><th>Description</th></tr>
<tr><td>_</td><td>Nominal (synchronized) stereo
or
No nominal stereo partner</td></tr>
<tr><td>A-Z then 0-9</td><td>Details TBD</td></tr>
</table>
</li>
<li>TT: Tile ID(ECAM Tile only). Identifies a unique tile within a reconstructed image. Valid values are 00-99.</li>
<li>ZZZ :Focal length in millimeters (ZCAM only).Valid values are 000-999.</li>
<li>_ : Literal underscore. As last character of camera specific field, identifies an ECAM Tile.</li>
<li>0 : Literal zero. Undefined field placeholder.</li>
</ul>
<p>Instrument teams with undefined (“0”) fields may define values at a later date.</p>
</td></tr>
<tr><td>Downsample</td><td>49 (1, i/a)</td><td><p>Downsample resolution identifier.</p>
<p>This value (n) indicates the level of downsampling applied to the image by the following equation:</p></td></tr>
</table>

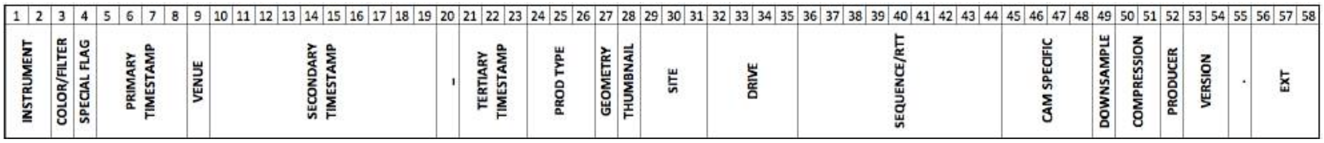


<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>Resolution = 2^n x 2^n
<table>
<tr><th>Valid values</th><th>Resolution</th></tr>
<tr><td>0</td><td>1x1</td></tr>
<tr><td>1</td><td>2x2</td></tr>
<tr><td>2</td><td>4x4</td></tr>
<tr><td>3</td><td>8x8</td></tr>
<tr><td>…</td><td>...</td></tr>
</table>
For the case of reconstructed ECAM images, this value represents the highest resolution tile(s) in the reconstructed image (least amount of downsampling).</td></tr>
<tr><td>Compression</td><td>50 (2, i/a)</td><td>Compression type identifier.

There are several modes of compression available, varying per instrument. The uplink team will decide which algorithm will provide the best results on a per sequence basis.
<table>
<tr><th>Type</th><th>Valid values</th><th>Description</th></tr>
<tr><td>JPEG
(lossy)</td><td>00
01-99
A0</td><td>Thumbnail
Jpeg quality level
Jpeg quality level 100</td></tr>
<tr><td>ICER
(lossy)</td><td>I1, I2, …, I8
I9</td><td>1 bpp, 2 bpp, …, 8 bpp
Anything higher than 8 bpp</td></tr>
<tr><td>Lossless</td><td>LI
LL
LM
LU</td><td>ICER
LOCO
Malin
Uncompressed</td></tr>
</table>
For ECAM reconstructed images, compression value represents the compression level of the best tile(s) in the reconstruction.</td></tr>
<tr><td>Producer</td><td>52 (1, a)</td><td>Identifier for the institution/team that created this product:
<ul>
<li>J : JPL (IDS/MIPL)</li>
<li>A: ASU (Mastcam-Z team)</li>
<li>P : Principal investigator of instrument.</li>
</ul>
<table>
<tr><th>Instrument</th><th>PI</th></tr>
<tr><td>ECAM</td><td>JPL</td></tr>
<tr><td>ZCAM</td><td>ASU (Tempe, AZ)</td></tr>
</table></td></tr>
</table>



<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>SCAM RMI</td><td>IRAP (France)</td></tr>
<tr><td>PIXL MCC</td><td>JPL</td></tr>
<tr><td>SHERLOC</td><td>JPL</td></tr>
<tr><td>MEDA Skycam</td><td>Ministry of Education and Science (Spain)</td></tr>
<tr><td>EDL Cameras</td><td>JPL</td></tr>
<tr><td>HELI NAV/RTE</td><td>JPL</td></tr>
</table>
<ul>
<li>A – I, K – O, Q – Z : Co-I to be identified per instrument at the discretion of the instrument PI.</li>
<li>_ : undefined/other</li>
</ul>
Other producer codes will be added in the future.</td></tr>
<tr><td>Version</td><td>53 (2, a)</td><td>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>01, 02 …, 99</td><td>1 thru 99</td></tr>
<tr><td>A0, A1, …, A9</td><td>100 thru 109</td></tr>
<tr><td>AA, AB, …, AZ</td><td>110 thru 135</td></tr>
<tr><td>B0, B1, B2 …, B9</td><td>136 thru 145</td></tr>
<tr><td>BA, BB, …, BZ</td><td>146 thru 171</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z0, Z1, …, Z9</td><td>1000 thru 1009</td></tr>
<tr><td>ZA, ZB, …, ZZ</td><td>1010 thru 1035</td></tr>
<tr><td>_ _</td><td>Value is out of range</td></tr>
</table>
Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</td></tr>
<tr><td>.</td><td>55 (1, a)</td><td>Separator for filename and extension. Always set to “.”</td></tr>
<tr><td>Extension</td><td>56 (3, a)</td><td>File extension:
<ul>
<li>VIC : Used for VICAR files.</li>
<li>IMG : Same as .VIC with ODL label.</li>
</ul></td></tr>
</table>



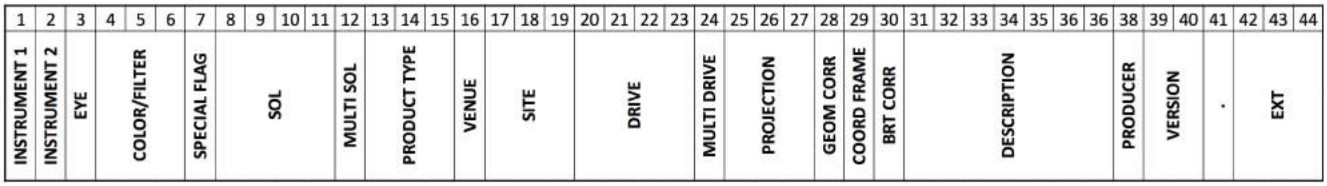
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
INSTRUMENT		COLOR/FILTER	SPECIAL FLAG	PRIMARY TIMESTAMP				VENUE	SECONDARY TIMESTAMP										-	TERTIARY TIMESTAMP			PROD TYPE			GEOMETRY	THUMBNAIL	SITE			DRIVE				SEQUENCE/RTT									CAM SPECIFIC				DOWNSAMPLE	COMPRESSION		PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
		• TIF : TIFF formatted image file. (no label) • JPG : JPEG formatted image file. (no label) • PNG : PNG formatted image file. (no label) • TXT : ASCII text file. • iv : Inventor-format file • ht : Height-map file (VICAR format) • rgb : Skin file in SGI RGB format • obj: Mesh file in Wavefront OBJ format • mtl : OBJ Material file • png : Lower case for meshes • mlp : MeshLab project file • XML : xml file • MXML : masked XYZ (MXY) xml file Note: other file extensions to be added in the future

7.1.2 Mosaic RDR Filename

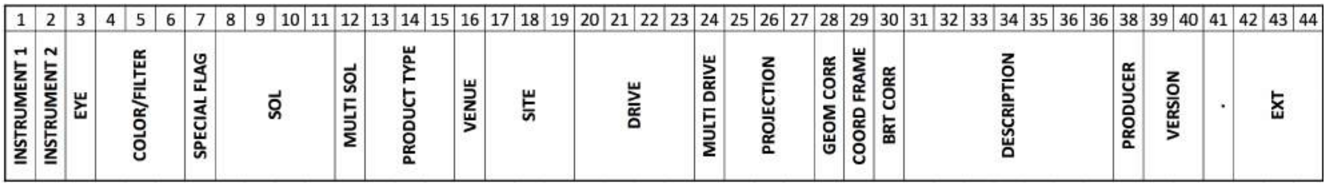
Each M2020 camera instrument RDR mosaic data product can be uniquely identified by incorporating into the product filename at least the Instrument ID Sol (or DOY), Product Type identifier, Site, Drive, and Version number. The convention is illustrated in table 7-2 below.

Table 7-2 – Camera instrument mosaic filenaming convention



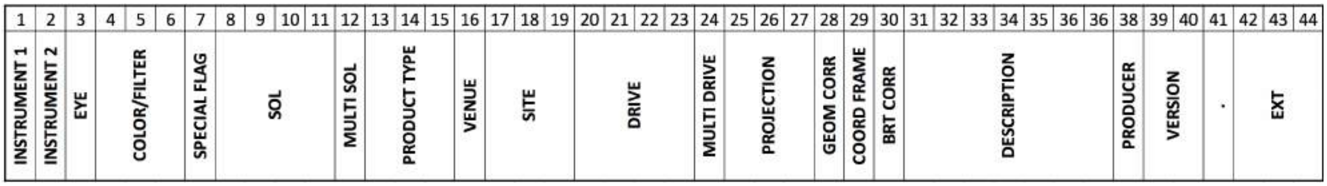
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
Instrument 1	1 (1,a)	First Instrument Identifier: • N : Navcam • Z : Mastcam-Z • L : Supercam RMI • P : PIXL Micro Context Cam (MCC) • I : SHERLOC Imaging (WATSON) • C : SHERLOC ACI



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	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
		• F : Front Hazcam (RCE-A) • B : Front Hazcam (RCE-B) • R : Rear Hazcam • E : EDL Camera • V : Mars Helicopter Navigation Cam
Instrument 2	2 (1,a)	Second Instrument Identifier: • N : Navcam • Z : Mastcam-Z • L : Supercam RMI • P : PIXL Micro Context Cam (MCC) • I : SHERLOC Imaging (WATSON) • C : SHERLOC ACI • F : Front Hazcam (RCE-A) • B : Front Hazcam (RCE-B) • R : Rear Hazcam • E : EDL Camera • V : Mars Helicopter Navigation Cam • X : Use when there are more than two instruments • _ : 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). • A : Anaglyph • C : Colorglyph • X : Mixed, used for by products which have more than one image. • L : Left eye • M : Mono • R : Right eye • S : Stereo 2-band
Color/Filter	4 (3,a)	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: • 0 or _: Filter not applicable • M : Grayscale image (Monochrome/Panchromatic) • U : UV (PIXL only)

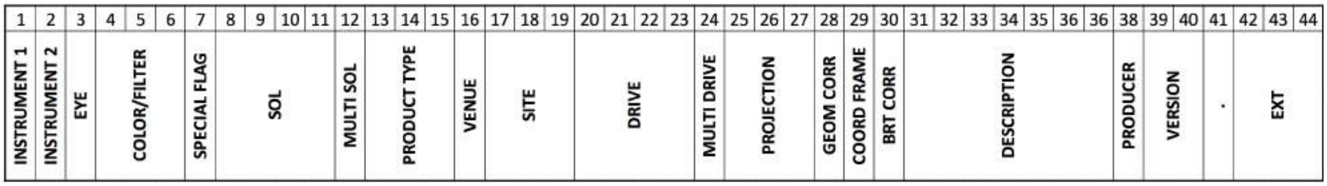


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
		<table><tr><th>Color Type</th><th>3-Band</th><th>Band 1</th><th>Band 2</th><th>Band 3</th></tr><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></table> Other color flags may be defined in the future.
Special	7 (1,a)	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. 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. 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.
Sol	8 (4,i)	• Sol : for flight (Cruise & Surface) or flight-like ground tests when the Secondary timestamp continuously increments, this field stores the 4-integer Sol (Mars solar day) of the first (i.e lowest) clock time acquired instrument data. • Year : for ground tests when the Secondary timestamp resets and repeats, the uniqueness as a timestamp is lost. Wall Clock is used to restore uniqueness, represented in a UTC-like format across Primary, Secondary, and Tertiary time fields. This field is used to store the Earth Year portion of the UTC-like time value, designated by a “Y” followed by the last three digits of the year. <table><tr><th colspan="2">Time Type</th><th>Values</th><th>Valid Time Range</th></tr><tr><td>Sol</td><td>Cruise</td><td>0000</td><td>NULL</td></tr></table>

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	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td>Surface</td><td>0001 – 9999</td><td>1 thru 9999</td></tr>
<tr><td></td><td></td><td>_ _ _ _</td><td>Value is out of range</td></tr>
<tr><td>Year of UTC</td><td>Cruise Test</td><td>Y000</td><td>NULL</td></tr>
<tr><td></td><td>Surface Test</td><td>Y017 – Y099</td><td>2017 thru 2099</td></tr>
<tr><td></td><td></td><td>_ _ _ _</td><td>Value is out of range</td></tr>
</table>
</td></tr>
<tr><td>Multisol</td><td>12 (1,a)</td><td>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:
• _ : Flags mosaic as single sol (insignificant multiple sol contribution)
• X : Flags mosaic as containing significant content from multiple sols</td></tr>
<tr><td>Product Type</td><td>13 (3,a)</td><td>Product Identifier. See Table 5.3.1.1 for details.</td></tr>
<tr><td>Venue</td><td>16 (1,a)</td><td>Mission venue identifier:
• _ : Flight (surface or cruise)
• A : AVSTB
• F : FSWTB
• M : MSTB
• R : “ROASTT”
• S : “Scarecrow”
• V : VSTB
• Other venue identifiers may be defined later.</td></tr>
<tr><td>Site</td><td>17 (3, i/a)</td><td>Site identifier (see section X.X on Site frames):
Site location count from the RMC where the data was acquired.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>000, 001, …, 999</td><td>0 thru 999</td></tr>
<tr><td>A00, A01, …, A99</td><td>1000 thru 1099</td></tr>
<tr><td>B00, B01, …, B99</td><td>1100 thru 1199</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z00, Z01, … Z99</td><td>3500 thru 3599</td></tr>
</table>
</td></tr>
</table>



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	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

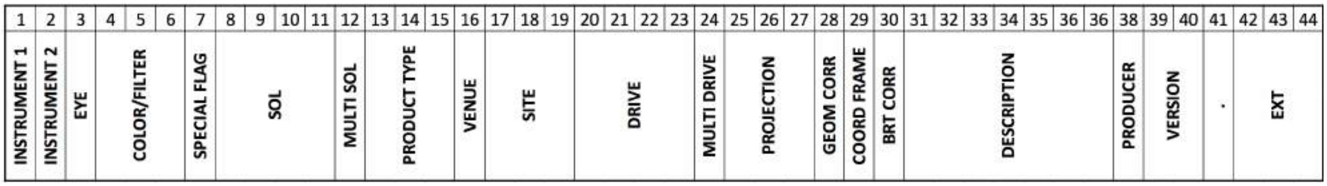
<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>AA0, AA1, …, AA9</td><td>3600 thru 3609</td></tr>
<tr><td>AB0, AB1, …, AB9</td><td>3610 thru 3619</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZ0, ZZ1, …, ZZ9</td><td>10350 thru 10359</td></tr>
<tr><td>AAA, AAB, …, AAZ</td><td>10360 thru 10385</td></tr>
<tr><td>ABA, ABB, …, ABZ</td><td>10386 thru 10411</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZA, ZZB, …, ZZZ</td><td>27910 thru 27935</td></tr>
<tr><td>0AA, 0AB, …, 0AZ</td><td>27936 thru 27961</td></tr>
<tr><td>0BA, 0BB, …, 0BZ</td><td>27962 thru 27987</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>7CA, 7CB, …, 7CZ</td><td>32720 thru 32745</td></tr>
<tr><td>7DA, 7DB, …, 7DV</td><td>32746 thru 32767</td></tr>
<tr><td>_ _ _</td><td>Value out of range</td></tr>
</table>
</td></tr>
<tr><td>Drive</td><td>20 (4, i/a)</td><td>Drive identifier:
Drive count (position within a Site location) from the RMC where the data was acquired.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>0000, 0001, …, 9999</td><td>0 thru 9999</td></tr>
<tr><td>A000, A001, …, A999</td><td>10000 thru 10999</td></tr>
<tr><td>B000, B001, …, B999</td><td>11000 thru 11999</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z000, Z001, … Z999</td><td>35000 thru 35999</td></tr>
<tr><td>AA00, AA01, …, AA99</td><td>36000 thru 36099</td></tr>
<tr><td>AB00, AB01, …, AB99</td><td>36100 thru 36199</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>AZ00, AZ01, …, AZ99</td><td>38500 thru 38599</td></tr>
<tr><td>BA00, BA01, …, BA99</td><td>38600 thru 38699</td></tr>
<tr><td>BB00, BB01, …, BB99</td><td>38700 thru 38799</td></tr>
<tr><td>…</td><td>…</td></tr>
</table>
</td></tr>
</table>

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	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td><table><tr><td>LJ00, LJ01, …, LJ35</td><td>65500 thru 65535</td></tr><tr><td>_ _ _ _</td><td>Value is out of range</td></tr></table></td></tr>
<tr><td>Multidrive</td><td>24 (1,a)</td><td>Flag indicating that a significant percentage of the data content was acquired on more than a single Drive position, i.e. across multiple drives. Valid values:
• _ : Flags mosaic as single drive position
• X : Flags mosaic as containing content from multiple Drive positions</td></tr>
<tr><td>Projection</td><td>25 (3,a)</td><td>Projection type. Valid values:
• CYL - Cylindrical
• CYP - Cylindrical-Perspective
• ORT - Orthographic
• ORR - Orthorectified
• PER - Perspective
• POL - Polar
• VRT - Vertical (special case of Orthographic)</td></tr>
<tr><td>Geometric Correction</td><td>28 (1,a)</td><td>Geometric correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are:
• _ : No correction (raw pointing)
• A : Auto-correction via tiepointing
• F : Auto-correction via tiepointing & auto-registration with fiducials
• G : Auto-correction via tiepointing & manual registration with fiducials
• T : Manual tiepointing
• R : Manual tiepointing & auto-registration with fiducials
• M : Manual tiepointing & manual registration with fiducials
• O : Other correction not listed above</td></tr>
<tr><td>Coordinate Frame</td><td>29 (1,a)</td><td>Coordinate system (frame) type. Valid values:
• S : Site frame
• L : Local Level frame
• R : Rover Frame
• U : Untilt (CYP only, is Rover frame with rotation)
• O : Other</td></tr>
<tr><td>Brt Corr.</td><td>30 (1, a)</td><td>Brightness correction type indicator. Specifies the correction type that was applied to the largest percentage of data content. Valid values are:
• _ : No correction</td></tr>
</table>

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	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<ul>
<li>B : Automatic brightness adjustment (multiplicative and/or additive factor applied to each frame)</li>
<li>M : Manual brightness adjustment (same factors as “B”)</li>
<li>V : Anti-vignetting adjustment applied to some or all frames</li>
<li>G : General brightness correction</li>
<li>A : General brightness correction that can vary across the frame (automatic)</li>
<li>O : Other correction not listed above</li>
</ul>
</td></tr>
<tr><td>Description</td><td>31 (7,a)</td><td>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.</td></tr>
<tr><td>Producer</td><td>38 (1,a)</td><td>Identifier for the institution/team that created this product:
<ul>
<li>J : JPL (IDS/MIPL)</li>
<li>A: ASU (Mastcam-Z team)</li>
<li>P : Principal investigator of instrument.</li>
</ul>
<table>
<tr><th>Instrument</th><th>PI</th></tr>
<tr><td>ECAM</td><td>JPL</td></tr>
<tr><td>ZCAM</td><td>ASU (Tempe, AZ)</td></tr>
<tr><td>SCAM RMI</td><td>IRAP (France)</td></tr>
<tr><td>PIXL MCC</td><td>JPL</td></tr>
<tr><td>SHERLOC</td><td>JPL</td></tr>
<tr><td>MEDA Skycam</td><td>Ministry of Education and Science (Spain)</td></tr>
<tr><td>EDL Cameras</td><td>JPL</td></tr>
<tr><td>HELI RTE/NAV</td><td>JPL</td></tr>
</table>
<ul>
<li>A – I, K – O, Q – Z : Co-I to be identified per instrument at the discretion of the instrument PI.</li>
<li>_ : undefined/other</li>
</ul>
Other producer codes will be added in the future.</td></tr>
<tr><td>Version</td><td>39 (2,a)</td><td>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists.
<table>
<tr><th>Values</th><th>Range</th></tr>
</table>
</td></tr>
</table>

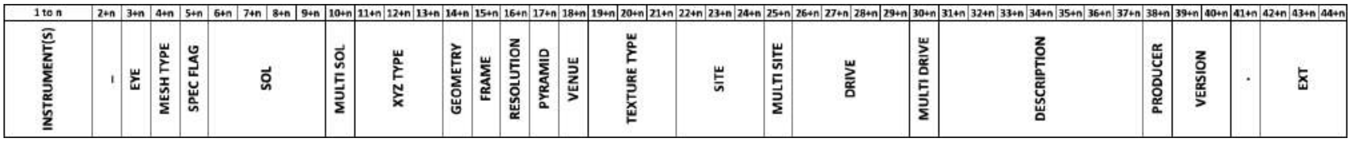


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	36	38	39	40	41	42	43	44
INSTRUMENT 1	INSTRUMENT 2	EYE	COLOR/FILTER			SPECIAL FLAG	SOL				MULTI SOL	PRODUCT TYPE			VENUE	SITE			DRIVE				MULTI DRIVE	PROJECTION			GEOM CORR	COORD FRAME	BRT CORR	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><th>Field</th><th>Position (size, type)</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>01, 02 …, 99</td><td>1 thru 99</td></tr>
<tr><td>A0, A1, …, A9</td><td>100 thru 109</td></tr>
<tr><td>AA, AB, …, AZ</td><td>110 thru 135</td></tr>
<tr><td>B0, B1, B2 …, B9</td><td>136 thru 145</td></tr>
<tr><td>BA, BB, …, BZ</td><td>146 thru 171</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z0, Z1, …, Z9</td><td>1000 thru 1009</td></tr>
<tr><td>ZA, ZB, …, ZZ</td><td>1010 thru 1035</td></tr>
<tr><td>_ _</td><td>Value is out of range</td></tr>
</table>
<ul><li>Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</li></ul>
</td></tr>
<tr><td>.</td><td>41 (1,a)</td><td>Separator for filename and extension. Always set to “.”</td></tr>
<tr><td>Extension</td><td>42 (3,a)</td><td>File extension:
<ul>
<li>VIC : Used for VICAR files.</li>
<li>IMG : Same as .VIC with ODL label.</li>
<li>TIF : TIFF formatted image file. (no label)</li>
<li>JPG : JPEG formatted image file. (no label)</li>
<li>PNG : PNG formatted image file. (no label)</li>
<li>TXT : ASCII text file.</li>
<li>xml : PDS4 label file</li>
</ul></td></tr>
</table>

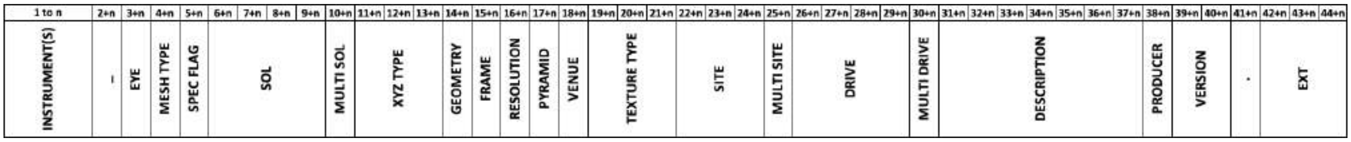
7.1.3 Unified Terrain Mesh RDR Filename

Table 7-3 – Camera instrument unified terrain mesh filenaming convention



1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	_	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

Field	Position (size, type)	Description
Instrument	1+n (n,a)	First Instrument Identifier: • N : Navcam • Z : Mastcam-Z • L : Supercam RMI • O: Orbital Mesh • P : PIXL Micro Context Cam (MCC) • I : SHERLOC Imaging (WATSON) • C : SHERLOC ACI • F : Front Hazcam (RCE-A) • B : Front Hazcam (RCE-B) • R : Rear Hazcam • E : EDL Camera • H : Helicopter Return to Earth Cam • V : Helicopter Navigation Cam
_	2+n (1,a)	Underscore for readability. Always set to "_".
Eye	3+n (1,a)	Identifies the eye of the data used to make this mesh. • L : Left-eye • R : Right-eye
Mesh Type	4+n (1,a)	Identifies the type of mesh. • T : Tactical • C : Contextual • H : Helicopter • O : Other
Special	5+n (1,a)	Special Processing flag. The special processing character is used to indicate off-nominal or special processing of the mesh. Examples include use of different correlation parameters, special stretches to eliminate shadows, reprocessing with different camera pointing, etc. 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. 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.

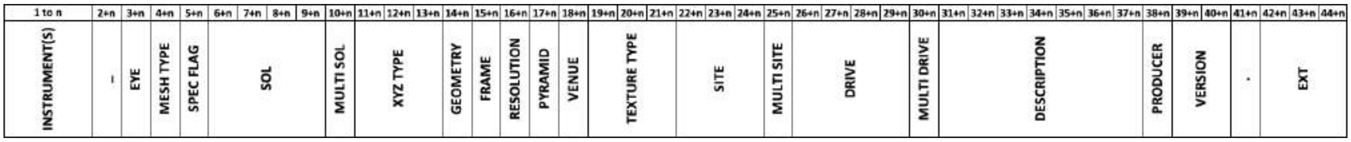


1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><td>Sol</td><td>6+n (4,i)</td><td>• Sol : for flight (Cruise & Surface) or flight-like ground tests when the Secondary timestamp continuously increments, this field stores the 4-integer Sol (Mars solar day) of the first (i.e lowest) clock time acquired instrument data.
• Year : for ground tests when the Secondary timestamp resets and repeats, the uniqueness as a timestamp is lost. Wall Clock is used to restore uniqueness, represented in a UTC-like format across Primary, Secondary, and Tertiary time fields. This field is used to store the Earth Year portion of the UTC-like time value, designated by a “Y” followed by the last three digits of the year.
<table>
<tr><th colspan="2">Time Type</th><th>Values</th><th>Valid Time Range</th></tr>
<tr><td rowspan="3">Sol</td><td>Cruise</td><td>0000</td><td>NULL</td></tr>
<tr><td rowspan="2">Surface</td><td>0001 – 9999</td><td>1 thru 9999</td></tr>
<tr><td>_ _ _ _</td><td>Value is out of range</td></tr>
<tr><td rowspan="3">Year of UTC</td><td>Cruise Test</td><td>Y000</td><td>NULL</td></tr>
<tr><td rowspan="2">Surface Test</td><td>Y017 – Y099</td><td>2017 thru 2099</td></tr>
<tr><td>_ _ _ _</td><td>Value is out of range</td></tr>
</table>
</td></tr>
<tr><td>Multi sol</td><td>10+n (1,a)</td><td>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:
• _ : Flags mosaic as single sol (insignificant multiple sol contribution)
• X : Flags mosaic as containing significant content from multiple sols</td></tr>
<tr><td>XYZ Type</td><td>11+n (3,a)</td><td>Indicates the XYZ type used for mesh data:
• See Table 6.3.1.2</td></tr>
<tr><td>Geometry</td><td>14+n (1,a)</td><td>Linearization flag:
• _ : Non-linearized (raw geometry)
• L : Product has been linearized with nominal stereo partner
• A : Product has been linearized with an actual stereo partner.

Note that for the “A” case, an image can have multiple stereo partners and the linearized images will be different for each partner. A user will need to look in the ODL/VICAR label to determine which partner was used for linearization.</td></tr>
<tr><td>Frame</td><td>15+n (1,a)</td><td>Coordinate system (frame) type. Valid values:
• S : Site frame</td></tr>
</table>

This document has been reviewed and determined not to contain export controlled technical data.

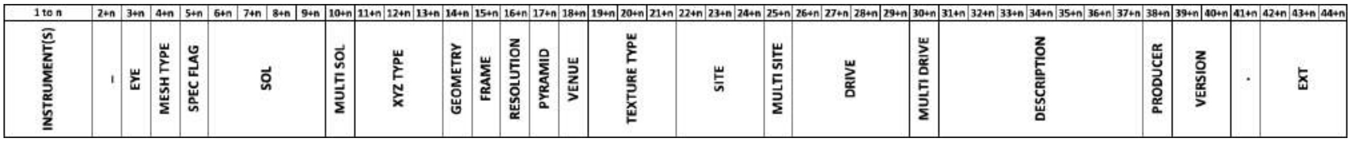


1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><td></td><td></td><td>• L : Local Level frame
• R : Rover Frame
• O : Other</td></tr>
<tr><td>Resolution</td><td>16+n (1,i/a)</td><td>Indicates if the input data is comprised of single or multi-resolution images (ECAM reconstructed). Number value indicates level of pixel averaging.
• 1 : 1x1
• 2 : 2x2
• 4 : 4x4
• 8 : 8x8
• M : multi-resolution</td></tr>
<tr><td>Pyramid</td><td>17+n (1,i)</td><td>Indicates the mesh pyramid level or downsampling relative to the highest resolution of the mesh. (powers of 2).
• 0 : full resolution
• 1 : 2x downsampling
• 2 : 4x downsampling
• 3 : 8x downsampling
• 4 : 16x downsampling
• <i>n</i> : 2^n downsampling</td></tr>
<tr><td>Venue</td><td>18+n (1,a)</td><td>Mission venue identifier:
• _ : Flight (surface or cruise)
• A : AVSTB
• F : FSWTB
• M : MSTB
• R : “ROASTT”
• S : “Scarecrow”
• V : VSTB

Other venue identifiers may be defined later.</td></tr>
<tr><td>Texture Type</td><td>19+n (3,a)</td><td>Indicates the product type used for the mesh skin:
• See Table 6.3.1.2
• _ _ _: no mesh skin</td></tr>
<tr><td>Site</td><td>22+n (3,i/a)</td><td>Site identifier (see section X.X on Site frames):

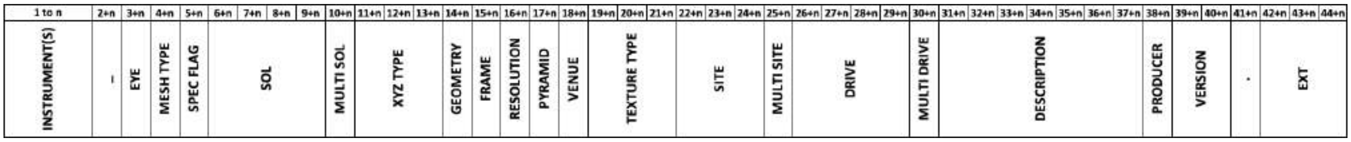
Site location count from the RMC where the data was acquired.<table><tr><th>Values</th><th>Range</th></tr><tr><td>000, 001, …, 999</td><td>0 thru 999</td></tr><tr><td>A00, A01, …, A99</td><td>1000 thru 1099</td></tr><tr><td>B00, B01, …, B99</td><td>1100 thru 1199</td></tr></table></td></tr>
</table>



1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

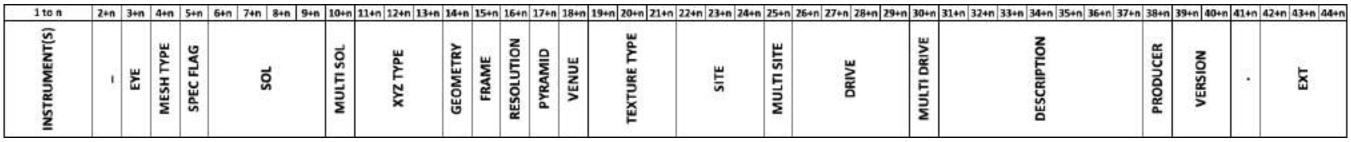
<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z00, Z01, … Z99</td><td>3500 thru 3599</td></tr>
<tr><td>AA0, AA1, …, AA9</td><td>3600 thru 3609</td></tr>
<tr><td>AB0, AB1, …, AB9</td><td>3610 thru 3619</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZ0, ZZ1, …, ZZ9</td><td>10350 thru 10359</td></tr>
<tr><td>AAA, AAB, …, AAZ</td><td>10360 thru 10385</td></tr>
<tr><td>ABA, ABB, …, ABZ</td><td>10386 thru 10411</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>ZZA, ZZB, …, ZZZ</td><td>27910 thru 27935</td></tr>
<tr><td>0AA, 0AB, …, 0AZ</td><td>27936 thru 27961</td></tr>
<tr><td>0BA, 0BB, …, 0BZ</td><td>27962 thru 27987</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>7CA, 7CB, …, 7CZ</td><td>32720 thru 32745</td></tr>
<tr><td>7DA, 7DB, …, 7DV</td><td>32746 thru 32767</td></tr>
<tr><td>_ _ _</td><td>Value out of range</td></tr>
</table>
</td>
</tr>
<tr>
<td>Multisite</td>
<td>25+n (1,a)</td>
<td>Flag indicating that a significant percentage of the data content was acquired on more than a single Drive position, i.e. across multiple drives. Valid values:
• _ : Flags mesh as single Site
• X : Flags mesh as containing content from multiple Sites</td>
</tr>
<tr>
<td>Drive</td>
<td>26+n (4, i/a)</td>
<td>Drive identifier:

Drive count (position within a Site location) from the RMC where the data was acquired.
<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>0000, 0001, …, 9999</td><td>0 thru 9999</td></tr>
<tr><td>A000, A001, …, A999</td><td>10000 thru 10999</td></tr>
<tr><td>B000, B001, …, B999</td><td>11000 thru 11999</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z000, Z001, … Z999</td><td>35000 thru 35999</td></tr>
<tr><td>AA00, AA01, …, AA99</td><td>36000 thru 36099</td></tr>
<tr><td>AB00, AB01, …, AB99</td><td>36100 thru 36199</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>AZ00, AZ01, …, AZ99</td><td>38500 thru 38599</td></tr>
</table>
</td>
</tr>
</table>



1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

<table>
<tr><td></td><td></td><td><table>
<tr><td>BA00, BA01, …, BA99</td><td>38600 thru 38699</td></tr>
<tr><td>BB00, BB01, …, BB99</td><td>38700 thru 38799</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>LJ00, LJ01, …, LJ35</td><td>65500 thru 65535</td></tr>
<tr><td>_ _ _ _</td><td>Value is out of range</td></tr>
</table></td></tr>
<tr><td>Multidrive</td><td>30+n (1,a)</td><td>Flag indicating that a significant percentage of the data content was acquired on more than a single Drive position, i.e. across multiple drives. Valid values:
• _ : Flags mesh as single Drive position
• X : Flags mesh as containing content from multiple Drive positions</td></tr>
<tr><td>Description</td><td>31+n (7,a)</td><td>General purpose mesh 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.</td></tr>
<tr><td>Producer</td><td>38+n (1,a)</td><td>Identifier for the institution/team that created this product:
• J : JPL (IDS/MIPL)
• A: ASU (Mastcam-Z team)
Other producer codes will be added in the future.</td></tr>
<tr><td>Version</td><td>39+n (2,a)</td><td>Product version number. Increments by one whenever a previously generated file with an otherwise identical filename exists.<table>
<tr><th>Values</th><th>Range</th></tr>
<tr><td>01, 02 …, 99</td><td>1 thru 99</td></tr>
<tr><td>A0, A1, …, A9</td><td>100 thru 109</td></tr>
<tr><td>AA, AB, …, AZ</td><td>110 thru 135</td></tr>
<tr><td>B0, B1, B2 …, B9</td><td>136 thru 145</td></tr>
<tr><td>BA, BB, …, BZ</td><td>146 thru 171</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>Z0, Z1, …, Z9</td><td>1000 thru 1009</td></tr>
<tr><td>ZA, ZB, …, ZZ</td><td>1010 thru 1035</td></tr>
<tr><td>_ _</td><td>Value is out of range</td></tr>
</table>• Every version need not exist. E.g. version 01, 02, and 04 may exist but not 03. In general, the highest-numbered version represents the best version of that product. This field increments independently of all fields.</td></tr>
<tr><td>.</td><td>40+n (1,a)</td><td>Separator for filename and extension. Always set to “.”</td></tr>
</table>



1 to n	2+n	3+n	4+n	5+n	6+n	7+n	8+n	9+n	10+n	11+n	12+n	13+n	14+n	15+n	16+n	17+n	18+n	19+n	20+n	21+n	22+n	23+n	24+n	25+n	26+n	27+n	28+n	29+n	30+n	31+n	32+n	33+n	34+n	35+n	36+n	37+n	38+n	39+n	40+n	41+n	42+n	43+n	44+n
INSTRUMENT(S)	-	EYE	MESH TYPE	SPEC FLAG	SOL				MULTI SOL	XYZ TYPE			GEOMETRY	FRAME	RESOLUTION	PYRAMID	VENUE	TEXTURE TYPE			SITE			MULTI SITE	DRIVE				MULTI DRIVE	DESCRIPTION							PRODUCER	VERSION		.	EXT		

Extension	41+n (3,a)	File extension: • iv • ht • mod • mtl • obj

7.2 PDS Standards

The M2020 camera instrument EDR data product complies with Planetary Data System standards for file formats and labels, as specified in the PDS Standards Reference [Ref 14]. See Section 4.2 for a description of the PDS Label and the specific conventions adopted by M2020.

7.3 Time Standards

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

7.4 Coordinate Frame Standards

The M2020 Frame Manager defines several dozen coordinate frames, which can be used for commanding pointing among other things. Refer to the Pointing, Positioning, Phasing and Coordinate Systems (PPPCS) document [Ref 1] or the Surface Attitude, Positioning and Pointing (SAPP) Functional Design Description (FDD) [Ref 2] for more details on all these coordinate frames. Only a few of them are used by the products and processes described by this SIS. This subset is described in detail in this section. The only place in this SIS where the full set of frames can appear is in the INSTRUMENT_COORD_FRAME_ID label, which is a command echo.

A subset of these frames needed for a specific image or data set are defined by the *_COORDINATE_SYSTEM groups.

Note that the PLACES database [Ref X] maintains both telemetered and re-localized versions of the Site and Rover Nav frames at every available index.

Table 7-4 – Coordinate frames used for M2020 surface operations

Frame Name (Label Keyword Value)	Short Name (SAPP FDD)	Reference Frame (Used to Define)	Coordinate Frame	
			Origin	Orientation
ROVER_NAV_FRAME	RNAV	Enclosing SITE_FRAME	Attached to rover	Aligned with rover
ROVER_MECH_FRAME	RMECH	ROVER_NAV_FRAME	Attached to rover	Aligned with rover
LOCAL_LEVEL_FRAME	LL	Enclosing SITE_FRAME	Attached to rover (coincident with Rover Nav Frame)	North/East/Nadir
SITE_FRAME	SITE(n)	Previous SITE_FRAME	Attached to surface	North/East/Nadir
RSM_HEAD_FRAME	RSM_HEAD	ROVER_NAV_FRAME	Attached to mast head	Aligned with pointing of mast head. This corresponds to RSM_HEAD in the Frame Manager
Arm Frames: TBD ARM_BASE_FRAME ARM_TURRET_FRAME ARM_DRILL_FRAME ARM_GDST_ FRAME ARM_TOOL_FRAME ARM_WATSON_FRAME ARM_SHERLOC_FRAME ARM_PIXL_FRAME	Arm Frames: TBD RA_BASE TURRET DRILL GDRT TOOL WATSON SHERLOC PIXL	ROVER_NAV_FRAME	Attached to the tool; see PPPCS for the specific tool frame.	Aligned with tool in some way; see PPPCS [Ref X] for the specific tool Frame.

7.4.1 Rover Navigation (Rover Nav) Frame

The Rover Nav frame (RNAV) is the one used for surface navigation and mobility. By definition, the frame is attached to the rover, and moves with it when the rover moves while on the surface. Its Y origin is centered on the rover and the X origin is aligned with the middle wheels' rotation axis for the deployed rover and suspension system on a flat plane. The Z origin is defined to be at the nominal surface, which is a fixed position with respect to the rover body. The actual surface will likely not be at exactly Z=0 due to the effects of suspension sag, rover tilt, rocker bogie angles, etc. The +X axis points to the front of the rover, +Y to the right side, and +Z down (perpendicular to the chassis deck). See 7-1.

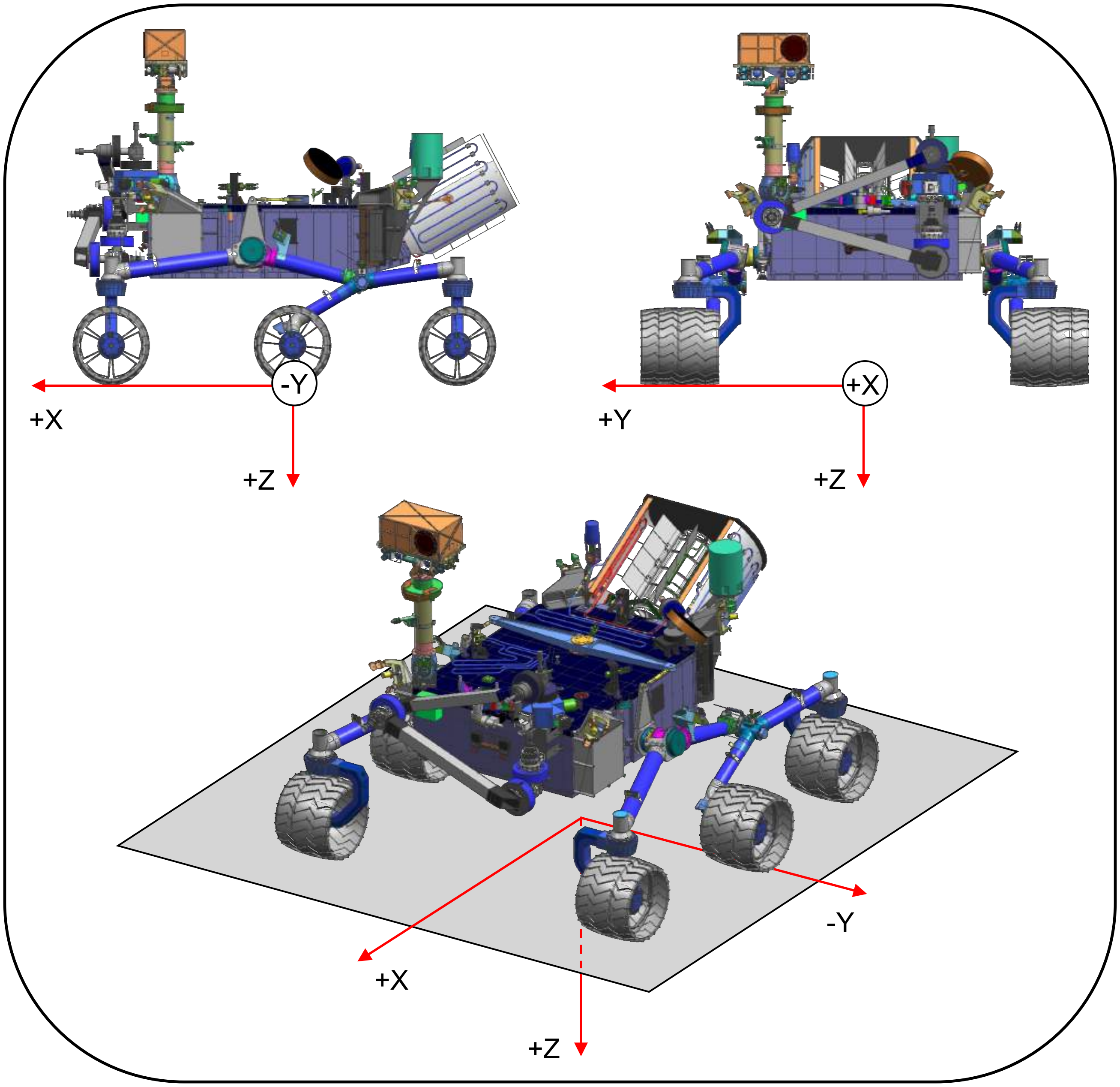


Figure 7-1 – Rover navigation coordinate frame (RNAV)

The Rover Nav frame is specified via an offset from the current Site frame, and a quaternion that represents the rotation between the two. A new instance of the Rover Nav frame, with a potentially unique offset/quaternion, is created every time the ROVER_MOTION_COUNTER increments.

Orientation of the rover (and thus Rover Nav) with respect to Local Level or Site is also sometimes described by Euler angles as shown in Figure 7-2, where ψ is heading or yaw, θ is attitude or pitch, and ϕ is bank or roll.

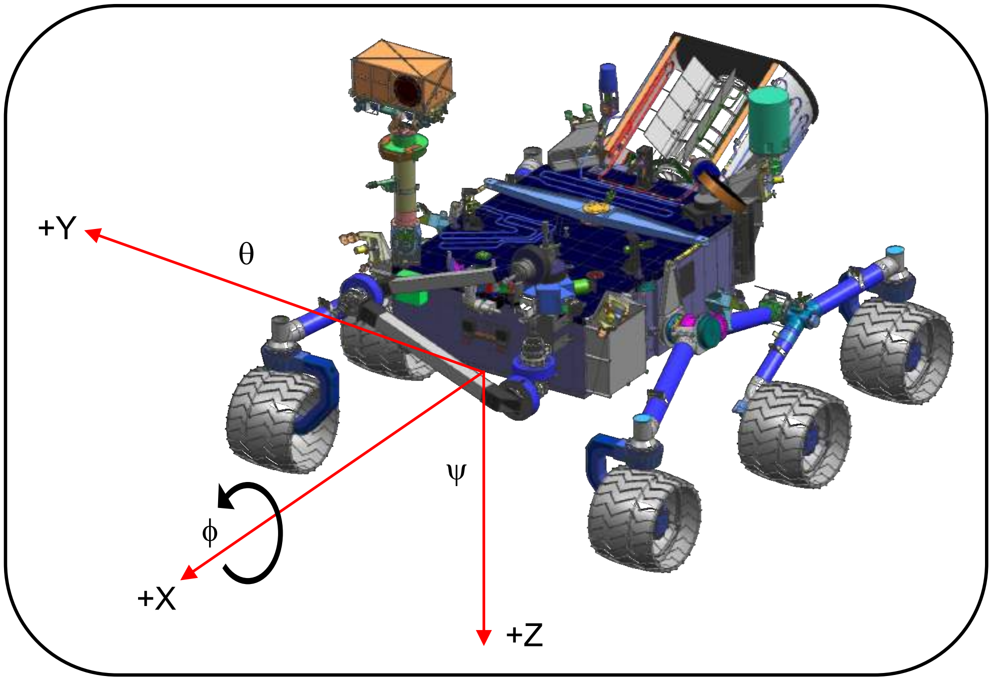


Figure 7-2 – Yaw, Pitch and Roll Definition

7.4.2 Rover Mechanical (Rover Mech) Frame

The Rover Mechanical (RMECH) frame is oriented identically to the Rover Nav frame. The origin is forward of Rover Nav by x=0.09002 meters with a Z origin at the top of the deck (TBC). Given a point expressed in Rover Mech, if you add (0.09002, 0.0, -1.13338) you will get the same point expressed in Rover Nav. Rover Mech is not used by any nominal products (EDR or RDR) but could appear in certain special products, generally having to do with arm kinematics.

7.4.3 Local Level Frame

The Local Level frame is coincident with the Rover Nav frame, i.e. they share the same origin at all times. The orientation is different, however. The +X axis points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system. Thus the orientation matches the orientation of Site frames.

Local Level frames are defined by an offset from the current Site frame, with an identity quaternion.

7.4.4 Site Frame

Site frames are used to reduce accumulation of rover localization error. They are used to provide a common reference point for all operations within a local area. Rover Nav and Local Level frames are specified using an offset from this origin. When a new Site is declared, that becomes the new reference, and the offset is zeroed. In this way, long-term localization error is relegated to the offset between Sites, becoming irrelevant to local operations, because the positions are reset with each new Site.

When a Site frame is declared, it is identical to the Local Level frame, sharing both orientation and position. However, the Site frame is fixed to the Mars surface; when the rover moves, Local Level moves with it but Site stays put. Therefore, for the Site frame, +X points North, +Z points down to nadir along the local gravity vector, and +Y completes the right-handed system.

Sites are indexed, meaning there are multiple instances. Site 1 by definition represents the landing location. New Sites are declared as needed during operations, as the rover moves away from the local area. See Figure 7-3.

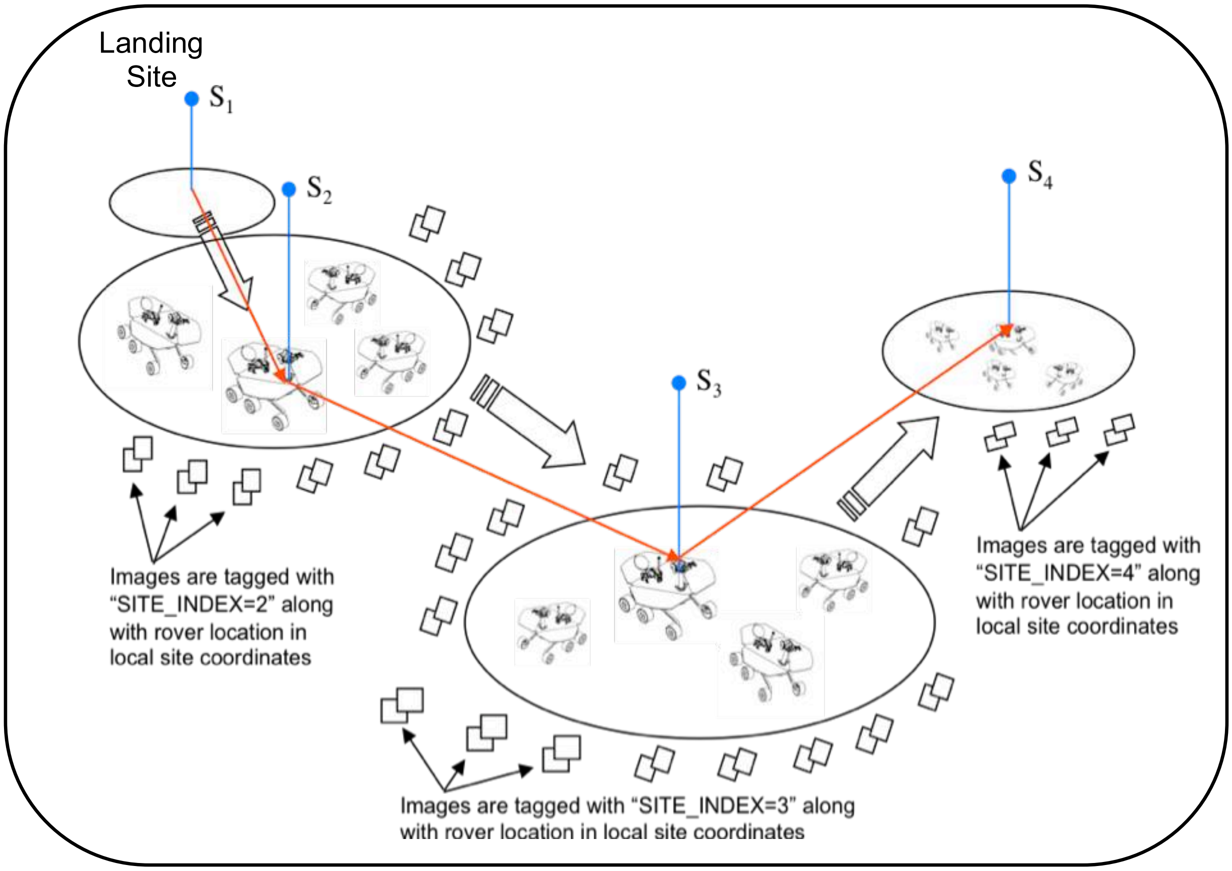

Landing Site
S_1
S_2
S_3
S_4
Images are tagged with "SITE_INDEX=2" along with rover location in local site coordinates
Images are tagged with "SITE_INDEX=4" along with rover location in local site coordinates
Images are tagged with "SITE_INDEX=3" along with rover location in local site coordinates

Figure 7-3 – Site and Rover frame examples

The PLACES database [Ref 10] stores the set of all site-to-site offsets; such offsets are not in every image label.

7.4.5 RSM Frame

The RSM frame is attached to the Remote Sensing Mast (RSM) camera head, and moves with it. See the PPPCS for specific definition. It is expressed as an offset and quaternion from the Rover Nav frame.

7.4.6 Arm Frames

The frame representing the currently selected arm tool is reported in the arm coordinate system group. The selected tool, given by ARTICULATION_DEV_INSTRUMENT_ID, is arbitrary for any given image and may be surprising; for example SHERLOC-WATSON may not be the selected tool for a SHERLOC-WATSON image. The various tool frames are attached to and aligned with the tool in some manner specific to that tool. See the PPPCS [Ref 1] for actual frame definitions.

8. APPLICABLE SOFTWARE

The instrument data downlink processing software is focused on rapid reduction, calibration, and visualization (in the case of images) of products in order to make discoveries, to accurately and expeditiously characterize the geologic environment around the rover, and to provide timely input for operational decisions concerning rover navigation and Arm target selection. Key software tools have been developed at JPL as part of the IDS and RPS subsystems, and at LANL/IRAP/CNES by the SuperCam team. These toolsets can be used to process data to yield substantial scientific potential in addition to their operational importance.

TBD: this really needs to be reworked, more along the lines of MSAM: what tools are available for users to use.

8.1 Utility Programs

Table 7.1 lists (in no particular order) the primary software tools that will be used to process and manipulate downlinked M2020 instrument payload data. At JPL, the "M2020edrgen" program will generate EDRs and the Mars Program Suite of VICAR programs will generate RDRs in PDS format. An IDS pipeline system will deliver the products to the FEI server for transfer to M2020's ODS as rapidly as possible after receipt of telemetry.

TBD: update the VICAR software list. Add Landform, ASTTRO, RSketch, marsviewer, datadrive, ….

Table 8-1 – Key software toolsets

Name	Description	Primary Development Responsibility
SuperCam Ops Software	LIBS software development is primarily performed in IDL code, with LANL (New Mexico) as lead institute.	Dot Delapp (LANL)
	RMI software development is primarily performed in IDL code, with IRAP (France) as lead institute.	Olivier Gasnault (IRAP)
M2020edrgen	Fetches the image Data Product Object (DPO) records from M2020 Data Product (DP) files, reconstructing the image file from the telemetry data into a PDS-labelled image EDR data product. VICAR code.	Alice Stanboli, Hyun Lee (JPL / MIPL)

Name	Description	Primary Development Responsibility
Mars Program Suite	Stereo image processing software using EDRs or calibrated images (RDRs), image mosaicking software, 3-D terrain building software. VICAR code: • MARSINVERTER – Generates inverse lookup table (ILUT) products. • MARSDEBAYER – Generates de-Bayered images. • 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. • MARSECORR – Generates a disparity map from a stereo pair of input EDRs, using on-the-fly epipolar alignment. • MARSCOR3 – Refines a disparity map from a stereo pair of input EDRs, applying a 2-D correlator. • MARSXYZ – Generates an XYZ image from an input disparity map. • MARSRFILT – Range filtering for XYZ images. • MARSRANGE – Generates a range image from an input XYZ map. • M2020REACH – Generates an arm reachability map from an input XYZ map. • M2020ROUGH – Generates roughness maps. • MARSERROR – Generates XYZ and range error maps. • MARSSLOPE – Generates slope maps. • MARSBRT – Generates brightness/contrast correction file for mosaic processing. • M2020FILTER – Generates XML file for image mask files. • MARSFILTER – Generates image mask files. • MARSMASK – Applies image mask files to image files. • MARSDISPCOMPARE – Checks consistency for left-to-right and right-to-left stereo image correlations. • MARSUVW - Generates a surface normal image, wherein XYZ is computed normal to the surface. • 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. • MARSMCAULEY – Generates a combination Cylindrical-Perspective projection mosaic from a list of input EDRs. • MARSTIE – Generates pointing corrections (tiepoint file) from an overlapping set of input EDRs. • MARSNAV – Generates an updated azimuth and elevation file based on comparison with existing image data that can be directly compared. • XVD – De facto image reader software capable of displaying VICAR-labeled image files.	Bob Deen (JPL / MIPL)
	• CRUMBS – 3-D terrain building software	Oleg Pariser (JPL / MIPL)
RPS / RSVP	Visualization, planning, and sequence generation software for use by Sequence Team to create Sol sequences based on activity lists generated by PSI during planning meetings. Java, C and C++ code.	Jeng Yen (JPL)

8.2 Applicable PDS Software Tools

PDS-labeled images and tables can be viewed with the program PDSView, developed by the PDS and available for a variety of computer platforms from the PDS web site https://github.com/NASA-PDS/pds-view . There is no charge for PDSView.

8.3 Software Distribution and Update Procedures

The Mars Image Processing Program Suite is available to researchers and academic institutions. Refer to the MIPL Web site at http://www-mipl.jpl.nasa.gov for contact information.

APPENDIX A. DATA PRODUCT AND DATA PRODUCT OBJECT NAMES

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
Hazcams		**Fullframe, Subframe, Downsampled Images:**	ImgIdph (B)	n/a	n/a
	317	ImgImageFhl (B)			
	318	ImgImageFhr (B)			
	333	ImgImageRhl (B)			
	334	ImgImageRhr (B)			
		Fullframe, Subframe, Downsampled Images (ICER):			
	319	ImgImageIcerFhl (B)			
	320	ImgImageIcerFhr (B)			
	323	ImgImageIcerRhl (B)			
	324	ImgImageIcerRhr (B)			
		Fullframe, Subframe, Downsampled Images (LOCO):			
	325	ImgImageLocoFhl (B)			
	326	ImgImageLocoFhr (B)			
	329	ImgImageLocoRhl (B)			
	330	ImgImageLocoRhr (B)			
		Fullframe, Subframe, Downsampled Images (JPEG):			
	50	ImgImageJpegFhl (S)			
	51	ImgImageJpegFhr (S)			
	52	ImgImageJpegRhl (S)			
	53	ImgImageJpegRhr (S)			
		Thumbnail Images:	ImgIdph (B)	n/a	n/a
	362	ImgThumbFhl (B)			
	363	ImgThumbFhr (B)			
	378	ImgThumbRhl (B)			
	379	ImgThumbRhr (B)			
		Thumbnail Images (ICER):			
	364	ImgThumbIcerFhl (B)			
	365	ImgThumbIcerFhr (B)			
	368	ImgThumbIcerRhl (B)			
	369	ImgThumbIcerRhr (B)			
		Thumbnail Images (LOCO):			
	370	ImgThumbLocoFhl (B)			
	371	ImgThumbLocoFhr (B)			
	374	ImgThumbLocoRhl (B)			
	375	ImgThumbLocoRhr (B)			
		Thumbnail Images (JPEG):			
	57	ImgThumbJpegFhl (S)			
	58	ImgThumbJpegFhr (S)			
	59	ImgThumbJpegRhl (S)			
	60	ImgThumbJpegRhr (S)			
		Histogram Products:	ImgIdph (B)	n/a	n/a
	311	ImgHistogramFhl (B)			
	312	ImgHistogramFhr (B)			
	315	ImgHistogramRhl (B)			
	316	ImgHistogramRhr (B)			
		Row-summed Products:	ImgIdph (B)	n/a	n/a

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	355	ImgRowsumsFhl (B)			
	356	ImgRowsumsFhr (B)			
	359	ImgRowsumsRhl (B)			
	360	ImgRowsumsRhr (B)			
		Column-summed Products:	ImgIdph (B)	n/a	n/a
	303	ImgColsumsFhl (B)			
	304	ImgColsumsFhr (B)			
	307	ImgColsumsRhl (B)			
	308	ImgColsumsRhr (B)			
Navcam		**Fullframe, Subframe, Downsampled Images:**	ImgIdph (B)	n/a	n/a
	331	ImgImageNl (B)			
	332	ImgImageNr (B)			
		Fullframe, Subframe, Downsampled Images (ICER):			
	321	ImgImageIcerNl (B)			
	322	ImgImageIcerNr (B)			
		Fullframe, Subframe, Downsampled Images (LOCO):			
	327	ImgImageLocoNl (B)			
	328	ImgImageLocoNr (B)			
		Fullframe, Subframe, Downsampled Images (JPEG):			
	54	ImgImageJpegNl (S)			
	55	ImgImageJpegNr (S)			
		Thumbnail Images:	ImgIdph (B)	n/a	n/a
	376	ImgThumbNl (B)			
	377	ImgThumbNr (B)			
		Thumbnail Images (ICER):			
	366	ImgThumbIcerNl (B)			
	367	ImgThumbIcerNr (B)			
		Thumbnail Images (LOCO):			
	372	ImgThumbLocoNl (B)			
	373	ImgThumbLocoNr (B)			
		Thumbnail Images (JPEG):			
	47	ImgThumbJpegNl (S)			
	48	ImgThumbJpegNr (S)			
		Histogram Products:	ImgIdph (B)	n/a	n/a
	313	ImgHistogramNl (B)			
	314	ImgHistogramNr (B)			
		Row-summed Products:	ImgIdph (B)	n/a	n/a
	357	ImgRowsumsNl (B)			
	358	ImgRowsumsNr (B)			
		Column-summed Products:	ImgIdph (B)	n/a	n/a
	305	ImgColsumsNl (B)			
	306	ImgColsumsNr (B)			
CacheCam		**Fullframe, Subframe, Downsampled Images:**	ImgIdph (B)	n/a	n/a
	13	ImgImageCc (B)			
		Fullframe, Subframe, Downsampled Images (ICER):			
	14	ImgImageIcerCc (B)			
		Fullframe, Subframe, Downsampled Images (LOCO):			

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	15 56	ImgImageLocoCc (B) **Fullframe, Subframe, Downsampled Images (JPEG):** ImgImageJpegCc (S)			
	16 17 18 49	**Thumbnail Images:** ImgThumbCc (B) **Thumbnail Images (ICER):** ImgThumbIcerCc (B) **Thumbnail Images (LOCO):** ImgThumbLocoCc (B) **Thumbnail Images (JPEG):** ImgThumbJpegCc (S)	ImgIdph (B)	n/a	n/a
	22	**Histogram Products:** ImgHistogramCc (B)	ImgIdph (B)	n/a	n/a
	23	**Row-summed Products:** ImgRowsumsCc (B)	ImgIdph (B)	n/a	n/a
	24	**Column-summed Products:** ImgColsumsCc (B)	ImgIdph (B)	n/a	n/a
SuperCam RMI	426 640 415 642	**Fullframe, Subframe Images:** ScamScidata (C) ScamRmiImage (S) **Fullframe, Subframe Images (ICER):** ScamRmiImageIcer (S) **Fullframe, Subframe Images (JPEG):** ScamRmiImageJpeg (S)	ImfIdphSubset6 (S)	ScamRmiHeader (S) ScamAncillaryTemperatures (S) ScamAncillaryArgs (S) ScamAncillaryParams (S) ScamFocalDistance (S)	n/a
	643 644 645	**Zstack:** ScamRmiZstackImage (S) **Zstack (ICER):** ScamRmiZstackImageIcer (S) **Zstack (JPEG):** ScamRmiZstackImageJpeg (S)	ImfIdphSubset6 (S)	ScamRmiHeader (S) ScamRmiRawMetadata (S) ScamAncillaryTemperatures (S) ScamAncillaryArgs (S) ScamAncillaryParams (S) ScamFocalDistance (S)	n/a
Mastcam-Z	418 419 677 689	**Images:** MczLScidata (C) MczRScidata (C) MczLImage (B) MczRImage(B)	DpoCidph (C) ImfIdphSubset6 (S)	MczImageAncillaryData (C) MmmImageAncillaryData (S)	n/a
	682 694	**Thumbnail Images:** MczLThumbnail (S) MczRThumbnail (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmImageAncillaryData (S)	n/a
	684 696	**Video:** MczLVideo (S) MczRVideo (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmVideoAncillaryData (S)	n/a
	679 691	**Recovered Product:** MczLRecoveredProduct (S) MczRRecoveredProduct (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRecoveredProductAncillaryData (S)	n/a
	680 692	**Recovered Thumbnail:** MczLRecoveredThumbnail (S) MczRRecoveredThumbnail (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRecoveredThumbnailAncillaryData (S)	n/a

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	685 697	**Zstack:** MczLZstack (S) MczRZstack (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmZstackAncillaryData (S)	n/a
	678 690	**Range Map:** MczLRangemap (S) MczRRangemap (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRangeMapAncillaryData (S)	n/a
SHERLOC **WATSON** **ACI**	425 720	**Images:** SrlcImgScidata (C) SrlcImgImage (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmImageAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	721	**Thumbnail Images:** SrlcImgThumbnail (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmImageAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	722	**Video:** SrlcImgVideo (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmVideoAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	723	**Zstack:** SrlcImgZstack (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmZstackAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	724	**Range Map:** SrlcImgRangemap (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRangeMapAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	725	**Recovered Product:** SrlcImgRecoveredProduct (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRecoveredProductAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	726	**Recovered Thumbnail:** SrlcImgRecoveredThumbnail (S)	DpoCidph (C) ImfIdphSubset6 (S)	MmmRecoveredAncillaryData (S)	SrlcImgSupplementaryCmdArguments (S)
	727	**Directory List:** SrlcImgDirectory	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	728	**Debug Info:** SrlcImgDebugInfo	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	729	**Image Status:** SrlcImgStatus	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	730	**BIST Status:** SrlcImgBistStatus	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	731	**Util Test:** SrlcImgUtilTest	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	732	**Memory Dump:** SrlcImgDumpMemory (S)	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
	733	**Zstack List:** SrlcImgZstackList (S)	ImfIdphSubset1 (S)	MmmDefaultAncillaryData (S)	n/a
PIXL	422 35 36	**Images:** PixlSciEng (C) PixlTracking (S) PixlSci (S) PixlSciDec (S)	PixlTracking: ImfIdphSubset4 (S) PixlSci/Dec: ImfIdphSubset1 (S)	n/a	n/a
MEDA **Skycam**	420 43 62 46	**Fullframe, Subframe, Downsamled Images:** MedaSciData (C) MedaUnprocessedImage (S) MedaPartialImage (S) **Fullframe, Subframe, Downsmpled Images (compressed):** MedaCompressedImage (S)	ImfIdphSubset6 (S)	ImgHdr (S)	n/a

Instrument	APID #	APID Names C=Cruise, S=Surface, B=Both	DPOs Used as Sources for Metadata		
			IDPH	Ancillary	Supplementary
	 45	**Thumbnail Images:** MedaThumbnailImage (S)	ImfIdphSubset6 (S)	ImgHdr (S)	n/a
EDL Cam **PUCAM1** **PUCAM2** **PUCAM3** **DDCAM** **RDCAM** **RUCAM**	 416	**Images**: ImgEdlcamScidata (B)	n/a	n/a	n/a
LCAM	 630 631 631	**Images**: VcemgrScidata (B) VcemgrLvsPackets (C) VcemgrVcePackets (S)	n/a	VcemgrMeasurementPacket (C)	n/a
Heli	 417 417	**Images**: HeloScidata (C) LeoSciData (S)	n/a	n/a	n/a

APPENDIX B. PRODUCT LABEL KEYWORD DEFINITIONS, VALUES, AND SOURCES

As described in the main text, there are three types of label keywords: VICAR, ODL, and PDS4. The VICAR and ODL labels are virtually identical and are referred to here collectively as “VICAR” labels.

This Appendix describes several tables that will be useful for understanding the details of these keywords. All of the tables are in separate files within the document collection, with names as described herein.

Operations Keyword Table

This table lists the VICAR keywords as used by the operations team. This is a hand-generated table containing the VICAR keyword and definition, metadata about the valid values, data types, units, and location, and (in the 3rd column) where in the telemetry the data comes from. This last column should not be needed for archive users to understand the data, since the telemetry is not archived, but it is critical for ops users, and may provide hints to archive users if the rest of the information is insufficient for some reason.

Archive users should generally avoid this table in favor of the tables below, since it contains no PDS4 label information, and the tables below should have sufficient information. However, it is being included in case it is of use. The tables can be cross-referenced using the VICAR keyword name.

The filename for this table is:

```
Mars2020_Camera_SIS_Operations_Label.pdf
```

PDS4 Keyword Tables

This set of tables describes the PDS4 keywords (classes and attributes in PDS 4 parlance). They include pointers to the matching VICAR keywords, as well as both the generic (multimission) definition of the keyword, and the specific “nuance” or supplemental information that applies only to Mars 2020.

These tables are created by examining a set of sample labels (incorporating all types of products being created) in order to determine the PDS4 classes and attributes that are actually used by the products described in this SIS. This list is then cross-referenced against the PDS4 data dictionaries in order to find the definitions, children, valid values, and data types. This list is then augmented with “property maps” that provide the Mars 2020-specific valid values, and the “nuance” definitions.

These tables are thus much more useful for most purposes than looking at the PDS4 data dictionaries directly, because they contain *only* the keywords that are *actually* used.

The first column contains the name of the PDS4 attribute (keyword) or class (container), and the PDS4 dictionary it comes from. Along with that, when applicable, are the VICAR keyword and property name(s) from which the values are derived. The property name is the section of the VICAR label. Not every entry has a VICAR keyword; some entries are merely containers, others contain constants or values that are derived in other ways. Some of the VICAR keywords refer to the class rather than the attribute; for example a VICAR vector keyword will typically refer to the vector's class rather than the x,y,z attributes individually.

The second column contains the definition. There are two components to many definitions, as alluded to above. The first, which is always present, is the standard PDS4 definition that applies to all missions, from the PDS4 data dictionary. The second (in italics) is a Mars 2020-specific nuance to the definition, providing additional context that applies specifically to Mars 2020.

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 items' definition.

Underneath the XPath is a field whose content varies based on the type. For attributes ("keyword"), this contains the valid values, when such are defined either by the PDS4 data dictionary or the Mars 2020-specific property maps. For classes (containers), the valid children are listed. Those that are blue hyperlinks are actually used by Mars 2020; clicking on them will go to that item's definition. Those that are not blue are defined by the PDS4 data dictionary but are not used by Mars 2020.

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 some have units defined.

These label tables are the primary source of information regarding the metadata in the labels. The rest of this document describes things at a high level; the label tables (along with the ops label table, above) define specifically what each label item means.

There are two versions of the table: sorted by PDS4 name, and sorted by VICAR keyword. The tables can be used in either direction. Given a label item you don't understand, you can look it up in the table (sorted by either 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 the item 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 image products, browse images, mosaics, meshes, calibration, and documentation files.

The cross-reference between PDS4 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).

Each of the tables is provided in both HTML and PDF format. The files are:

```
Mars2020_Camera_SIS_Labels_sort_pds.pdf
Mars2020_Camera_SIS_Labels_sort_pds.html
Mars2020_Camera_SIS_Labels_sort_vicar.pdf
Mars2020_Camera_SIS_Labels_sort_vicar.html
```

APPENDIX C. CALIBRATION PARAMETERS

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 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 few 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 four directories. An overview of each type of file is described below.

Many of the filenames start with a "venue". For Mars 2020 this venue is "`M20`", for the flight hardware. Other testbeds, e.g. "`M20VSTB`", exist for various testbeds at JPL. Cal data files specific to the testbeds are not included in the delivery, but files that contain information for both testbed and flight retain the testbed values. 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 Linux 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 InSight. 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

`M20_SN_xxxx.cahv{or}{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.

M20_SN_xxxx.interp{t,f,z}: These files are the camera model interpolation files for temperature (Navcam) or zoom, focus, and temperature (Mastcam-Z). They contain the zoom, focus, or temperature value and the specific camera model filename for that focus or temperature point.

Other *.cahv{or}{e}: These are "original" camera model files (see Appendix introduction). The CAL_nnn allows for multiple recalibrations.

flat_fields Directory

M20_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. These are "softlink" files (see Appendix introduction). The files with "RAWBAYER" in their names have the Bayer pattern encoded in them; they should be used for Bayer-pattern images with the non-"RAWBAYER" files used for color images.

Others: Other .IMG or .VIC files in this directory are the "original" flat field files (see Appendix introduction).

ilut Directory

M20_MMM_LUT*n*.txt: These are the inverse lookup table (ilut) files needed to create the decompanded ILT products (Section **Error! Reference source not found.**) for the Mastcam-Z and SHERLOC cameras. The format is simple: each line has "input_DN output_DN" which converts from 0-255 to 0-4095.

M20_LUT*n*.txt: These are the inverse lookup table files for the engineering cameras.

param_files Directory

MSL_camera_mapping.xmlx: This file contains information for each camera on Mars 2020. 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. This file needs to be renamed with a ".xml" extension if they are to be used with the Mars software suite.

M20_*hardware*.point: This file contains pointing parameters for the mast- or arm- mounted cameras. The calibration position and quaternion define the pose of the mast or 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 or mast kinematics for a given image. The pointing error values are used by the pointing correction mechanism as the initial perturbation for each parameter. Most of the rest of the values are parameters for the mast kinematics (see [Bell2017]).

M20_flat_fields.parms: This file defines the radiometric 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; the units are thus W/m^2/nm/steradian/(DN/sec).

`M20_rover_filter.xmlx`: This file defines the mask used to filter all images. It contains only a description for a horizon mask with an elevation of 90 degrees, which effectively disables it.

`M20_filter_exclude.txt`: This file defines a list of “objects” to exclude when making a rover mask.