The Compendium of Controlled Extensions (CE)

for the

National Imagery Transmission Format (NITF)

APPENDIX Z

General Electro-Optical (Visible, Infrared, Multi- and Hyperspectral) Sensor Parameters (SENSRB) Tagged Record Extension (TRE)

VERSION 2.1

December 2013

(This page intentionally left blank.)

CHANGE LOG

Date	***Pages Affected***	***Mechanism***
29 December 2010	All	Initial Publication of STDI-0002, Appendix Z: SENSRB
4 December 2013	All	Document update, completion of all TBDs, addl figures

TBR (TO BE RESOLVED)

Date	***Pages Affected***	***Issue***
TBR01	Recommended Uniform Implementation	Determine whether the recommended uniform implementation for coordinates should be geocentric or geodetic

FOREWORD

1. The National Imagery Transmission Format Standard (NITFS) is the suite of standards for formatting digital imagery and imagery-related products and exchanging them among the Department of Defense (DOD), other Intelligence Community (IC) members, and other United States (US) Government departments and agencies. Resulting from a collaborative US Government and industry effort, it is the common standard used to exchange and store files composed of images, symbols/graphics, text, and associated data.

2. This appendix to the Controlled Extension (CE) compendium provides one of the approved CE specifications to be used with the National Imagery Transmission Format (NITF) versions 2.0 (NITF2.0) or 2.1 (NITF2.1). Support Data Extension (SDE) implementation criteria are defined in N0105-98, NITFS Standards Compliance and Interoperability Test and Evaluation Program Plan, 25 August 1998.

3. The NITFS Technical Board (NTB) develops, coordinates, reviews, and plans for the NITFS. It is a consensus-based government/industry forum that responds to the Geospatial Intelligence Standards Working Group (GWG). The GWG manages geospatial and imagery standards for the DOD and IC encompassed by the National System for Geospatial-Intelligence (NSG).

4. Changes to this appendix are controlled via the GWG, the NTB and the National Geospatial-Intelligence Agency (NGA) Enterprise Standards Working Group (ESWG). Comments, suggestions, or questions should be addressed to the NGA National Center for Geospatial Intelligence Standards (NCGIS), Mail Stop P-106, 12310 Sunrise Valley Drive, Reston, VA 20191-3449, or emailed to ncgismail@nga.mil. Since contact information can change, you may want to verify the currency of this address information using the ASSIST Online database at www.dodssp.daps.mil.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Z.1. INTRODUCTION

SENSRB is a National Imagery Transmission Format Standards (NITFS) tagged record extension (TRE) for imaging electro-optical (EO) sensors. These sensors may be passive or active; measuring and recording light—including infrared (IR), visible, or ultraviolet (UV)—either as a single broad panchromatic band or in multiple bands—as with multispectral, hyperspectral, or polarimetric sensors. This controlled extension is required for airborne EO sensors (see *STDI-0002, Appendix E: ASDE 2.1*) and can also be used for other EO sensors—such as ground-based, maritime,[1] or spaceborne. The TRE is intended for applications where the earth's surface is imaged; nevertheless, it may also accommodate the imaging of features or phenomena above or below the earth's surface.

SENSRB updates SENSRA with expanded system descriptions, improved sensor attitude definitions, and increased precision levels. This extension also now allows for sensor calibration parameters, time stamping, pixel referencing, and uncertainty estimates,[2] which were not accommodated by SENSRA, but are required for various implementations and applications.

Through the course of collecting an image, some of the sensor parameter values captured in a SENSR TRE may be changing. Using SENSRB, such dynamic values can be recorded in looping fields within a single TRE (see Z.4.8 Looping Fields Concept). Alternatively, the values can be recorded in multiple SENSRB TREs. In either case, all SENSRB TREs are stored within a NITF file's image subheaders and/or TRE-overflow data extension segment as described in *MIL-STD-2500C*, section 5.8. (See also Z.4.9 General SENSRB Usage.)

The development of SENSRB was carried out with consultation between industry, academic, and government experts. It was developed to facilitate the storage and subsequent use of geometric parameters from a wide variety of EO sensor systems. It serves as an encoding scheme for the abstract formulations contained in the Community Sensor Model Working Group's implementation guidance documents for geopositioning from frame, pushbroom, and whiskbroom sensors. In addition, the extension was designed to characterize still images derived from motion imagery sources implementing the Motion Imagery Standards Board's *UAS Datalink Local Data Set (MISB Standard 0601)* and *Engineering Guideline—Profile 1: Photogrammetry Metadata Set for Digital Motion Imagery (MISB EG 0801)*. The relationships between SENSRB and these and other documents are, at the time of this publication, under review and are intended to be described in a future version of *STDI-0005: Implementation Practices of the NITFS (IPON)*.

Because SENSRB was developed to address several immediate needs and was designed for utilization by a variety of existing and potential users, a variety of valid methods for providing data are accommodated by this TRE. Nonetheless, new implementations of SENSRB should be designed toward a narrower standard using a smaller, defined set of optional implementation methods. Guidelines for this more uniform implementation are provided in Z.7 Recommended Uniform Implementation, at the end of this appendix. Existing systems are also encouraged to migrate toward these uniform implementation recommendations, as they are able to do so.

A companion software package, ***SENSRB Data Conversion Tool***, has been developed for distribution with this document. This tool provides a working reference for the interpretation of the SENSRB specification. It allows the user to create sample SENSRB TREs using interactive input forms and permits the user to read and verify existing SENSRB TREs. Visualization options allow users to confirm expected results. The tool also converts SENSRB data between various standardized structures and formats. A current version of the *SENSRB Data Conversion Tool* and user documentation is available for download at the NITF Technical Board's demonstration software website, http://www.gwg.nga.mil/ntb/baseline/software/demo.html. See Z.6.4 SENSRB Data Conversion Tool.

[1] Although SENSRB might appropriately be used with underwater sensors, such applications have not yet been sufficiently explored to recommend this usage.

[2] Many of the parameter values stored in this TRE are estimates based on instrument readings or assumed settings of the actual value. These values, therefore, are always subject to uncertainties, which should be provided.

This appendix is organized with the following sections:

1. Introduction
2. Structure and Format Overview
3. Field Specifications
4. General Technical Concepts
5. Module-Specific Implementation Notes
6. Advanced Technical Concepts
7. Recommended Uniform Implementation
8. Acronyms, Symbols, and Glossary

Z.2. STRUCTURE AND FORMAT OVERVIEW

This section provides an overview of the SENSRB structure and format, sufficient to prepare the reader to use the specification table in Z.3 Field Specifications. Additional technical guidance for SENSRB users is provided in the sections following the specification. The *SENSRB Data Conversion Tool* (see Z.6.4 SENSRB Data Conversion Tool) provides an interactive reference to further aid in the interpretation.

Z.2.1. DATA MODULES

A SENSRB TRE makes available fifteen data "modules" as illustrated below in Figure Z.2-1. Data modules are bundles of associated data fields; data fields are byte allotments for prescribed field values. Two data modules are mandatory for any and all instantiations of SENSRB. The others are conditional or optional depending on prescribed requirements. Conditional and optional modules contain a mandatory flag field indicating the presence, and sometimes quantity, of the associated data fields. Either a "Y" or a non-zero value in a flag field indicates that the module's data fields are present (the module "exists"). Conversely, if a module's flag field is populated with an "N" or a zero value, no data fields are present for that module (the module "does not exist"). The intention is to allow the SENSRB implementer to tailor a metadata population plan to the sensor system and imagery product being characterized by this TRE.

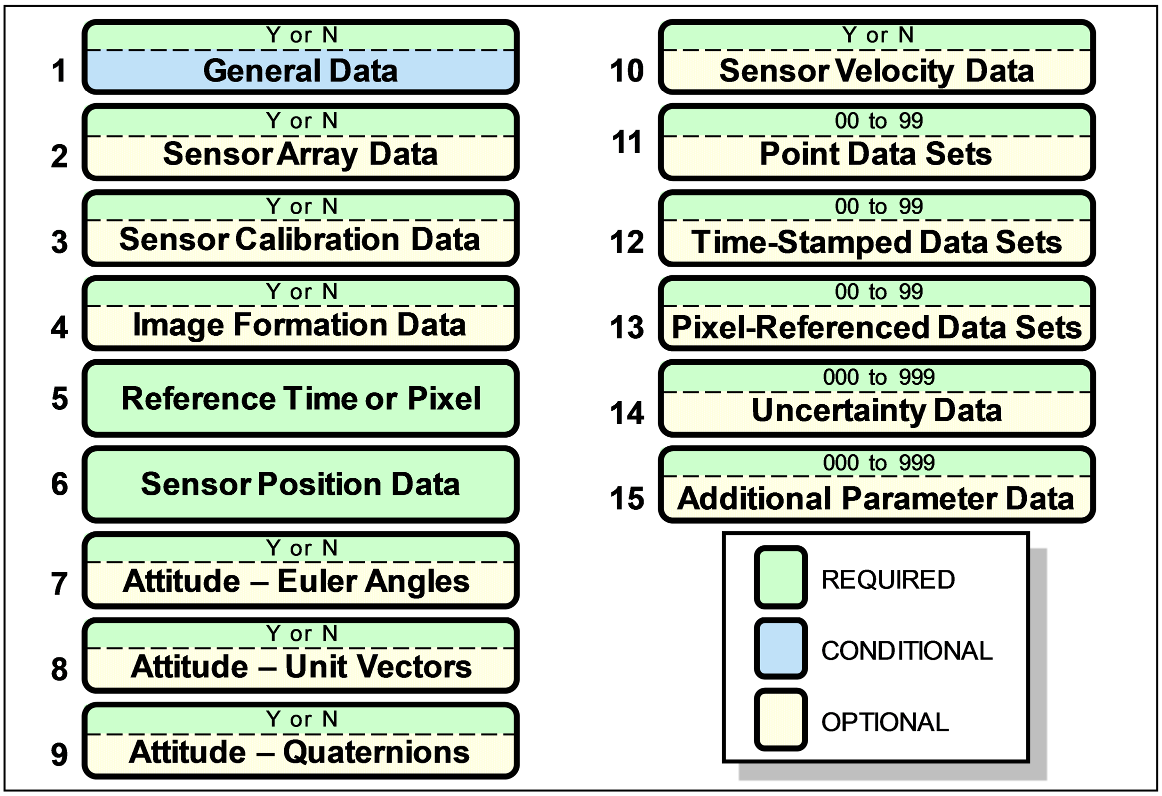


Figure Z.2-1. Schematic View of SENSRB Modules. Labels such as "Y or N" and "00 to 99" denote required flag fields, which indicate the existence of subsequent data fields in conditional or optional modules.

The presence of data fields within a module (the module's "existence") depends on the data's availability and applicability but is also determined by one of three hierarchical requirements:

- Mandatory existence (required)—modules have no flag field and their data fields are required for each and every SENSRB instantiation. (Modules 5 and 6).
- Specification-required existence (conditional)—module's data fields are required only for the first (or sole) SENSRB instantiation associated with each image segment. (Module 1).
- Application-required existence (optional)—modules' data fields must exist to facilitate certain exploitation applications. (Modules 2 to 4 and 7 to 15). (See Z.5.1.2 Application-Required Content Level for examples.)

Z.2.2. DATA FIELDS

The SENSRB data fields are specified in Z.3 Field Specifications. The following paragraphs define the SENSRB data field types and explain how the fields are referenced, defined, and formatted.

Z.2.2.1. Field Types and Unspecified Values

There are three types of data fields specified in Table Z.3-1 SENSRB Data Fields Specifications (see TYPE column at far right of the table):

- **R**equired,
- **C**onditional, and
- **L**ooping-Conditional.

Each is described here:

- Required Fields shall always be present in every SENSRB instantiation. These mandatory fields consist mostly of flag fields for the conditional and optional modules and fields contained within the required modules. SENSRB specifies a total of 24 required fields, including the NITF fields known as CETAG and CEL.
- Conditional Fields are fields whose presences are determined by "Y" or "N" flag fields or other application-specific conditions. These are mostly data fields contained within Modules 1 to 4 and 7 to 10.
- Looping-Conditional Fields are fields that can have multiple occurrences as determined by numeric flags or count fields. These conditional fields are found in Modules 11 to 15. The use of these fields requires diligence by both the SENSRB implementer and exploiter. Additional guidance regarding the use of looping fields can be found in Z.4.8 Looping Fields Concept and in the individual modules' implementation notes (Sections Z.5.11 to Z.5.15).

Unspecified. The SENSRB specification allows that certain fields from each of the three field types be given a value "unspecified" when meaningful data is not available or when the field is not applicable. These fields are indicated with square brackets around the type indicator ([R], [C], or [L]). ***Basic Character Set (BCS) hyphens (minus signs) (code 0x2D) will be used as the Unspecified Indicator for all such fields.*** For the SENSRB TRE, this indicator definition takes precedence over the standard definition provided in *MIL-STD-2500C*, paragraph 5.1.7.c. All zeros (0's), all nines (9's), all BCS spaces (blanks), or any potentially meaningful values are *not* valid indicators.

In the SENSRB Data Conversion Tool, these unspecified fields are indicated in the interactive module forms using the symbol 'N/A' after a TRE has been created or read in. The appropriate unspecified indicator (all minus signs) is output when the tool writes out a TRE.

Z.2.2.2. Field References

SENSRB data fields can be referenced by indices and by short names. Each is indicated in Table Z.3-1 SENSRB Data Fields Specifications (see INDEX and FIELD NAME columns at far left of the table).

Indices are utilized to allow select fields to reference other fields more efficiently for such purposes as time stamping, pixel referencing, and uncertainty reporting. Indices are made up of the data module number (two digits) and—without separation—a single lower-case letter taken sequentially from the alphabet. Thus they have the format *MMa*. (The index for each module's required flag is an exception, as it is only the two-digit module number without the lower-case letter.)

In the case of looping fields, additional integer counters are appended to the index to distinguish between multiple instantiations of a given field (see Z.4.8.2 Looping Field Indexing).

Short Names are the proposed text strings associated with each data field when SENSRB is parsed for display. These are given in the FIELD NAME column of Table Z.3-1.

Z.2.2.3. Field Definitions

The DESCRIPTION and UNITS columns in Table Z.3-1 SENSRB Data Fields Specifications help define each field and its value.

The DESCRIPTION column typically includes a long field name and a definition of the field value. Several definitions include brief implementation guidance to further clarify the value's meaning. Some definitions include references to subsequent sections where the data field is further defined—often using equations and illustrations. This additional defining information may be critical to the interpretation of the field.

The UNITS column indicates what units are used for the field values. If multiple unit options are available in the table, the choice of units is established by values set in other specific fields.[1] For some looping fields, the units are defined with table footnotes. For unitless values, the units are either denoted with a hyphen (-) or a description of the kind of value allowed in the field. The kinds of values that are noted in the table's UNITS column include:

- Flag—a value indicating a yes or no ('Y' or 'N'),
- Date—a standard numerical year, month, and day value (see Z.4.2 Times and Dates),
- Count—a numerical value indicating how many events have occurred (generation count) or how many fields of a certain type follow (looping fields),
- Index—an appropriate field-referencing index (see Z.2.2.2 Field References above).

Z.2.2.4. Field Formats

Table Z.3-1 SENSRB Data Fields Specifications specifies the byte allotment and Basic Character Set (BCS) requirements for each data field. These are presented in the table's BYTE SIZE and CHAR SET columns, respectively. Each field value, if present, must occupy the fixed byte size (number of characters) indicated in the table using only characters from the assigned set. The four character sets allowed in SENSRB are:

- BCS-Alphanumeric (BCS-A)
- BCS-Numeric (BCS-N)
- BCS-Numeric Integer (BCS-NI),
- BCS-Numeric Positive Integer (BCS-NPI)

These are each defined in *MIL-STD-2500C*, paragraph 5.1.7.a. The general use of these character sets in NITF is outlined in *MIL-STD-2500C*, paragraph 5.1.7.b; but the following paragraph provides the guidance and clarification to be used in the SENSRB TRE. Further critical guidance with regard to the representation of numerical values is provided in Z.4.3 Numerical Expressions.

In the case where the field value would not populate the entire field size allotment, the following guidance shall be followed, depending on whether the data is alphanumeric or numerical and based on the character set assigned to the field.

[1] These fields are GEODETIC_TYPE (index 01h), LENGTH_UNIT (index 01j), ANGULAR_UNIT (index 01k), or CALIBRATION_UNIT (index 03a).

- Alphanumeric Data. This data can only be represented when the BCS-A character set is assigned. In this case, the field value will be left justified, padded on the right with trailing BCS spaces (code 0x20).
- Numeric Data. Numeric data can be provided using any of the four character sets.[1] In every case, numeric data shall be right justified with padding on the left. If allowed[2] and if used, a plus (+) or minus (-) sign will immediately precede the first numeric digit (including a leading zero used for padding).
  - If BCS-A is assigned to the field, the left padding can include BCS spaces (code 0x20) and/or zeros. If used, spaces must be used only as the leftmost characters—preceding any leading zeros (and sign, if used).
  - The numeric character sets (BCS-N, BCS-NI, and BCS-NPI) do not include the BCS space; so for these cases, padding can only be done using leading zeros (preceded with a sign, if used). Of course, trailing zeros could also be used but as that implies greater precision than given by the data, this is discouraged.

  Additional guidance for the appropriate representation of numerical values is given in Z.4.3 Numerical Expressions.

Z.2.2.5. Field Value Ranges

The VALUE RANGE column in Table Z.3-1 SENSRB Data Fields Specifications identifies the range of permitted field values. These are either a listing of the allowed text options, the allowed index range, the allowed numerical range, or other instructions.

For numerical values, VALUE RANGE provides the field value limits as described below.

- "any" any number, positive or negative or zero that will fit in the field.
- "any positive" any positive number, including zero, that will fit.
- "any > 0" any positive number, not including zero.
- "0 to value" any positive number from 0 to value inclusive[3]
- "0 to ±value" any number from -value to +value including zero[3]

If the BCS-A character set is used for the field, scientific notation is allowed. The proper format and interpretation of numerical values are further described in Z.4.3 Numerical Expressions.

The alternative value range instructions—"See NTB registry for approved values", "User defined", and "See description"—are explained here:

- See NTB registry for approved values. The value placed in the field must first be registered with the NITFS Technical Board (NTB). See Z.4.1 NTB-Registered Field Values for instructions on reviewing and adding to the NTB registry.
- User defined. Any appropriate combination of allowed characters may be used without the requirement to register the value.
- See description. The range is adequately and most appropriately defined in the DESCRIPTION column.

---

[1] The potential use of IEEE 754 floating-point representations of numerical values was investigated as part of the SENSRB development. At the present time, the benefit of reduced byte size for a given precision was determined to not outweigh the benefit of BCS readability.

[2] The plus and minus signs ('+' and '-') are not included in the BCS-NPI character set. As a consequence, the BCS-NPI character set cannot be assigned to a field allowing an unspecified value (all hyphens). Instead, the field uses the BCS-NI, and the valid numerical range is defined to restrict the value to the appropriate positive values.

[3] When the value range is limited by an integer value (such as "0 to ±90"), this range still allows for non-integer values (such as, "-33.26" or "000.5"), if supported by the field's assigned character set (see Z.2.2.4 Field Formats).

Z.2.3. Summary of Formats and a TRE Sample

Z.2.3.1. Field Format Summary

Table Z.2-1 shows an example of the format of each of the basic data field values. Additional details are provided in Z.4.2 Times and Dates and Z.4.3 Numerical Expressions.

Table Z.2-1 Field Format Summary

Representation	CHAR SET	Example Index	Sample Value
Alphanumeric	BCS-A	01a	ACESHY
Integer	BCS-NPI	02b	00000768
Floating Decimal	BCS-N	02f	000005.5
Exponential	BCA-A	03f	-002.7974e-9
Date	BCS-NI	01l	19970405
Time	BCS-N	05a	00000487.215

Z.2.3.2. A Sample SENSRB TRE

Table Z.2-2 below represents a simple yet complete sample of a valid TRE. There are no separators or line feeds in a TRE so the following would properly be shown as one very long string on a single line.

Table Z.2-2 Sample SENSRB TRE

```
SENSRB00445YACESHY
MQ-1 Predator                                                   Airborne
0WGS84GHAESIDEG1997040500000000000000199704050000000000000000----------
--------YVisible              00000768000010240000000200000003000003.5--
--------------NNN00000487.215----------------+00038.8845-
000077.03330003600.7780000000000000000000000000Y1-0017.0165-
18.98002000000000N----------------------------NNN000000000000
```

Z.2.3.3. Another Representation of this same TRE

Table Z.2-3 is one of a number of output formats that can be generated from the SENSRB Data Conversion Tool (see Z.6.4 SENSRB Data Conversion Tool). This tabular format contains the same data (in the third column) but is clearly more human readable. The first column shows the TRE number (1.) followed by the index code (such as 01e). The data begins after the TRE name, SENSRB, and the length of this data, 00445 bytes (characters).

Table Z.2-3 Sample SENSRB Name Table

Index	Field Name	Value
1.01	General Data	Y
1.01a	Sensor	ACESHY
1.01b	Sensor URI	
1.01c	Platform	Super Hornet
1.01d	Platform URI	
1.01e	Operational Domain	Airborne
1.01f	Content Level	0
1.01g	Geodetic System	WGS84
1.01h	Geodetic Type	G
1.01i	Elevation Datum	HAE
1.01j	Length Unit	SI
1.01k	Angular Unit	DEG
1.01l	Start Date	19970405

```
1.01m   Start Time                 00000000000000
1.01n   End Date                   19970405
1.01o   End Time                   00000000000000
1.01p   Generation Count           00
1.01q   Generation Date            --------
1.01r   Generation Time            ----------
1.02    Sensor Array Data          Y
1.02a   Detection                  Visible
1.02b   Row Detectors              00000768
1.02c   Column Detectors           00001024
1.02d   Row Metric                 00000002
1.02e   Column Metric              00000003
1.02f   Focal Length               000003.5
1.02g   Row Fov                    --------
1.02h   Column Fov                 --------
1.02i   Calibrated                 N
1.03    Sensor Calibration Data    N
1.04    Image Formation Data       N
1.05a   Reference Time             00000487.215
1.05b   Reference Row              --------
1.05c   Reference Column           --------
1.06a   Latitude                   000038.8845
1.06b   Longitude                  -000077.0333
1.06c   Altitude                   0003600.778
1.06d   Sensor X Offset            00000000
1.06e   Sensor Y Offset            00000000
1.06f   Sensor Z Offset            00000000
1.07    Attitude Euler Angles      Y
1.07a   Sensor Angle Model         1
1.07b   Sensor Angle 1             -0017.0165
1.07c   Sensor Angle 2             -18.98002
1.07d   Sensor Angle 3             0000000000
1.07e   Platform Relative          N
1.07f   Platform Heading           ---------
1.07g   Platform Pitch             ---------
1.07h   Platform Roll              ----------
1.08    Attitude Unit Vectors      N
1.09    Attitude Quaternions       N
1.10    Sensor Velocity            N
1.11    Point Sets                 00
1.12    Time Stamps Sets           00
1.13    Pixel Reference Sets       00
1.14    Uncertainties              000
1.15    Additional Parameters      000
```

Additional output formats can be created using the *SENSRB Data Conversion Tool* (see Z.6.4 SENSRB Data Conversion Tool) including an “Index Table” containing only the first and third columns of the “Name Table” shown above. One can also create an “XML Table” output. In addition, the system is also able to create interactive graphic Google Earth representations of the geometric data contained in the TRE (see section Z.6.4.).

The *SENSRB Data Conversion Tool* serves as a software reference to the SENSRB Specification. As such, it is highly recommended that SENSRB generators and exploiters use this tool to ensure proper implementation of the specification.

Z.3. FIELD SPECIFICATIONS

Table Z.3-1, on the following pages, provides the specifications for the SENSRB data fields. The purpose and use of the table columns are discussed above in Z.2.2 Data Fields. Technical notes regarding the specifications in general are provided in Z.4 General Technical Concepts, whereas guidance specific to individual modules is given in Z.5 Module-Specific Implementation Notes. Additional technical clarification and guidance is contained in Z.6 Advanced Technical Concepts and Z.7 Recommend Uniform Implementation.

Table Z.3-1. SENSRB Data Fields Specifications.

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
	CETAG	Controlled Extension Name. Unique tagged record extension type identifier	-	SENSRB	6	BCS-A	R
	CEL	Total Length of CEDATA. Number of bytes of data present for fields 01 to 15d. That is the length of this controlled extension minus the eleven bytes allotted for CETAG (6 bytes) and CEL (5 bytes); refer to *MIL-STD-2500C*, section 5.8.1 for guidance.	bytes	106 *to* 99985	5	BCS-NPI	R
01	GENERAL_DATA	General Data Flag. Flag field indicating the presence of general data. This module provides identifying information for the associated NITF image segment and sets reference systems for subsequent module parameters. 'Y' in this field indicates the presence of the Fields 01a to 01r; 'N' omits their presence. The value of this field must be 'Y' for the first instance of SENSRB associated with each NITF image segment. **See Z.5.1 General Data Module for additional guidance regarding these fields**.	*flag*	Y *or* N	1	BCS-A	R
01a	SENSOR	Sensor Registered Name or Model. Identifies the common name for the payload sensor that collected the image segment. The twenty-five character name must be unique, explicit, and registered with the NTB.[c] For sensor payloads made up of multiple of sensors, SENSOR_URI (index 01b), DETECTION (index 02a), METHOD (04a), and MODE (04b) fields allow for further characterization of the imagery source.	-	*see NTB registry for approved values.*[c]	25	BCS-A	C
01b	SENSOR_URI	Sensor Uniform Resource Identifier. This optional field allows a unique identifier specific to the collecting sensor. This field supports serial numbers or, preferably, Uniform Resource Identifiers (URI)—which can facilitate access to more detailed sensor information, such as geometric or radiometric calibration data, via the internet. The 32-character field supports Internet Protocol v6. See Z.5.1.1 Sensor and Platform Identification and Characterization.	-	*user defined*	32	BCS-A	[C]
01c	PLATFORM	Platform Common Name. Identifies the platform type upon which the sensor is operating. The twenty-five character name must be unique, explicit, and registered with the NTB.[c]	-	*see NTB registry for approved values.*[c]	25	BCS-A	C
01d	PLATFORM_URI	Platform Uniform Resource Identifier. This optional field allows a unique platform identification, especially for non-aircraft platforms where ACFTB TRE AC_TAIL_NO does not apply. The thirty-two byte field accommodates a Uniform Resource Identifiers (URI), which can facilitate access to more detailed platform information via Internet Protocol v6. See Z.5.1.1 Sensor and Platform Identification and Characterization.	-	*user defined*	32	BCS-A	[C]
01e	OPERATION_ DOMAIN	Operational Domain. Specifies the sensor platform's domain of operation during the collection—providing some indication of the imagery perspective. Currently approved values are:[c] **Airborne** **Waterborne** **Spaceborne** **Ground**	-	*see NTB registry for approved values.*[c]	10	BCS-A	C
01f	CONTENT_LEVEL	Content Level. Quantifies the level of SENSRB data content to enhance data discovery; see Z.5.1.2 Application-Required Content Level for examples. This single value allows users to imply and/or infer if imagery will meet certain exploitation requirements and discover data with a specific content level.	-	0 *to* 9	1	BCS-NPI	C

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.

This table continues on the next page.

Table Z.3-1 (contd.). SENSRB Data Fields Specifications. (Module 1—contd.) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
01g	GEODETIC_SYSTEM	Geodetic Reference System. Specifies the geodetic system to which the geocoordinates in this TRE are referenced. The default is **WGS84** for the World Geodetic System—1984.[c] (See Z.4.6.1 Geospatial Coordinate Systems.)	-	*see NTB registry for approved values.*[c]	5	BCS-A	C
01h	GEODETIC_TYPE	Geodetic Coordinate Type. Specifies the coordinate system used to report the sensor location and the reference system for attitudes and velocities. The two allowed field values are: G (Geographic/Geodetic) and C (geocentric Cartesian); see also Z.4.6.1 Geospatial Coordinate Systems. The local geographic coordinate frame shall be North-East-Down (NED).	-	G *or* **C**	1	BCS-A	C
01i	ELEVATION_DATUM	Elevation and Altitude Datum. Specifies the reference datum from which elevations and altitudes will be reported. The three allowed field values are: HAE (height above ellipsoid), MSL (height above mean sea level), and AGL (height above ground level). **HAE** is *strongly* encouraged; see Z.4.6.1 Geospatial Coordinate Systems.	-	**HAE**, MSL, *or* AGL	3	BCS-A	C
01j	LENGTH_UNIT	Length Unit System. Specifies the unit system used for the spatial parameters within this TRE. The two allowed field values are: SI (International System of Units) and EE (English Engineering Unit System). **SI** is encouraged. (See Z.4.4.1 Linear Units.)	-	**SI** *or* EE	2	BCS-A	C
01k	ANGULAR_UNIT	Angle Unit Type. Specifies the angular units used for the angular parameters within this TRE (unless explicitly overridden by the UNITS column). The three allowed field values are: DEG (degrees), RAD (radians), or SMC (semi-circles). **DEG** is encouraged. (See Z.4.4.2 Angular Units.)	-	**DEG**, RAD, *or* SMC	3	BCS-A	C
01l	START_DATE	Imaging Start Date. Date at the start of the NITF image segment collection, formatted as YYYYMMDD and referenced to UTC. (See Z.4.2 Times and Dates.)	*date*	*see description*	8	BCS-NI	C
01m	START_TIME	Imaging Start Time. The number of UTC seconds into the day, specified in START_DATE (index 01l), when the first photon contacted the detector for the first collected pixel stored in the NITF image segment. The day starts at the UTC zero seconds (0.000000000 s) and ends just before the start of the next day (86399.99999999 s). The value may be equal to END_TIME for imagery sensors modeled as instantaneous collectors. (See Z.4.2 Times and Dates.)	s	*see description*	14	BCS-N	C
01n	END_DATE	Imaging End Date. Date at the end of the NITF image segment collection, formatted as YYYYMMDD and referenced to UTC. Must be the same as the imaging start date (or after—for imagery collections extending into a subsequent day). (See Z.4.2 Times and Dates.)	*date*	*see description*	8	BCS-NI	C
01o	END_TIME	Imaging End Time. The number of UTC seconds into the day, specified in END_DATE (index 01n), when the *last* photon contacted the detector for the *last* collected pixel stored in the NITF image segment. See START_TIME (index 01m) for more guidance. (See also Z.4.2 Times and Dates.) END_DATE and END_TIME may be equal to START_DATE and START_TIME, respectively, for imagery collections modeled as instantaneously collected.	s	*see description*	14	BCS-N	C

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.

This table continues on the next page.

Table Z.3-1 (contd.). SENSRB Data Fields Specifications. (Module 1—contd.) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
01p	GENERATION_ COUNT	Generation Count. The number of times the data contained in this TRE has been updated or adjusted. Zero (00) shall represent that the data is the original NITF form, which might have undergone manipulations reflected in Module 4. If subsequent modifications are made to this metadata through resection or adjustments, this count will be incremented accordingly. (See Z.5.1.5 Image Parameter Post-Collection Adjustments.)	*count*	00 *to* 99	2	BCS-NPI	C
01q	GENERATION_DATE	Generation Date. The date when the current resection or adjustment was made, formatted as YYYYMMDD. This value is ignored if generation count is zero but must contain a valid date if generation count is greater than zero. (See Z.4.2 Times and Dates.)	*date*	*see description*	8	BCS-NI	[C]
01r	GENERATION_TIME	Generation Time. The UTC time of day when the current resection or adjustment was made, formatted as HHMMSS.sss (See Z.5.1.5 Image Parameter Post-Collection Adjustments for specific instructions regarding this field's precision). This value is ignored if generation count is zero but must be a valid time if generation count is greater than zero.	-	000000.000 *to* 235959.999	10	BCS-N	[C]

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 2) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
02	SENSOR_ARRAY_ DATA	Sensor Array Data Flag. Flag field indicating the presence of data describing the sensor array. This module provides information regarding the image collection hardware and process, as is often needed for geopositioning. 'Y' in this field indicates the presence of the Fields 02a to 02i; 'N' omits their presence. Units depend on the values for LENGTH_UNIT (index 01j) and ANGULAR_UNIT (index 01k). **See Z.5.2 Sensor Array Data Module for additional guidance regarding these fields.**	*flag*	Y *or* N	1	BCS-A	R
02a	DETECTION	Detection Type. Specifies the detection spectrum of the sensor array. (See Z.5.2.1 for descriptions of previously approved sample values.)	-	*see NTB registry for approved values.*[c]	20	BCS-A	C
02b	ROW_DETECTORS	Number of Detector Rows. The number of detector rows used in the "instantaneous" collection process. (This number is equivalent to the number of detectors in a column.) This number shall correspond with the used sensor array's physical dimension (ROW_METRIC, index 02d) (See Z.5.2.2.1. Row and Column Detectors.)	-	any > 0	8	BCS-NPI	C
02c	COLUMN_ DETECTORS	Number of Detectors Columns. The number of detector columns used in the "instantaneous" collection process. (This number is equivalent to the number of detectors in a row.) This number shall correspond with the used sensor array's physical dimension (COLUMN_METRIC, index 02e) (See Z.5.2.2.1. Row and Column Detectors.)					
02d	ROW_METRIC	Physical Dimension of Set of All Used Rows. The physical length of the sensor array used in the "instantaneous" collection process as measured along the direction of increasing row indices.[e] This dimension shall correspond with the number of rows of detectors used (ROW_DETECTORS, index 02b). (See Z.5.2.2.2. Row and Column Metrics.)	cm or in	any > 0	8	BCS-N	[C][de]
02e	COLUMN_METRIC	Physical Dimension of Set of All Used Columns. The physical length of the sensor array used in the "instantaneous" collection process as measured along the direction of increasing column indices.[e] This dimension shall correspond with the number of columns of detectors used (COLUMN_DETECTORS, index 02c). (See Z.5.2.2.2. Row and Column Metrics.)					

This table continues on the next page.

02f	FOCAL_LENGTH	Best Known Focal Length. The best known value of the effective focal length.[e] If the sensor's focal length varies with band, provide a nominal value and use the BANDSB TRE to provide the per-band focal lengths. (See Z.5.2.3. Focal Length)	cm or in	any > 0	8	BCS-N	[C][de]
02g	ROW_FOV	Field of View encompassed by Sensor Array Rows. The angle measuring the effective field-of-view projected onto the sensor array center column (i.e. Sensor Vertical Field of View—*not* the half-angle FOV).[e] (See Z.5.2.4. Fields of View.)	deg, rad, or smc	0 *to* 270 deg 0 to 3/2π rad 0 to 3/2 smc	8	BCS-N	[C][de]
02h	COLUMN_FOV	Field of View encompassed by Sensor Array Columns. The angle measuring the effective field-of-view projected onto the sensor array center row (i.e. Sensor Horizontal Field of View—*not* the half-angle FOV).[e] (See Z.5.2.4. Fields of View.)					
02i	CALIBRATED	Focal Length Calibration Flag. Indicates if the provided focal length and/or fields of view are based on a geometric calibration process. 'Y' in this field will indicate that the focal length and/or the fields of view with the detector metrics are based on a geometric calibration. 'N' in this field will indicate that they are not. (See Z.5.2.6. Focal Length Calibration Flag)	*flag*	Y *or* N	1	BCS-A	C

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.
[d] To provide the photogrammetric data typically required for geopositioning, this module must contain, for each dimension (row and column), meaningful values (neither 0 nor the unspecified indicator) in either: (ROW/COLUMN_FOV—indices 02g and 02h) – or – (both ROW/COLUMN_METRIC—indices 02d and 02e—and FOCAL_LENGTH—index 02f); see Z.5.2.5 Sensor Array Relationships for further clarification of the required fields.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 3) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
03	SENSOR_ CALIBRATION_DATA	Sensor Calibration Data Flag. Flag field indicating the presence of geometric sensor calibration parameters. This module provides sensor geometric calibration values and coefficients to facilitate precision geopositioning. 'Y' in this field indicates the presence of the Fields 03a to 03l; 'N' omits their presence. **Additional definitions and implementation guidance for these parameters are provided in Z.5.3 Sensor Calibration Data Module.**	*flag*	Y *or* N	1	BCS-A	R
03a	CALIBRATION_UNIT	Calibration Unit System. Identifies the unit system used for the subsequent calibration parameters. If 'mm', then the parameters are referenced to millimeters. If 'px', then the parameters are referenced in pixel units.	-	mm *or* px	2	BCS-A	C
03b	PRINCIPAL_POINT_ OFFSET_X	Principal Point Offset in *x*-direction (x_0) and *y*-direction (y_0). The number of row-aligned and column-aligned units (±) from the center of the image where the optical axis of the sensor intersects the effective sensor array;[e] see Z.5.3.2 Principal Point Offset. Positive values are in the direction of the positive X_I axis and positive Y_I axis; see Z.4.6.4 Image Coordinate System.	mm or pixels	any	9	BCS-N	[C] [e]
03c	PRINCIPAL_POINT_ OFFSET_Y						
03d	RADIAL_DISTORT_1	First, Second, and Third Radial Distortion Coefficient (k_1, k_2, k_3). These values are defined in Z.5.3.3 Radial Distortion.[e]	mm^{-2} or pixels^{-2}	any	12	BCS-A[f]	[C] [e]
03e	RADIAL_DISTORT_2		mm^{-4} or pixels^{-4}				
03f	RADIAL_DISTORT_3		mm^{-6} or pixels^{-6}				
03g	RADIAL_DISTORT_ LIMIT	Limit of Radial Distortion Fit. The maximum valid radial distance (from the principal point) for the application of the polynomial distortion correction given in Z.5.3.3 Radial Distortion.	mm or pixels	any	9	BCS-N	[C]
03h	DECENT_DISTORT_1	First and Second Decentering Distortion Coefficient (p_1, p_2). These values are defined in Z.5.3.4 Additional Distortions.[e]	mm^{-1} or pixels^{-1}	any	12	BCS-A[f]	[C] [e]
03i	DECENT_DISTORT_2						
03j	AFFINITY_DISTORT_1	First and Second Affinity Distortion Coefficient (b_1, b_2). These values are defined in Z.5.3.4 Additional Distortions.[e]	-	any	12	BCS-A[f]	[C] [e]
03k	AFFINITY_DISTORT_2						
03l	CALIBRATION_DATE	Calibration Report Date. Date when the calibration values above, indices 03a to 03k, were computed. (See Z.5.3.6 Calibration Notes.) Field shall be formatted as YYYYMMDD.	*date*	*see description*	8	BCS-NI	[C]

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.
[f] This field value must be numerical and can be formatted as exponential or floating decimal, see Z.4.3 Numerical Expressions.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 4) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
04	IMAGE_ FORMATION_ DATA	Image Formation Data Flag. Flag field indicating the presence of parameters describing the image formation process. This module provides information regarding how the image array (which may be equal to the recorded NITF array) was formed from the sensor array. This data is typically needed for geopositioning. 'Y' in this field indicates the presence of the Fields 04a to 04k; 'N' omits the presence of Fields 04a to 04s. **See Z.5.4 Image Formation Data Module for additional guidance regarding the definitions and implementation of these fields.**	*flag*	Y *or* N	1	BCS-A	R
04a	METHOD	Imaging Method. Specifies the method the sensor utilized to collect the image data. (See Z.5.4.1.1 for descriptions of previously approved values.)	-	*see NTB registry for approved values.*[c]	15	BCS-A	C
04b	MODE	Imaging Mode. A registered three-digit code indicating the actual operating mode within the sensor system that is used to collect or detect the image data. These three-digit codes are the same codes as used in ACFTB's MPLAN, which relates a physical sensor type with its various modes of operation. MODE will generally be the same as the "planned mode" given in ACFTB's MPLAN. It would differ from MPLAN if the actual mode (MODE) were different from the planned mode (MPLAN).[c]	-	*see NTB registry for approved values*[c]	3	BCS-A	C
04c	ROW_COUNT	Row and Column Count. The number of rows (m) and columns (n) in the Initial Image array. (See Z.5.4.2.1. Row and Column Count.)	pixel	any positive	8	BCS-NPI	C
04d	COLUMN_COUNT						
04e	ROW_SET	Row and Column Detection Set. The number of rows and columns collected simultaneously to form the physical sensor array. Typically these values will be equal to the module 02 fields ROW_DETECTORS and COLUMN_DETECTORS, (indices 02b and 02c). These values can also be less than the module 02 detectors count when only a subset of the physical sensor is used to populate each framelet of the Initial Image array. It is generally the case that the ratio of the row set to the row count will be an integer, and similarly for the column set to the column count, meaning that an integer number of sets (framelets) are required to populate each direction of the Initial Image. For instantaneous framing sensors, these values shall be equal to the row and column counts (indices 04c and 04d). For a single-line pushbroom (as an example), one field shall have a meaningful value (equal to the row or column count) and the other shall be one.	pixel	any positive	8	BCS-NI	C
04f	COLUMN_SET						

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 4—contd.) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
04g	ROW_RATE	Row and Column Detection Rate. The durations in time, the “step time”, between starting the capture of the first detection set and starting the capture of the second detection set in that same row or column when forming the image array. For framing sensors, both of these values shall be zeros (or optionally the shutter speed or integration time). For pushbroom sensors, one field (in the direction of the scanning) shall have a meaningful value and the other shall be zero (or the shutter speed/integration time). See Z.5.4.2.3 for details regarding use of this field to define the directions of motion in a scanning sensor.	s	any	10	BCS-N	C
04h	COLUMN_RATE						
04i	FIRST_PIXEL_ROW	Row and Column of First Collected Pixel. The row and column within the Initial Image array containing the first collected pixel or line set of the image array. See Z.5.4.2.4.	pixel	any positive	8	BCS-NPI	C
04j	FIRST_PIXEL_ COLUMN						
04k	TRANSFORM_ PARAMS	Number of Image Transform Parameters Provided. This flag field indicates the number of subsequent fields (indices 04l to 04s) that are present; “0” omits the presence of these fields. See Z.5.4.5 Image Transformations for additional information.	*count*	0 *to* 8	1	BCS-NPI	C
04l	TRANSFORM_ PARAM_1	Image Transform Parameters ($h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8$)[f]. These parameters are defined in Z.5.4.5.3 Image Transformation Equations. They provide the coefficients of the transformation equations that convert between the image coordinates associated with the metadata in this TRE and those corresponding to the NITF image segment’s data array. If the metadata values in this TRE correspond with the NITF-stored image array, these parameters would be the elements of an identity matrix, which would be assumed if 04k is 0. A Transform Parameter must be provided if its number is less than or equal to TRANSFORM_PARAMS. Otherwise the parameter is not provided. A meaningful value for each provided parameter is required.	-	any	12	BCS-A[f]	C
04m	TRANSFORM_ PARAM_2						
04n	TRANSFORM_ PARAM_3						
04o	TRANSFORM_ PARAM_4						
04p	TRANSFORM_ PARAM_5						
04q	TRANSFORM_ PARAM_6						
04r	TRANSFORM_ PARAM_7						
04s	TRANSFORM_ PARAM_8						

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[f] The field value must be numerical and can be formatted as exponential or floating decimal, see Z.4.3 Numerical Expressions.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Modules 5 and 6) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
REFERENCE TIME/PIXEL		Either Reference Time or Reference Pixel of Applicability must be provided in each and every instantiation of the SENSRB TRE. It specifies the time or pixel location to which the dynamic parameters apply. If both reference methods are provided, the reference pixel location will take precedence over the reference time. **See Z.5.5 Reference Time or Pixel Module for additional guidance regarding the implementation of these fields.**					
05a	REFERENCE_TIME	Reference Time of Applicability. This is the time for which the values in the TRE are to be applied. The value is in seconds before (negative) or after (positive) START_TIME. A zero value places the reference time at the imaging start time, such as would be the case for sensors modeled with instantaneous collections—where START_TIME equals END_TIME. This field shall contain a valid time if either REFERENCE_PIXEL_ROW or REFERENCE_PIXEL_ COLUMN (index 05b or 05c) is unspecified.	s	any	12	BCS-N	[R]
05b	REFERENCE_ROW	Reference Pixel Row and Column of Applicability. Reference row and column pixel index relative to the NITF image segment at which the field values in this TRE apply. The pixel indices shall refer to the original NITF image segment associated with this TRE (see Z.4.5 Array Types). Pixel indexing is per the standard NITF raster order—column indices increasing to the right and row indices increasing downward from the viewer's perspective; see Z.4.6.5 Image Array Common Coordinate System. These fields shall contain valid values if REFERENCE_TIME (index 05a) is unspecified.	decimal pixel	any	8	BCS-N	[R]
05c	REFERENCE_ COLUMN						
SENSOR POSITION DATA		These fields, from which the sensor location can be determined, must be included in each and every instantiation of the SENSRB TRE. **Additional guidance regarding these fields is provided in Z.5.6 Sensor Position Data Module.**					
06a	LATITUDE_OR_X	Sensor or Platform Latitude, Longitude, and Altitude or ECEF X, Y, Z Position. These values provide the platform or sensor geolocation;[e] see Z.5.6 Sensor Position Data Module. They shall reflect the geodetic type specified in GEODETIC_TYPE (index 01h).	deg *or* m *or* ft	*see description*	11	BCS-N	R[e]
06b	LONGITUDE_OR_Y	If geodetic coordinates are used, the latitude and longitude shall be in degrees and between 0 to ±90 and 0 to ±180, respectively. The altitude will be relative to the datum specified by ELEVATION_DATUM (index 01i). Its units are specified by LENGTH_UNIT (index 01j) and can be any numeric value (positive or negative) fitting within the field size.	deg *or* m *or* ft	*see description*	12	BCS-N	R[e]
06c	ALTITUDE_OR_Z	If geocentric coordinates are used, the X_E, Y_E, and Z_E coordinates shall reflect the units specified in LENGTH_ UNIT (index 01j). Z.4.6.1.2 Geocentric Coordinate System provides reasonable limits on these geocentric Cartesian coordinate values.	m *or* ft	*see description*	11	BCS-N	R[e]
06d	SENSOR_X_OFFSET	Sensor X, Y, and Z Position Offset Relative to Platform Coordinate System. The location of the sensor perspective center relative to the above-reported position;[e] measured respectively in the *platform's* X, Y, and Z direction. See Z.5.6 Sensor Position Data Module for further clarification. These values shall reflect the units specified in LENGTH_UNIT (index 01j).	m *or* ft	any	8	BCS-N	R[e]
06e	SENSOR_Y_OFFSET						
06f	SENSOR_Z_OFFSET						

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 7) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
07	ATTITUDE_ EULER_ANGLES	Attitude Euler Angle Flag. Flag field indicating the combined presence of platform and sensor Euler angle values. This module provides platform and sensor attitude measurements in terms of their respective Euler angles. The sensor attitude, as provided either by this module or by Modules 8 or 9, is typically required for geopositioning. If provided, the parameter values in Modules 8 and 9 take precedence over Module 7. ‘Y’ in this field indicates the presence of the Fields 07a to 07h; ‘N’ omits their presence. The units depend on the value of ANGULAR_UNIT (index 01k). **Additional definitions and implementation guidance for these fields are provided in Z.5.7 Euler Angles Module.**	*flag*	Y *or* N	1	BCS-A	R
07a	SENSOR_ANGLE_ MODEL	Type of Sensor Angle Rotations. Specifies the Euler angle model using a coded value. Typically the model type will depend on the sensor’s mounting and/or pointing gimbal layout. The definitions for the types of sensor angle rotations are provided in Z.5.7.3.1 Sensor Angle Model.	-	1, 2, *or* 3	1	BCS-NPI	C
07b	SENSOR_ANGLE_1	First, Second, and Third Sensor Rotation Angles. Rotations[e] to the sensor coordinate system from the platform or local-level coordinate system. The rotation definitions are given in Z.5.7.3 Sensor Euler Angles. The definitions depend on SENSOR_ANGLE_MODEL (index 07a) and PLATFORM_RELATIVE (index 07e); see Z.5.7.3. Sensor Euler Angles. All three fields shall be populated with meaningful values when this module is present.	deg, rad, or sc	0 *to* ±180 deg	10	BCS-N	C [e]
07c	SENSOR_ANGLE_2			0 *to* ±90 deg	9	BCS-N	C [e]
07d	SENSOR_ANGLE_3			0 *to* ±180 deg	10	BCS-N	C [e]
07e	PLATFORM_ RELATIVE	Sensor Angles Relative to Platform Flag. If this flag is set to ‘Y’, the above sensor angles are relative to the platform coordinate system (see Z.5.7.3 Sensor Euler Angles); otherwise, the above angles are relative to the time-relevant, sensor-local NED coordinate system, regardless of the GEODETIC_TYPE (index 01h). In other words, even when GEODETIC_TYPE is geocentric, these values (if not platform relative) are local NED relative.	*flag*	Y *or* N	1	BCS-A	C
07f	PLATFORM_ HEADING	Platform Heading, Pitch, and Roll Angle. These three Euler angles[e] define the attitude of the platform coordinate system; see Z.5.7.2 Platform Euler Angles. They are relative to the time-relevant, platform/sensor-local NED, regardless of the GEODETIC_TYPE (index 01h). Meaningful values for these three angles are required if PLATFORM_RELATIVE (index 07e) is set to ‘Y’; otherwise, these angles may contain the unspecified indicator.	deg, rad, or sc	0 *to* 360 deg	9	BCS-N	[C] [e]
07g	PLATFORM_PITCH			0 *to* ±90 deg	9	BCS-N	[C] [e]
07h	PLATFORM_ROLL			0 *to* ±180 deg	10	BCS-N	[C] [e]

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Modules 8 and 9) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
08	ATTITUDE_ UNIT_VECTORS	Attitude Unit Vector Flag. Flag field indicating the presence of image coordinate system's unit vectors components. This module provides the image coordinate system direction cosines relative to the sensor-local NED or the ECEF coordinate system. The sensor attitude, as provided either by this module or by Modules 7 or 9, is typically required for geopositioning. The parameter values in this module take precedence over those in Modules 7 and 9. 'Y' in this field indicates the presence of the Fields 08a to 08i; 'N' omits their presence. **See Z.5.8 Unit Vectors Module for additional information regarding these fields.**	*flag*	Y *or* N	1	BCS-A	R
08a	ICX_NORTH_OR_X	Image Coordinate (IC) System X, Y, and Z Axes Attitude Unit Vectors Relative to NED or ECEF Coordinate Frame. The coordinates of the image coordinate system-aligned unit vectors (or equivalently the image coordinate system's direction cosines). These component values are relative to either the sensor-local NED or the ECEF coordinate system, as specified by GEODETIC_TYPE (index 01h). All fields shall be populated with meaningful values when this module is present. The uncertainties[e] associated with these values are reported in Module 14 (Uncertainty Data) using a unique method as described in Z.5.14.5 Attitude Unit Vector Uncertainties.	-	0 *to* ±1	10	BCS-N	C[e]
08b	ICX_EAST_OR_Y						
08c	ICX_DOWN_OR_Z						
08d	ICY_NORTH_OR_X						
08e	ICY_EAST_OR_Y						
08f	ICY_DOWN_OR_Z						
08g	ICZ_NORTH_OR_X						
08h	ICZ_EAST_OR_Y						
08i	ICZ_DOWN_OR_Z						
09	ATTITUDE_ QUATERNION	Attitude Quaternion Flag. Flag field indicating the presence of a sensor attitude quaternion. This module provides the sensor attitude using a quaternion relative to the sensor-local NED or the ECEF coordinate system. The sensor attitude, as provided either by this module or by Modules 7 or 9, is typically required for geopositioning. The parameter values in this module take precedence over those in Module 7, but those in Module 8 take precedence over these. 'Y' in this field indicates the presence of the Fields 09a to 09d; 'N' omits their presence. **See Z.5.9 Quaternions Module for additional information regarding these fields.**	*flag*	Y *or* N	1	BCS-A	R
09a	ATTITUDE_Q1	Attitude Quaternion Vector Components. Three vector elements of a normalized quaternion[e] defining the conceptual sensor attitude relative to either the sensor-local NED or the ECEF coordinate system, as specified by GEODETIC_ TYPE (index 01h).	-	0 *to* ±1	10	BCS-N	C[e]
09b	ATTITUDE_Q2						
09c	ATTITUDE_Q3						
09d	ATTITUDE_Q4	Attitude Quaternion Scalar Component. Fourth (scalar) element of a normalized quaternion[e] (absolute magnitude of the four-element quaternion is unity). See ATTITUDE_Q1 (index 09a) description above.	-	0 *to* ±1	10	BCS-N	C[e]

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Modules 10 and 11) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
10	SENSOR_VELOCITY_DATA	Sensor Velocity Data Flag. Flag field indicating the presence of sensor velocity data. This module provides the sensor velocity components relative to the sensor-local NED or the ECEF coordinate system, as may be useful with a pushbroom or whiskbroom sensor. 'Y' in this field indicates the presence of the Fields 10a to 10c; 'N' omits their presence. **See Z.5.10 Sensor Velocity Module for additional information regarding these fields.**	*flag*	Y *or* N	1	BCS-A	R
10a	VELOCITY_ NORTH_OR_X	Sensor North, East, and Down Velocity Vectors. The velocity vector components of the sensor's velocity vector.[e] The velocity components are relative to either the sensor-local NED or the ECEF coordinate system, as specified by GEODETIC_TYPE (index 01h). Units are specified by LENGTH_UNIT (index 01j).	m/s or ft/s	any	9	BCS-N	C [e]
10b	VELOCITY_ EAST_OR_Y						
10c	VELOCITY_ DOWN_OR_Z						
11	POINT_SET_DATA	Polygon or Point Set Count Flag. Flag field indicating the number of points sets associated with the NITF image segment. This module provides point sets (a set of one or more points such as polygon vertices) to identify or bound features within the image and provides geometric information associated with those points. '00' in this field omits the presence of Fields 11a to 11h. A non-zero value defines the number of point sets included in the TRE; see Z.4.8 Looping Fields Concept. **See Z.5.11 Point DataSets Module for additional information regarding these fields.**	*count*	00 *to* 99	2	BCS-NPI	R
11a	POINT_SET_ TYPE_MM	Type of M^{th} Point Set. The type of point set. This field identifies the type of feature being identified or bound by the m^{th} point set. This field requires NTB registration, current values are:[c] **Sensor Aimpoint, Image Center, Image Footprint, Ground Area, Ground Points, Point of Interest, Area of Interest, LRF Measurements**	-	*see NTB registry for approved values.*[c]	25	BCS-A	L
11b	POINT_COUNT_MM	Number of Points in M^{th} Set. The number of points used to identify or bound the m^{th} feature identified by POINT_SET_TYPE_MM (index 11a.*m*). For the m^{th} occurrence of fields 11a and 11b, fields 11c through 11h occur POINT_COUNT_MM (index 11b.*m*) times.	*count*	001 *to* 999	3	BCS-NPI	L
11c	P_ROW_NNN	Row and Column Location for N^{th} Point. The NITF-stored array coordinates for the n^{th} point in the m^{th} point set.	decimal pixel	any	8	BCS-N	L
11d	P_COLUMN_NNN						
11e	P_LATITUDE_NNN	Latitude, Longitude, Elevation, and Range for N^{th} Point. The estimated geographic coordinates and elevation, as well as measured range (line-of-sight distance from sensor to point), for the n^{th} point in the m^{th} point set.[e] The geographic coordinates of the imaged feature or region will be provided using the latitude and longitude, *regardless* of the value specified in GEODETIC_ TYPE (index 01h). The elevation will be relative to the datum specified by ELEVATION_ DATUM (index 01i). Fields needing data that is unavailable should use the unspecified indicator.	deg	0 *to* ±90	10	BCS-N	[L] [e]
11f	P_LONGITUDE_NNN		deg	0 *to* ±180	11	BCS-N	[L] [e]
11g	P_ELEVATION_NNN		m or ft	*see note* g	6	BCS-N	[L] [e]
11h	P_RANGE_NNN		m or ft	any positive	8	BCS-N	[L] [e]

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.
[e] Uncertainties associated with these values should be provided in Module 14 (Uncertainty Data); see Z.5.14 Uncertainty Data Module.
[g] Earth-surface elevations would fall between -500 to +9000 m (-1500 to 30,000 ft).

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Modules 12 and 13) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET [a]	TYPE [b]
12	TIME_STAMPED_ DATA_SETS	<u>Time Stamp Count Flag</u>. Flag field indicating the number of dynamic parameters recorded with time-stamps in this TRE. This module associates parameter values directly to multiple specific reference times. '00' in this field omits the presence of Fields 12a to 12d. A non-zero value defines the number of time-stamped parameters present in the module; see Z.4.8 Looping Fields Concept. The presence of this module indicates the values stored in the index-referenced fields prior to this module shall be nominal, average, or approximate values per the entire image segment. **See Z.5.12 Time-Stamped Data Module for additional guidance and possible exception.**	*count*	00 *to* 99	2	BCS-NPI	R
12a	TIME_STAMP_ TYPE_MM	<u>Index of the M^{th} Time-Stamped Parameter</u>. The m^{th} time-stamped parameter's index value. Indexing is limited to reasonably applicable fields (dynamic parameters) in the value range.	*index*	02b *to* 10c	3	BCS-A	L
12b	TIME_STAMP_ COUNT_MM	<u>Number of Occurrences of the M^{th} Parameter</u>. The number of time stamps for which this parameter's value is recorded. This field determines the number of times the two fields below (12c and 12d) repeat to record the time stamps and values of the m^{th} indexed parameter.	*count*	1 *to* 9999	4	BCS-NPI	L
12c	TIME_STAMP_ TIME_NNNN	<u>The N^{th} Instance of a Time-Stamp Time</u>. The reference time in seconds relative to START_ TIME (index 01m) associated with the subsequent value of the m^{th} parameter. Negative times are possible, if the parameter is measured before the imaging start time.	s	any	12	BCS-N	L
12d	TIME_STAMP_ VALUE_NNNN	<u>The N^{th} Instance of the M^{th} Time-Stamped Parameter's Value</u>. The parameter value at the time-stamp time above (TIME_STAMP_TIME_NNNN, index 12c.*m.n*).	*see note* h *below*				L
13	PIXEL_ REFERENCED_ DATA_SETS	<u>Pixel Reference Count Flag</u>. Flag field indicating the number of dynamic parameters recorded with references to pixel indices. This module associates parameter values directly to multiple NITF image segment locations. '00' in this field omits the presence of Fields 13a to 13e. A non-zero value defines the number of pixel-referenced parameters present in the module; see Z.4.8 Looping Fields Concept. The presence of this module indicates the values stored in the index-referenced fields prior to this module shall be nominal, average, or approximate values per the entire image segment. **See Z.5.13 Pixel-Referenced Data Module for additional guidance and possible exception.**	*count*	00 *to* 99	2	BCS-NPI	R
13a	PIXEL_REFERENCE_ TYPE_MM	<u>Index of the M^{th} Pixel-Referenced Parameter</u>. The m^{th} pixel-referenced parameter's index value. Indexing is limited to reasonably applicable fields (dynamic parameters) in the range.	*index*	02b *to* 10c	3	BCS-A	L
13b	PIXEL_REFERENCE_ COUNT_MM	<u>Number of Occurrences of the M^{th} Parameter</u>. The number of pixel reference locations for which this parameter's value is recorded. This field determines the number of times the three fields (13c, 13d, 13d) repeat to record the pixel locations and values of the m^{th} parameter.	*count*	1 *to* 9999	4	BCS-NPI	L
13c	PIXEL_REFERENCE_ ROW_NNNN	<u>The N^{th} Instance of a Reference Pixel Row and Column Index</u>. The row and column index for the reference pixel associated with the subsequent value of the m^{th} parameter. The center row or center column index shall be used, respectively, when the parameter is only column- or row-varying. Each set of pixel coordinates pertains to the NITF-stored image segment.	decimal pixel	any	8	BCS-N	L
13d	PIXEL_REFERENCE_ COLUMN_NNNN						
13e	PIXEL_REFERENCE_ VALUE_NNNN	<u>The N^{th} Instance of the M^{th} Pixel-Referenced Parameter's Value</u>. The parameter value associated with the reference pixel location defined above (indices 13c.*m.n* and 13d.*m.n*).	*see note* h *below*				L

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[h] The units, range, size, and character set permitted for this value are specified as they are for the m^{th} indexed parameter identified by index 12a.*m* or 13a.*m*.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 14) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR SET[a]	TYPE[b]
14	UNCERTAINTY_ DATA	Uncertainty Data Flag. Flag field indicating the number of uncertainty-related values in the module. This module provides estimated standard deviations or correlation coefficients for select parameters. '000' in this field omits the presence of Fields 14a to 14c. A non-zero value defines the number of data sets (parameter index pairs and uncertainty values) found in this module; see Z.4.8 Looping Fields Concept. When provided, parameter uncertainties can be projected or propagated into the object space as geoposition uncertainty estimates. These estimated geoposition uncertainties are required for a number of applications. **See Z.5.14 Uncertainty Data Module for additional guidance regarding these fields.**	*count*	000 *to* 999	3	BCS-NPI	R
14a	UNCERTAINTY_ FIRST_TYPE_NNN	First Index of Parameter with Reported Uncertainty or Correlation. The first index for the n^{th} parameter pair to which the subsequent (n^{th}) uncertainty estimate or correlation coefficient (UNCERTAINTY_VALUE_NNN, index 14c.*n)* pertains. Indexing is limited to reasonably applicable fields in the value range (spatial or temporal parameters with potential uncertainties). Indices for looping-conditional fields can be followed by up to seven digits with a decimal point separator. The digits before the separator identify the m^{th} parameter (outer loop), and the digits after the separator identify the n^{th} instance (inner loop) of that m^{th} parameter. See Z.4.8.2 Looping Field Indexing for additional clarification.	*index*	*any appropriate index between* 02b *to* 15d	11	BCS-A	L
14b	UNCERTAINTY_ SECOND_TYPE_NNN	Second Index of Parameter with Reported Uncertainty. The second index for the n^{th} parameter pair to which the subsequent uncertainty estimate or correlation coefficient pertains. Shall be either be unspecified or identical to the first index to indicate that the subsequent value (index 14c.*n*) is a standard deviation. Must be an appropriate index, different from the first index, to indicate that the subsequent value is a correlation coefficient (relating the two indexed parameters). See UNCERTAINTY_FIRST_TYPE_NNN (index 14a) above for additional guidance.	*index*	*any appropriate index between* 02b *to* 15d	11	BCS-A	[L]
14c	UNCERTAINTY_ VALUE_NNN	Uncertainty or Correlation Value. The uncertainty estimate or correlation coefficient associated with the previously-identified-parameters' value(s). If the n^{th} UNCERTAINTY_SECOND_ TYPE_NNN (index 14b.*n*) is unspecified or is identical to the n^{th} UNCERTAINTY_FIRST_ TYPE_ NNN (index 14a.*n*), this n^{th} value shall be the standard deviation (1σ) estimate for the indexed parameter. If two different indices appear in the two index fields (14a.*n* and 14b.*n*), then the value is a correlation coefficient relating those two indexed parameters.	*see note* i *below*	*for standard deviations:* any positive *for correlation coefficients:* -1 *to* +1	10	BCS-A[f]	L

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[f] The field value must be numerical and can be formatted as exponential or floating decimal, see Z.4.3 Numerical Expressions.
[i] Standard deviations will have the same units as the indexed parameter; see Z.5.14.4 Geodetic Coordinate Uncertainties for the exceptions to this rule. Correlation coefficients are unitless.

This table continues on the next page.

Table Z.3-1. (contd.) SENSRB Data Fields Specifications. (Module 15) (See Z.2.2 Data Fields for explanations regarding this table and its content.)

INDEX	FIELD NAME	DESCRIPTION	UNITS	VALUE RANGE	BYTE SIZE	CHAR[a] SET	TYPE[b]
15	ADDITIONAL_ PARAMETER_DATA	Additional Parameter Flag. Flag field indicating the number of additional parameters in the module. This module allows for the registration of new data fields to accommodate requirements not already met by SENSRB. '000' in this field omits the presence of Fields 15a to 15d. A non-zero value defines the number of parameter sets (parameter name, size, count, and value(s)) found in this module; see Z.4.8 Looping Fields Concept. **See Z.5.15 Additional Parameter Data Module for additional guidance regarding these fields.**	*count*	000 *to* 999	3	BCS-NPI	R
15a	PARAMETER_ NAME_MMM	Additional Parameter Name. Name of the *m*th additional parameter. This name must be unique, explicit, and registered with the NTB.[c] Registration will ensure uniqueness to avoid ambiguity and will document the parameter definition and utility for community awareness and future TRE developments. As with all alphanumeric data, the name should be left justified and blank filled. (See Z.2.2.4. Field Definitions)	-	*see NTB registry for approved values.*[c]	25	BCS-A	L
15b	PARAMETER_ SIZE_MMM	Additional Parameter Field Size. The size of the *m*th additional parameter value field. This user-specified size allows for parameters of variable sizes. This size shall apply to all instantiations of PARAMETER_VALUE_NNNN associated with the *m*th additional parameter.	bytes	1 *to* 999	3	BCS-NPI	L
15c	PARAMETER_ COUNT_MMM	Number of Occurrences of the Additional Parameter. The number of times the *m*th parameter is recorded herein. This field determines the number of times the field below (PARAMETER_VALUE_NNNN, index 15d) is repeated for the *m*th additional parameter.	*count*	1 *to* 9999	4	BCS-NPI	L
15d	PARAMETER_ VALUE_NNNN	Additional Parameter Value. The value for the *n*th instantiation of the *m*th additional parameter.	*See note* j *below*				L

[a] BCS-A: Basic Character Set–Alpha, BCS-N: Basic Character Set–Numeric, BCS-NI: Basic Character Set–Numeric Integer, BCS-NPI: Basic Character Set–Numeric Positive Integer.
[b] R: Required, C: Conditional, L: Looping-conditional, [ ]: BCS hyphens allowed for entire field.
[c] For instruction on how to review the current NTB registry and for guidance regarding the registration of additional values, see Z.4.1 NTB-Registered Field Values.
[j] The value's units, range, and character set will be established for each additional parameter as part of the NTB registration process. The size (byte allotment) for this parameter is as specified by PARAMETER_SIZE_MMM (index 15b.*m*)

Z.4. GENERAL TECHNICAL CONCEPTS

This section provides guidance and clarification on concepts that extend across multiple modules.

Z.4.1. NTB-REGISTERED FIELD VALUES

Several fields in SENSRB require a value from the list of those currently registered with the NTB. In some cases, registered values are listed directly in Table Z.3-1 SENSRB Data Fields Specifications. Currently approved lists of registered values for fields in SENSRB and other TREs can be found at the website, http://jitc.fhu.disa.mil/cgi/nitfcte/trereg.aspx.

The process for registering a new value with the NTB can be found at the website, http://jitc.fhu.disa.mil/nitf/tag_reg/submit_values.html or by writing the NTB at ntbchair@nga.mil. SENSRB implementers intending to register new field values should consider carefully their proposed values. A field value should be unambiguous (unique) and meaningful (explicit) to potential users of the TRE. It should also be consistent with the pattern of naming conventions. Note that SENSRB has adopted a convention of Title Case rather than ALL CAPS for non-acronyms (such as 'Airborne').

Z.4.2. TIMES AND DATES

Any absolute times and dates in SENSRB (such as the imaging start and end times and dates) are all based on Universal Coordinated Time (UTC).

Dates in SENSRB are according to the Gregorian calendar and are presented in the YYYYMMDD format—where YYYY represents the year, MM represents the month within the year, and DD represents the day within the month. Single digit months and days shall include a leading zero. For example, the date 5 April 1997 would be represented as "19970405". If an element of a date, such as a day and/or month (or a portion thereof), is unknown; the unknown elements shall be replaced with a hyphen. For example, a date only known to be during the first half of 2010 would be represented by "20100---".

All time values in SENSRB are presented in UTC seconds and decimal portions thereof relative to the start of the day or the imaging start time. The exception to this is GENERATION_TIME (index 01r), which is in the HHMMSS.sss format relative to the start of the day; see Z.5.1.5 Image Parameter Post-Collection Adjustments for additional details.

Note: UTC may differ from GPS time by an integer number of leap seconds.

Z.4.3. NUMERICAL EXPRESSIONS[1]

Numerical data must be expressed in decimal, either with or without a decimal point, and either with or without an accompanying exponent of 10. These values shall be right justified with padding on the left. Padding is accomplished with leading zeros, unless the BCS-A character set is used; in which case, BCS spaces may precede the digits (and sign, if used). For additional clarification, see Z.2.2.4 Field Formats.

Z.4.3.1. Plus and Minus Signs

Positive values *may* be preceded by a plus sign (+), but negative values *must* be preceded by a minus sign (-). If used, a plus or minus sign shall immediately precede the first numeric digit (including leading zeros used for padding). Thus, these signs—when used—must be the leftmost character, for fields using BCS-N or BCS-NI. On the other hand, if BCS-A is used, a leading sign may be preceded by BCS spaces. For fields using the BCS-NPI character set, a plus sign is implied; but no sign is allowed.[2] The use of plus or minus signs may also be further specified by a field's format description; see Z.4.6.1.1 Geodetic Coordinates, as an example.

---

[1] See footnote in Z.2.2.4 Field Formats (Numeric Data) regarding IEEE 754 numerical representations.
[2] The plus and minus signs ('+' and '-') are not included in the BCS-NPI character set.

Note: A numerically-zero value may also be preceded by a plus or minus sign, if that sign provides additional information.

Z.4.3.2. Number Representations

Numerical values can be represented within a SENSRB field in up to three ways—as allowed by that field's value range and character set.[1] In each case, plus and minus signs and leading zeros shall be employed as described in the previous two paragraphs.

- Integer. These values are made up of numerical digits, but do not have a decimal point. (Integer numbers can be represented using BCS-A, BCS-N, BCS-NI, or BCS-NPI.)
- Floating Decimal. These values are populated with the possibility of a decimal point being included anywhere[2] within the series of digits (including the leading position, when no other means exists to express the desired precision). (Floating decimal numbers can be represented using BCS-A or BCS-N.)
- Exponential Notation. These values are presented using scientific notation. BCS spaces may be used before the mantissa's first digit (or sign, if any) to help fill the specified field size. Leading zeros may also be used in the mantissa, but a decimal point shall follow the mantissa's first *significant* digit. The exponent value shall be indicated with either a preceding 'E' or 'e' from the BCS-A. A positive exponent may include a leading plus sign; a negative exponent must include a leading minus sign. Apart from this sign (if used), the exponent shall have only one or two digits, which may include a leading zero; but it shall not have a decimal point nor a BCS space. (Exponential notation numbers can only be represented using BCS-A.)

Z.4.3.3. Numerical Precision

SENSRB TRE implementers should use good judgment in providing numerical values—expressing each number to the precision that both reflects its uncertainty and, if possible, facilitates intended calculations. For example, if a value were only known to one-tenth of a unit and it was to be reported in an 8-character field, it would be best reported as 000034.6 (or +00034.6), rather than as 34.62537 or even as 34.60000. This is because the latter two representations might be used to imply that the value is known to better than the nearest ten-thousandth of a unit. In any case, those populating the TRE are strongly encouraged to provide appropriate uncertainty values in Module 14 (Uncertainty Data), whenever possible.

SENSRB TRE users are also cautioned that the accuracy of values reported cannot be reliably inferred from the reported precision. On the other hand, uncertainty values reported in Module 14 are meant to indicate the accuracy estimates and should be used for that purpose whenever they are made available.

SENSRB TRE users should also use caution in reading metadata into machine-usable numerical values (such as single-precision or double-precision floating point). Care should be used to ensure the receiving machine formats can accommodate both the numerical range and the full precision of the metadata elements. Values should always be read and stored in a format which maintains both the provided precision and the numerical integrity of the values throughout any subsequent computations.

Z.4.4. MEASUREMENT UNITS

The systems of units used in SENSRB depend on values set in LENGTH_UNIT (index 01j), ANGULAR_UNIT (index 01k), and CALIBRATION_UNIT (index 03a). The value set in GEODETIC_TYPE (index 01h) further determines whether the sensor position is reported using geodetic or geocentric coordinate units; see Z.5.6 Sensor Position Data Module for an explanation.

[1] See footnote in Z.2.2.4 Field Formats (Numeric Data) regarding IEEE 754 numerical representations.
[2] Latitudes and longitudes, as well as the generation time, have prescribed formats that somewhat limit where a decimal point may be used; see Z.4.6.1.1 Geodetic Coordinates and Z.5.1.5 Image Parameter Post-Collection Adjustments, respectively.

Z.4.4.1. Linear Units

Most linear values use either the metric (international system—SI) or English (English engineering—EE) unit systems as specified in LENGTH_UNIT (index 01j). The metric system is preferred. The unit options for both cases are listed here:

International System of Units	English Engineering System of Units
centimeters (cm)	inches (in)
meters (m)	feet (ft)
meters per second (m/s)	feet per second (ft/s)

Additionally, the basic linear unit of measure for sensor calibration data shall be either millimeters (mm) or pixels, as specified in CALIBRATION_UNIT (index 03a)—see Z.5.3 Sensor Calibration Data Module.

Z.4.4.2. Angular Units

Most angular values are reported in degrees (deg), radians (rad), or semi-circles (smc) as specified in ANGULAR_UNIT (index 01k). The exception is geodetic coordinates (latitudes and longitudes), which are always reported in degrees. All angles are planar such that a circle is made up of 360°, 2π radians, or 2 semi-circles. As an example, 90° is equivalent to $\pi/2$ radians and 0.5 semi-circles. The use of degrees is preferred. The use of radians is discouraged because of potentially ambiguous numerical definitions of π.

Z.4.5. SENSOR AND IMAGE ARRAY TYPES

Three types of arrays are commonly referred to within this documentation. The difference between these arrays may potentially be vague for certain applications. Therefore, it is incumbent upon those populating the SENSRB TRE values to decide upon, and *consistently* apply, array definitions suitable to their implementation. Under these conditions, the exploiter can appropriately use the values provided in SENSRB without necessarily knowing all the sensor and collection specifics. The following provides guidance on defining and using the three array types in SENSRB.

Z.4.5.1. Physical Sensor Array

The physical sensor array is defined as a group of electro-optical detectors within the sensor. Sensors may group multiple detector arrays with internal processing to create a single effective array. The physical sensor array should be conceptualized as the most basic aggregation of detectors which can appropriately be modeled by the parameters of Module 2 (Sensor Array Data). For framing sensors, this would be the two-dimensional focal plane array used to create a framelet. For scanning sensors—such as pushbroom or whiskbroom systems, this would be the linear array used to create a line of the image. The sensor calibration, position, attitude, and velocity data (Modules 3 and 6-10) shall all apply to the physical sensor array and its perspective center. See also Z.4.6.3 Sensor Coordinate System.

Z.4.5.2. Initial Image Array

The initial image array is made up of the image rows and columns as collected by the physical sensor array and initially assembled by the sensor system. The formation of the initial image array from the physical sensor array is defined by the image formation data (Module 4). The initial image array is the same as the original NITF-stored image array, unless additional image transformations are applied, as described in Z.5.4.5 Image Transformation Parameters.

Z.4.5.3. NITF-Stored Image Array

This is the image segment array as *originally* stored into a NITF file. (Subsequent NITF files may contain a further manipulated image. These further manipulations are defined using the ICHIPB TRE—see *STDI-0002, Appendix B: ICHIPB*.) The rows and columns of this array are as they were formed into the initial image array with the additional application of the image transformation parameters (TRANSFORM_PARAM_1-8, indices 04l to 04s). Reference pixels (Module 5) and point set data (Module

11) shall apply to the NITF-stored image array's pixel coordinates; see Z.4.6.5 Image Array Common Coordinate System.

Z.4.5.4. Array Type Relationships

The arrays described in the previous paragraphs are depicted in Figure Z.4-1 below for three different cases. In the first row, a framing sensor is depicted, where the physical sensor array (a focal plane array) produces equivalent initial image and NITF-stored image arrays. The transformation from one array to the next is accomplished with an identity (**I**) operation. In the second row, an unstable focal plane array is depicted. In this case, the initially formed image is manipulated (stabilized) before the image is first stored as a NITF file. The stabilization is represented by a transformation **H** defined by image transformation parameters that can be stored within Module 4 (Image Formation Data). In the last row, a pushbroom sensor is depicted. In this case the physical sensor may only produce a line of the initial image at a given instant. As the lines are collected, the initial image is formed as described by the fields in Module 4 (Image Formation Data). The initial image array is then stored as the NITF-stored array.

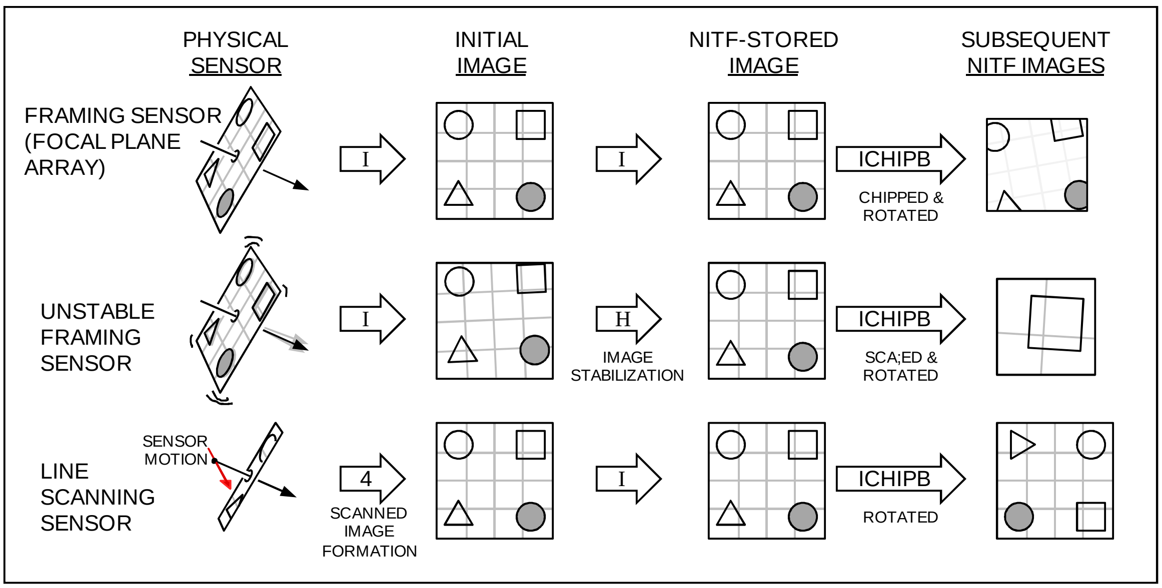


Figure Z.4-1. SENSRB Array Relationships. The flow (left to right) of data for three separate scenarios.

In each of the three cases, subsequent NITF images that have been chipped, scaled, and/or rotated can be created from the NITF-stored array. These subsequent transformations are not accounted for in SENSRB, but are defined in the image chipping TRE—see *STDI-0002, Appendix B: ICHIPB*.

Z.4.6. COORDINATE SYSTEM DEFINITIONS

This section defines the various coordinate systems that are relevant to the SENSRB TRE. These include geospatial, platform, conceptual sensor, and three image-related coordinate systems. Transformations between coordinate systems are detailed in Z.6.2 Coordinate System Transformations.

Z.4.6.1. Geospatial Coordinate Systems

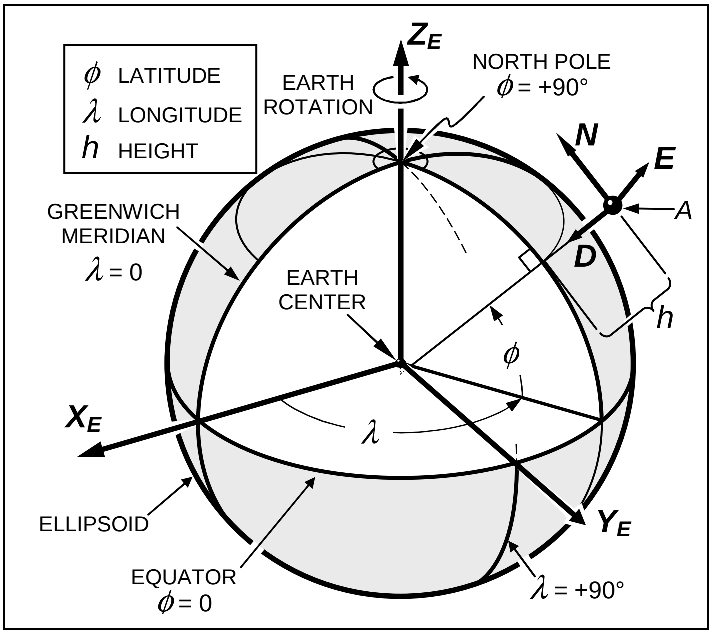


Figure Z.4-2. Earth-Fixed and Local-Level Coordinate Systems. The location of a point, *A*, can be specified with geodetic coordinates using a longitude relative to the Greenwich meridian and a latitude relative to the equatorial plane with a height above the ellipsoid (or other surface) at that location. Alternatively, the point's position can be defined by its geocentric coordinates relative to the earth's center using the shown positive axes—X_E extending through the Greenwich meridian at the equator; Z_E parallel to the earth's axis of rotation and pointing through the north pole; and Y_E completing the right-handed, orthogonal system by extending through the 90° east meridian at the equator. A rectangular, local-level system can also be defined at a point. This location-specific system is illustrated here for point *A* using the north (*N*), east (*E*), and down (*D*) positive axes.

The geodetic and geocentric coordinate systems are defined in *STDI-0002, Appendix E: ASDE 2.1*. The discussion there includes the definitions of the potential elevation (or altitude) datums and provides the definition for the local-level coordinate system. This section summarizes these definitions as they apply to SENSRB.

Within SENSRB, geospatial positions are referenced to an earth-fixed system, with coordinates typically provided in a geodetic system using latitude, longitude, and height values. These are based on the ellipsoid specified as the geodetic reference system (GEODETIC_SYSTEM, index 01g)—currently the WGS-84 ellipsoid.[1] For compatibility with *MISB EG 0801* and with orbital systems, the sensor location can alternatively be reported in Cartesian, geocentric coordinates relative to the center of the WGS-84 ellipsoid. Both coordinate types are illustrated in Figure Z.4-2. The figure also provides an example of a location-specific, local-level coordinate system for point *A*.

SENSRB also allows for the sensor attitude and velocity components to be stored relative to either the local-level or geocentric coordinate system. The system used for reporting the sensor location and referencing its attitude and velocity is specified by the geodetic coordinate type (GEODTIC_TYPE, index 01h)—see Z.5.1.3 Datum and Unit Declarations.

Z.4.6.1.1. Geodetic Coordinates

Geodetic coordinates in SENSRB provide the latitude and longitude expressed in decimal degrees, *regardless* of angle unit type (ANGULAR_UNIT, index 01i). The generic formats are ±DD.dddd and ±DDD.dddd, respectively. The value *must* include a sign (+ or –) to indicate hemisphere: + (North) and – (South) for latitude, and + (East) and – (West) for longitude. The latitude ranges from 0° to ±90° (-90.0000000 to +90.0000000), and the longitude ranges from 0° to ±180° (-180.0000000 to

[1] As defined in NGA's TR8350.2, *Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems*

+180.0000000). The number of digits allowed in the field will determine its potential resolution. Note that a level of precision should not be used to infer a level of accuracy. For example, just because a coordinate is provided in degrees with seven places after the decimal point—allowing for resolutions of a centimeter, it should not be assumed that the coordinates are accurate to within ±0.5 centimeter—see Z.4.3.2 Numerical Precision.

Z.4.6.1.2. Elevation and Altitude Datums

The elevation or altitude values included as part of the geodetic coordinates within SENSRB are *all* provided relative to the elevation or altitude datum specified by ELEVATION_DATUM (index 01i)—up being positive. The possible elevation datums are described in detail in *STDI-0002, Appendix E: ASDE 2.1* (Section E.2.1.2). HAE values are ***strongly*** recommended for use in SENSRB and are available using GPS. MSL (height above mean sea level) is less desirable due to potential differences in geoid models (see Section E.2.1.2). AGL (height above ground level), if used, shall be referenced to the terrain directly below the sensor (not at the image location). All subsequent elevations will be referenced to that same sensor-local horizontal plane, which passes through the terrain surface height directly below the sensor. AGL is the least desirable reference due to additional uncertainties in terrain elevation estimates. The elevation (or altitude) units of measurement are meters (m) or feet (ft) depending on LENGTH_UNIT (index 01j).

Z.4.6.1.3. Geocentric Coordinates

Geocentric coordinates are three-dimensional, Cartesian-style coordinates relative to the orthogonal axes (X_E, Y_E, and Z_E) defined in Figure Z.4-2 Depending on the value of GEODTIC_TYPE (index 01h), the sensor's three-dimensional location, attitude[1], and velocity can be specified relative to these axes. (See Z.5.1.3 Datum and Unit Declarations.)

Z.4.6.1.4. Local-Level Coordinate System

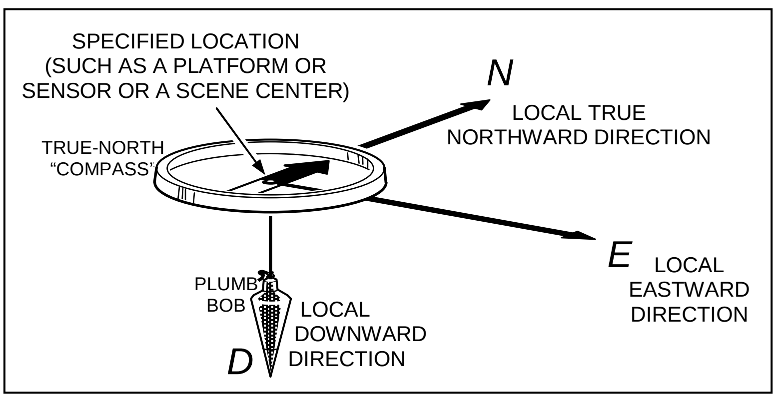


Figure Z.4-3. Conceptual Local-Level Coordinate System. The local-horizontal true north and east directions are symbolized by the true-north "compass," and the local vertical is represented by the plumb bob.

In SENSRB, the platform and sensor attitudes, as well as the sensor velocity components, can be specified relative to a local-level coordinate system—shown conceptually in Figure Z.4-3. This earth-fixed coordinate system is centered at a specific geolocation (such as the platform's navigation center)—its orientation being unique to that location.[2] The positive north (*N*) and east (*E*) axes are defined as being parallel to the local horizontal true north and east directions, respectively. The positive down (*D*) axis is parallel to the local gravitational potential vector and approximately normal to the ellipsoid. This coordi-

[1] Attitudes relative to the geocentric axes can only be specified in Modules 8 or 9 (Attitude–Unit Vectors or Attitude–Quaternions), as the Euler angles of Module 7 are defined to always be relative to the local-level (NED) coordinate system.

[2] At long ranges, this platform/sensor-local coordinate system will have a small, but potentially non-trivial, difference in orientation compared to the local-level coordinate system at the area being imaged. See "Appendix E: ASDE 2.1" (E.2.1.4) for an exaggerated example of how the local-level coordinate system's orientation changes with location.

nate system is sometimes referred to as the North-East-Down (NED) system and is the only local-level coordinate system used in SENSRB.

Z.4.6.2. Platform Coordinate System

The platform coordinate system is fixed to the sensor platform body, with its origin typically at the platform's navigation center or center of gravity. In SENSRB, any position offsets (indices 06d to 06f) from the sensor or platform position (indices 06a to 06c) to the sensor's perspective center location are reported relative to the platform coordinate system's axes (see Z.5.6.2 Sensor Position Offsets). The attitude of the platform coordinate system—and the platform itself—is defined relative to the local-level (NED) coordinate system as described in Z.5.7.2 Platform Euler Angles.

The actual relationship between the platform body and its coordinate system is not necessarily relevant to the use of this intermediate coordinate system. Nonetheless, an appropriate relationship can lend itself to the intuitive use of the platform Euler angle names (PLATFORM_HEADING, PLATFORM_PITCH, and PLATFORM_ ROLL; indices 07f to 07h—see Z.5.7.2 Platform Euler Angles). Some typical orientations are shown in Figure Z.4-4. The generic airborne platform coordinate system shown in the figure is defined below. This definition can easily be extended to the other platform types.

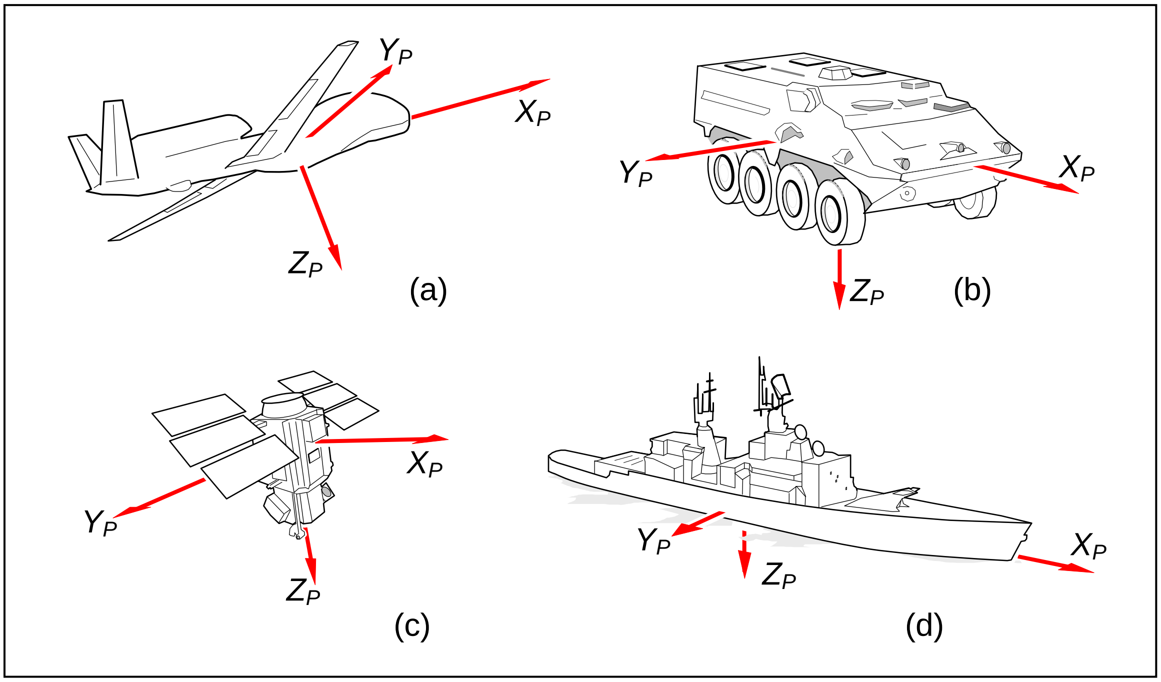


Figure Z.4-4. Typical Platform Coordinate Systems. Shown are typical body coordinate systems for (a) an aircraft, (b) a land vehicle, (c) a satellite, and (d) a maritime vessel.

The positive X_P axis (sometimes called the platform's roll axis) is aligned with the aircraft's nose so as to be approximately level during level flight. The Y_P axis (sometimes called the platform's pitch axis) is in the same level plane with its positive axis pointing in the direction of the right wing. The Z_P axis (sometimes called the platform's yaw axis) completes the orthogonal system with its positive axis extending through the aircraft's belly. Using this convention, the platform coordinate system would be aligned with the local-level (NED) system during straight-and-level, northward flight (in the absence of any cross-flight forces, which would cause crabbing of the aircraft).

Z.4.6.3. Sensor Coordinate System

The sensor coordinate system is associated with the physical sensor and initial image arrays (see Z.4.5 Array Types) in such a way to define the position and attitude of what would be effectively a single-frame (focal plane array) sensor. Because such a sensor does not always physically exist as part of the sensing system, this essential coordinate system may only be conceptual in nature.

The origin of the coordinate system is located at the sensor's instantaneous perspective center. The axes are oriented such that the Z_S axis is aligned with the instantaneous, effective optical axis (positive toward the object being imaged); the X_S axis is parallel to what will be the initial image array's first row (positive in the direction of increasing column indices); and the Y_S axis is parallel to what will be the initial image array's first column (positive in the direction of increasing row indices).

A pictorial representation of this coordinate system is shown for a notional framing sensor in Figure Z.4-5.

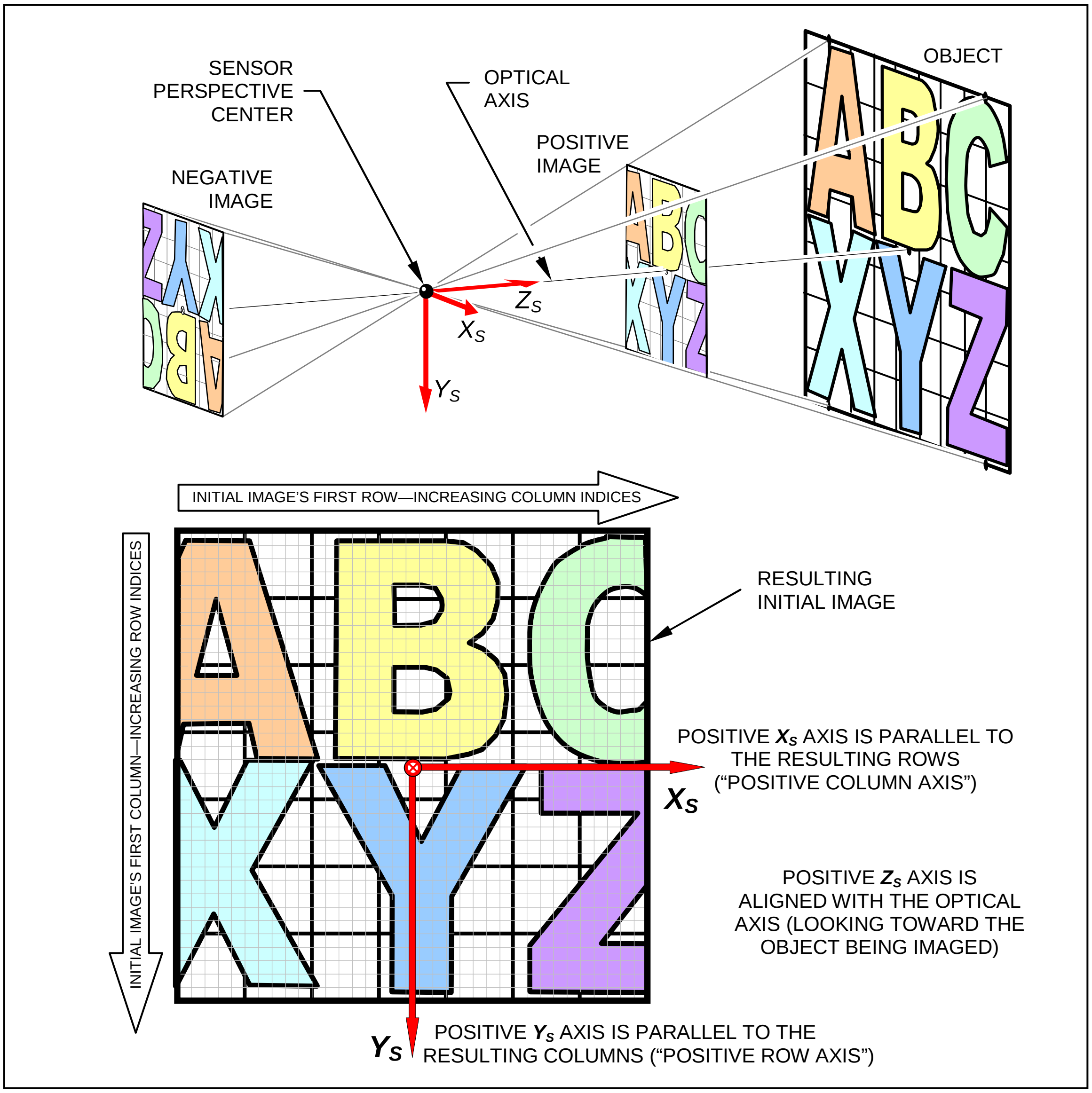


Figure Z.4-5. Sensor Coordinate System for a Notional Framing Sensor. Shown (top) is a notional imaging scenario (with both the positive and negative images included). The X_S and Y_S axes' orientation is determined in part by the layout of the resulting initial image array (bottom).

Alternative coordinate system possibilities for a pushbroom sensor are shown in Figure Z.4-6. Because the sensor coordinate system's X_S and Y_S axes are parallel to detectors that will contribute to the initial image's rows and columns, respectively; the orientation of the sensor coordinate system depends on the way the initial image array is stored. This is illustrated for a case where the object is imaged as shown at the top right of the figure. In this case, the collected lines of imagery form columns of the initial image array, with the objects farthest from the sensor appearing along the first (top) row of the image. For this case, the sensor coordinate system is as depicted (top left) regardless of the direction of sensor motion (into or out of the page). Conversely, if the collected imagery lines form the rows of the initial image array, the sensor coordinate system will be different depending on which line makes up the first (top) row of the initial image. Two cases are shown in the bottom right of the figure, each with its resulting sensor coordinate system orientation illustrated.

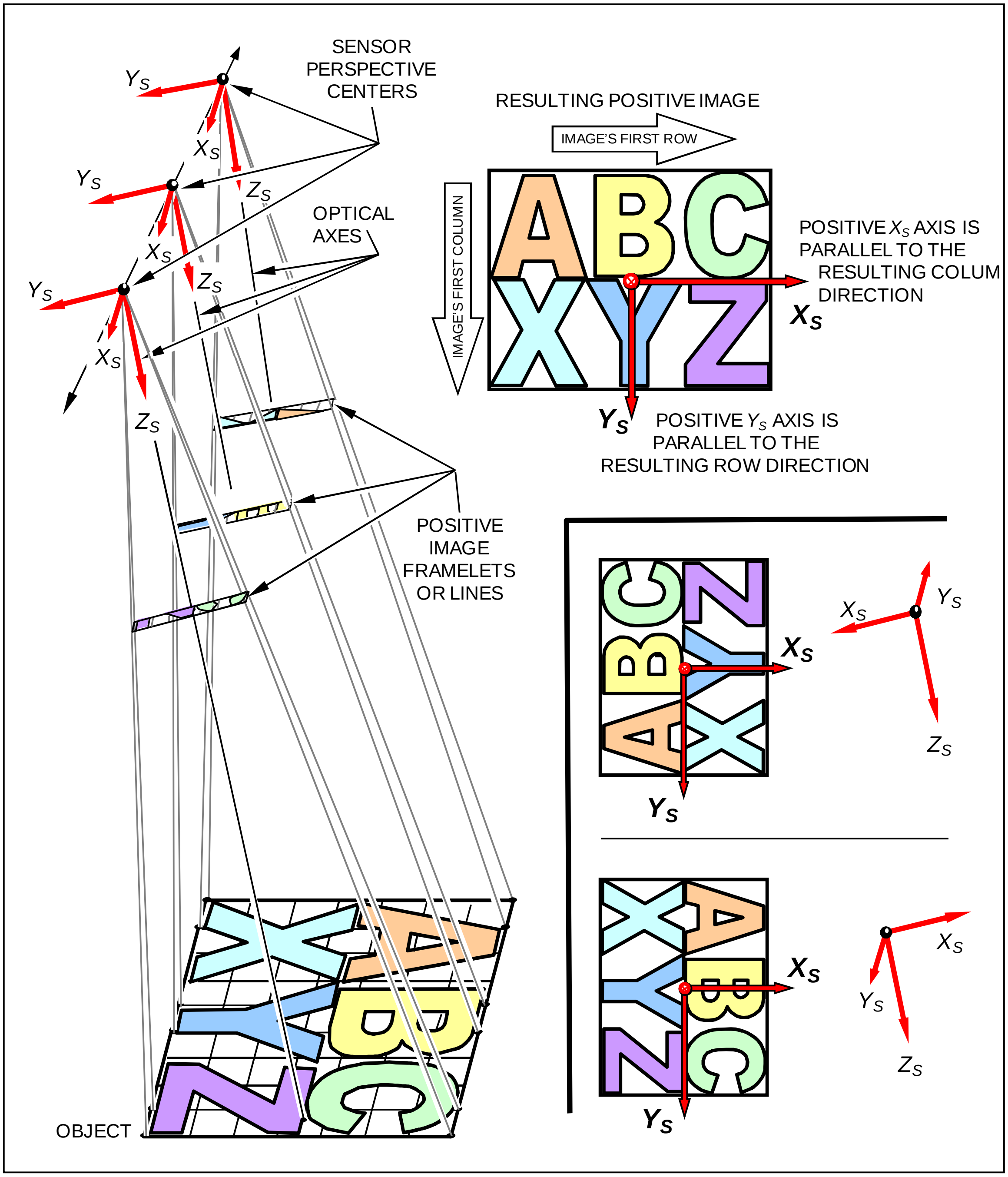


Figure Z.4-6. Sensor Coordinate Systems for a Notional Pushbroom Sensor. Shown (left) is a notional imaging scenario. Three likely initial image arrays are depicted on the right. The corresponding sensor coordinate system orientations are shown for each of the three cases.

Z.4.6.4. Image Coordinate System

This image coordinate system, illustrated in Figure Z.4-7, is used in traditional photogrammetric applications. The sensor calibration parameters, if included, with their respective equations are based on the image coordinate system. The attitude unit vectors module (Module 8) provides the attitude of this coordinate system (not the sensor coordinate system). Like the sensor coordinate system, the image coordinate system has its origin at the sensor perspective center. Its axes (X_I, Y_I, and Z_I) are aligned respectively (1) parallel with the image's (or framelet's) first row (positive in the direction of increasing column indices), (2) parallel with the first column (positive in the direction of *decreasing* row indices), and (3) with the sensor's effectual, instantaneous optical axes (positive in the direction *away* from the object being imaged).

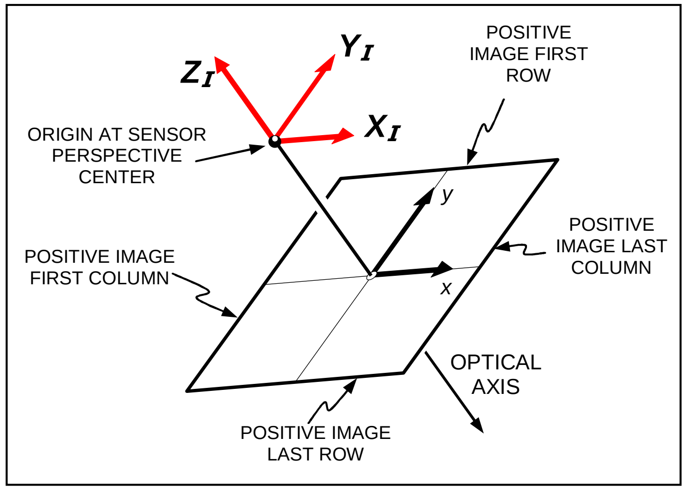


Figure Z.4-7. Image Coordinate System.

Z.4.6.5. Image Array Common Coordinate System

The image array common coordinate system is defined in *MIL-STD-2500C*, section 4.7. This two-dimensional coordinate system is used to define locations within an image array, including non-integer values. It shall be applied separately to both the initial image and the NITF-stored arrays, as these two arrays may be different—see Z.4.5.4 Array Type Relationships. The image array common coordinate system can also be applied to the physical sensor array—see Z.5.3 Sensor Calibration Data Module. An example of the image array common coordinate system is illustrated in Figure Z.4-8 for a given image array.

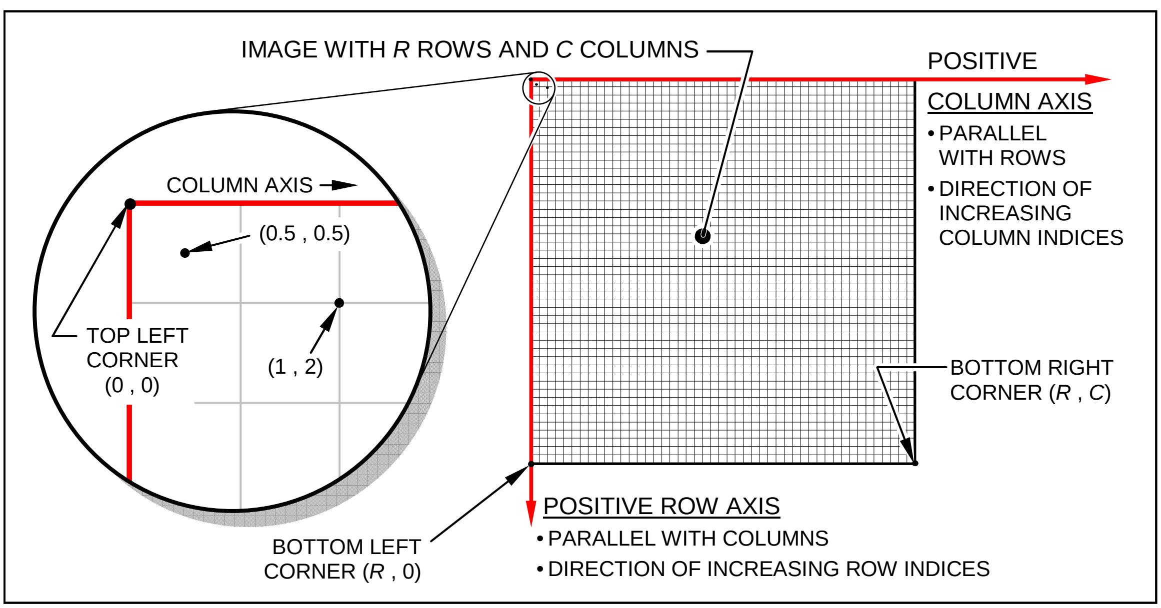


Figure Z.4-8. Image Array Common Coordinate System. Shown are the two axes used by this system. Illustrative points (on an image array containing *R* rows and *C* columns) are also indicated with their coordinates.

Note: Although the common coordinate system as defined in *MIL-STD-2500C* does not account for negative coordinate values, negative coordinate values are sometimes allowed in SENSRB. This use of negative values allows for references to be made to pixel locations outside of the valid image.

Z.4.6.6. Pixel Coordinate System

The pixel coordinate system, illustrated in Figure Z.4-9, is used to reference an individual row, column, or pixel in an image. Generally, when this documentation refers to a "row" or a "column" within an image, that reference is with respect to the pixel coordinate system.

The pixel coordinate system is always addressed in integer values. The pixel located in the very first column of the very first row is pixel (1, 1). For an image composed of "m" rows and "n" columns, (in this figure, 8 rows by 12 columns) the last pixel in an image would be pixel location (m, n), or location (8, 12) in this case.

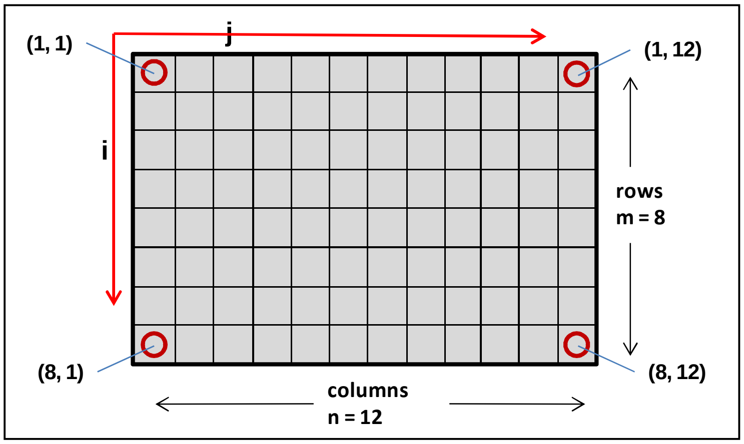


Figure Z.4-9. Pixel Coordinate System.

It is possible to convert a location in image array space to a pixel location in pixel space, that is, to find the pixel that contains the image space location. That equation is:

$$\text{Pixel}(i, j) = \lceil \text{Coord}(x, y) \rceil$$

where ⌈ ⌉ represents the ceiling function, i.e. rounding **up** to an integer. For example, the image array coordinate location (123.33, 150.0) would be located in pixel (124, 150).

In the reverse conversion, from pixel space to image array space, one can generally only compute the image array coordinate of the center of a specific pixel as:

$$\text{Coord}(x, y) = \text{Pixel}(i, j) - (0.5, 0.5)$$

For example, the image array space coordinate for pixel (124, 150) is (123.5, 149.5), that is, the center of the area covered by the pixel. Another way to look at this is that pixel (124, 150) covers the image array coordinates from (123.0, 149.0) to (124.0, 150.0).

Finally, note that the pixel coordinate system is *not* the same as the typical computer-screen graphics approach used by programmers. Computer screen addressing is generally a left-hand coordinate system with the first index moving from left to right across the screen, "x", and the second coordinate, "y", moving down the screen, which is the transpose of the pixel coordinate system.

Z.4.7. STATIC AND DYNAMIC DATA

Metadata stored in a SENSRB TRE can be thought of as static or dynamic. In the case of static data, the values are not expected to change before, after, or during the collection of the imagery. On the other hand, dynamic data is expected to change over time. As examples, the PLATFORM (index 01c) value is static as it is not expected to change during a mission. Conversely, sensor position data (indices 06a through 06c) is likely to change before, after, and possibly during the image collection. Such dynamic data can be stored in the SENSRB TRE using looping fields and/or multiple TREs. Looping fields and the use of multiple TREs are described in the following sections.

Z.4.8. LOOPING FIELDS CONCEPT

Looping fields allow for a variable-length set of values to be stored within those fields. They are preceded by a field containing an integer count value. The subsequent looping fields are then repeated (looped) the number of times specified by that count value. In SENSRB, a looping field loop is often “nested” within another outer loop; see Figure Z.4-10.

Looping fields are employed in Modules 11 through 15. Guidance for the use and interpretation of the looping fields in each module is provided in its respective module-specific implementation notes (Z.5.11 through Z.5.15).

Fields within a loop can be referenced in other fields by their index (Z.4.8.2 Looping Field Indexing). The following subsection demonstrates the use of looping fields with a generalized example. The example is followed with instructions for referencing the fields by index and resolving conflicts with their values.

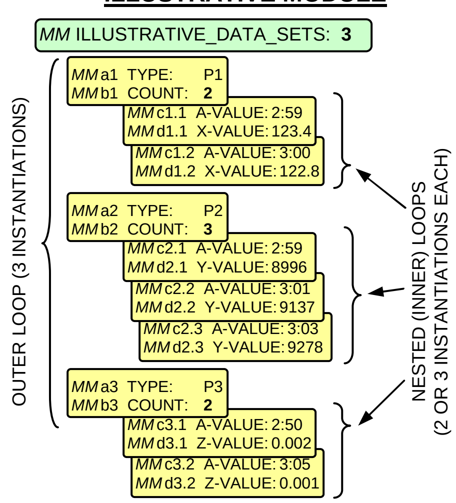
Z.4.8.1. Generalized Looping Field Example

Figure Z.4-10 provides a notional example of a set of nested looping fields making up an illustrative module, *MM*. For this example, values are available for three dynamic parameters, as shown in the upper left. The diagram on the lower left shows schematically how these values might be stored in SENSRB. On the right of the figure, the same information is provided in tabular form.

ILLUSTRATIVE DATA

PARAMETER	ASSOCIATED VALUES	
P1	2:59	123.4
	3:00	122.8
P2	2:59	8996
	3:01	9137
	3:03	9278
P3	2:50	0.002
	3:05	0.001

SCHEMATIC VIEW OF ILLUSTRATIVE MODULE



TABULAR VIEW OF ILLUSTRATIVE MODULE

INDEX	NAME	VALUE
MM	Number of Parameter Data Sets	**3**
MM a1	Name of 1ST Parameter	P1
MM b1	Number of sets for 1ST Parameter	**2**
MM c1.1	1ST A-Value for 1ST Parameter	2:59
MM d1.1	1ST X-Value for 1ST Parameter	123.4
MM c1.2	2ND A-Value for 1ST Parameter	3:00
MM d1.2	2ND X-Value for 1ST Parameter	122.8
MM a2	Name of 2ND Parameter	P2
MM b2	Number of Sets for 2ND Parameter	**3**
MM c2.1	1ST A-Value for 2ND Parameter	2:59
MM d2.1	1ST Y-Value for 2ND Parameter	8996
MM c2.2	2ND A-Value for 2ND Parameter	3:01
MM d2.2	2ND Y-Value for 2ND Parameter	9137
MM c2.3	3RD A-Value for 2ND Parameter	3:03
MM d2.3	3RD Y-Value for 2ND Parameter	9278
MM a3	Name of 3RD Parameter	P3
MM b3	Number of Sets for 3RD Parameter	**2**
MM c3.1	1ST A-Value for 3RD Parameter	2:50
MM d3.1	1ST Z-Value for 3RD Parameter	0.002
MM c3.2	2ND A-Value for 3RD Parameter	3:05
MM d3.2	2ND Z-Value for 3RD Parameter	0.001

Figure Z.4-10. Illustrative Nested Looping Field Example. The parameter data (top left) might be stored in the notional Module *MM* as shown in the schematic (bottom left) and tabular (right) views.

Z.4.8.2. Looping Field Indexing

As illustrated in Figure Z.4-10, integer counters shall be appended onto looping field indices to specify which instantiation of the field is being referenced. If used in the SENSRB TRE, these counters immediately follow the three-character base index (see Z.2.2.2 Field References), without space or other leading characters. For a non-nested loop, a single counter value is sufficient and shall be used. For a nested loop, the “outer” and “inner” loop counters both shall be appended (in that order), separated with a decimal point.

For example, the value for the 14^{th} occurrence (inner loop) of the 7^{th} time-stamped parameter (outer loop) in Module 12 might be referenced as 12d7.14, 12d07.014, or 12d07.0014.[1] In the case of an outer loop index, such as TIME_STAMP_TYPE (index 12a); the 7^{th} occurrence would be referenced with the index 12a7 or 12a07.

Z.4.8.3. Value Precedence

Looping field values are expected to be populated in roughly chronological order. If multiple, conflicting values exist for a parameter; the value occurring last in the looping field will take precedence over any previous conflicting values at the same time. See Z.4.10 Dynamic Value Interpolations for additional considerations.

Z.4.9. GENERAL SENSRB USAGE

As described in the introduction, sensor parameters can be stored in a single SENSRB TRE or in multiple SENSRB TREs. The choice of how to use SENSRB and implement its individual modules may depend on the collection scenario involved. This subsection discusses the general usage of the SENSRB TRE and introduces three typical scenarios types and some implementation nuances associated with each.

Z.4.9.1. TRE Placement

The preferred location for the SENSRB TRE is in the image subheader. If the size of a SENSRB TRE causes the image subheader’s size limitation to be exceeded, one or more SENSRB TREs can be placed in the TRE-overflow data extension segment. (See *MIL-STD-2500C*, section 5.8.) In a file with multiple image segments it is expected that each TRE will be placed in the appropriate image subheader—clearly associating it with the correct image segment.

It is assumed that, if they exist, multiple SENSRB TREs will be stored in chronological order; thus if there are conflicting values between multiple TREs, the later TRE will take precedence over the former. See Z.4.10 Dynamic Value Interpolations for additional considerations. Furthermore, if one or more SENSRB TREs are generated with new or refined parameters (based on adjustments or discoveries made after the initial NITF file generation), these updated TREs will be stored after the previously generated TREs to take precedence over those outdated occurrences. (See Z.5.1.5 Image Parameter Post-Collection Adjustments.)

Z.4.9.2. TRE Size Limitations

Although a full implementation of all the possible SENSRB looping fields would result in a much larger TRE, the controlled extension data in a single TRE (the data following CETAG and CEL) cannot exceed 99,985 bytes. Furthermore, the size of a SENSRB TRE shall not cause the image subheader’s or TRE-overflow data extension segment’s size limitation to be exceeded. (See *MIL-STD-2500C*, section 5.8.)

Z.4.9.3. Multiple SENSRB TREs and Module Existence

The metadata within the SENSRB modules can generally be categorized as either static (unchanging) or dynamic (changing). When multiple SENSRB TREs are used to define a single image-segment collection, the following guidance applies to the existence of the modules containing the dynamic or static data.[2]

---

[1] An appropriate number of leading zeros may be used with each appended counter to keep uniform the length of indices associated with a given field. Furthermore, spaces can *follow* a composite index as described for alphanumeric data in Z.2.2.4 Field Formats.

[2] Basic guidance regarding the existence of SENSRB modules is given in Z.2.1 Data Modules.

Dynamic Data. By requirement, each and every instantiation of SENSRB must include the Reference Time or Pixel Location (Module 5) and the Sensor Position Data (Module 6)—as this data potentially changes across multiple TREs. Additionally, the attitude data (Modules 7, 8, or 9) and Sensor Velocity Data (Module 10) may also include dynamic data. These modules would then be included in each (or many) of the multiple TREs to update the changing field values. See Z.4.10 Dynamic Value Interpolations.

The Time-Stamped (Module 12) and Pixel-Referenced (Module 13) Data Sets allow dynamic data to be recorded within a single SENSRB TRE. These modules may exist across multiple TREs if needed, but such usage is not typically expected (see Z.7 Recommended Uniform Implementations).

Static Data. By requirement, the first instantiation of the SENSRB TRE associated with an image segment, must include the General Data Module (Module 1). Other modules containing typically static data—such as Sensor Array Data (Module 2), Sensor Calibration Data (Module 3), Image Formation Data (Module 4), Point Data Sets (Module 11), Uncertainty Data (Module 14), and Additional Parameter Data (Module 15)—*if they are to be provided at all, shall also be included in the first instantiation of SENSRB*.

These generally static metadata modules (1-4, 11, 14, and 15) typically would not be included in subsequent instantiations of the TRE for that image segment. If, however, field values associated with one of these modules does change during a collection, the new value shall be provided. This can be done by including the appropriate module in another instantiation of the TRE or by including the value in the looping fields of Modules 12 or 13 (see Z.4.8 Looping Fields Concept and Z.5.12 Time-Stamped Data Module or Z.5.13 Pixel-Referenced Data Module). The new value will be assumed to apply from that reference time or pixel location forward, unless changed again. See Z.4.10 Dynamic Value Interpolations.

In other words, a field "inherits" the corresponding value from an earlier TRE unless that field is overridden with a new value in a subsequent TRE, which is inherited by that field in even later TREs.

Note: The content level value (CONTENT_LEVEL, index 01f) in Module 1 provides an indication of which modules exist in the SENSRB TRE (see Z.5.1.2 Application-Required Content Level). In the case of multiple SENSRB TREs, this single encoded value applies to the first SENSRB TRE, which—according to the above instructions—is representative of the aggregation of all the TREs associated with the particular image segment.

Z.4.9.4. Typical Collection Scenarios

Although SENSRB can be implemented in many ways, the most typical implementations of the TRE can be demonstrated by three generalized collection scenarios—those for: framing (focal plane array) sensors, still images derived from motion imagery sensors, and scanning (such as pushbroom, whiskbroom, or step-stare) sensors. These are each described in the following subsections.

Z.4.9.4.1. Instantaneous Collections (Framing Sensors)

For the framing sensor scenario, the assumption is that all image pixels and the associated parameter values are collected at approximately the same instant in time. Such a collection will typically only necessitate a single SENSRB TRE for a given image segment. This single extension will always include Modules 1, 5, and 6. The inclusion of additional modules depends on the availability and applicability of the data associated with those modules. Figure Z.4-11 provides schematic representations of the TRE for two differing instantaneous collection implementations—full and minimal.

In both instances, the values of Module 1 are likely to be fixed for a given platform/sensor combination. See Z.4.7. Static and Dynamic Data.

In the full implementation (Figure Z.4-11, left), data is available for many of the SENSRB parameters. Modules providing the sensor attitude, various uncertainties, the sensor calibration parameters, and other data might be included in the TRE. On the other hand, since the imagery and the parameter values are all available for the single time instant of interest, it is not anticipated that the time- and pixel-referencing modules (Modules 12 and 13) will contain any data for a framing sensor collection.

In the minimal implementation (Figure Z.4-11, right), only the required flag fields would be present for each of the optional modules. Such an implementation is undesirable for most exploitation applications, but it is allowed and thereby accommodates sensor systems incapable of providing the additional data. In this minimal case, the implementation of the SENSRB TRE is further simplified in that Modules 2 through 4 would be represented by three characters “NNN” indicating that those modules do not otherwise exist. Similarly, Modules 7 through 15 could be represented by the characters “NNNN000000000000” indicating that modules 7, 8, 9, and 10 do not exist (“N”), and the loop counts for modules 11, 12, 13, 14, and 15 are all zero.

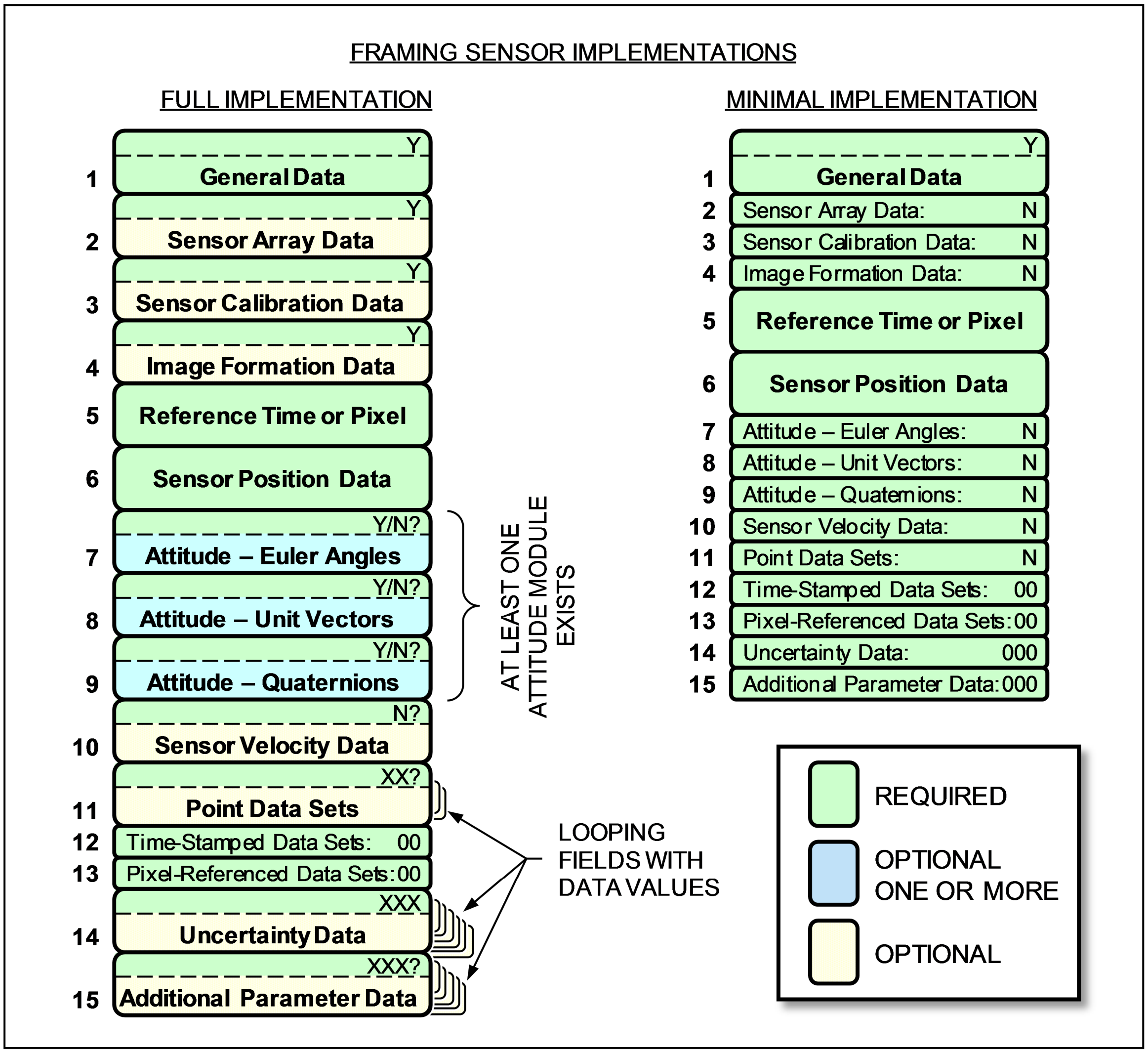


Figure Z.4-11. Framing Sensor Implementations. In the full implementation (left), most modules are populated. Data in Modules 10, 11, and 15 are not necessarily provided. In the undesirable, but allowed, minimal implementation (right); only the required fields are provided.

Although the full implementation is desirable for interoperability and full geometric capability, any implementation using the minimum content level or higher conforms to this specification. The existence of the various modules is indicated to the SENSRB user in the content level field value (CONTENT_LEVEL, index 01f)—see Z.5.1.2 Application-Required Content Level.

Z.4.9.4.2. Asynchronous Data Collections (Motion Imagery Sensors)

For still images derived from motion imagery, the assumption is that the image pixels are all collected at approximately the same instant (from a single or a few frames). However, the values for various parameters associated with the collection are likely to be available at disparate times. This asynchronous data can be stored with its respective timing information in SENSRB's Module 12 (Time-Stamped Data Sets). The time-stamped data sets allow multiple values to be stored for each of the dynamic parameters. These values can represent instances both before and after the image collection time—allowing the user to estimate the parameter values at the instant of collection. See Z.4.10 Dynamic Value Interpolations.

This asynchronous data scenario could apply to various systems—each being able to provide differing amounts of sensor metadata. The optional modules would be populated accordingly, as suggested previously in Z.4.9.4.1 Instantaneous Collections (Framing Sensors). Again, since all the image pixels are collected at the same instant, it is not expected that any parameters would be referenced to individual pixels or lines; and Module 13 (Pixel-Referenced Data Sets) would not be populated.

Figure Z.4-12 (left side) represents schematically a SENSRB TRE for an asynchronous data collection with many of the optional modules existing.

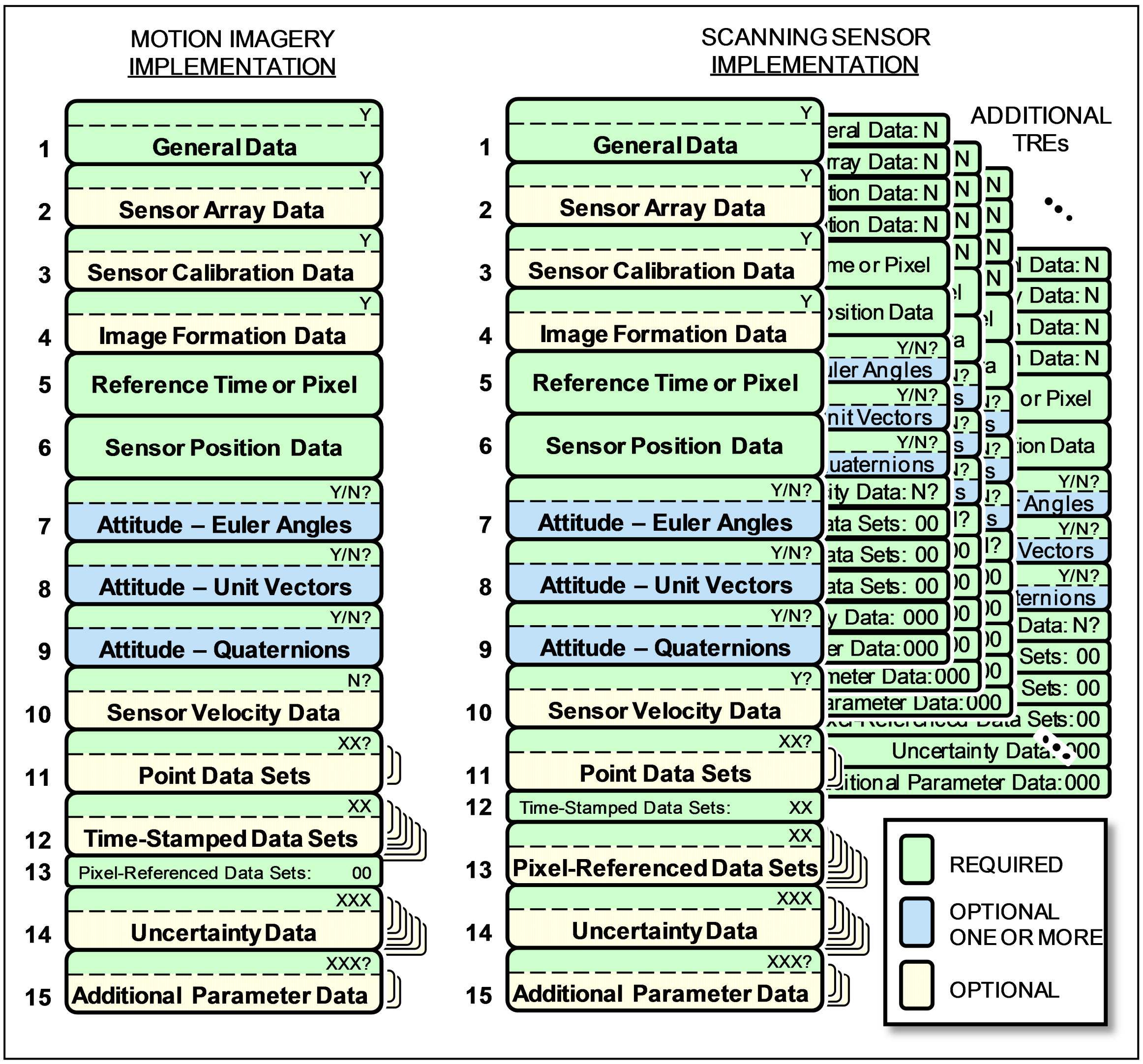


Figure Z.4-12. Motion Imagery and Scanning Sensor Implementations. For illustrative purposes, full implementations—with many of the optional modules existing—are shown for both collection scenarios. For the scanning sensor, both multiple TREs and populated looping fields in Module 13 are shown. In real application, only one method or the other would probably be used to store the dynamic data values.

Z.4.9.4.3. Scanning Collections (Scanning Sensors)

For a scanning sensor, such as a pushbroom or whiskbroom system, the assumption is that the image pixels are collected in lines or framelets over time and then recorded as a single image. Parameter values that change over the course of the collection (such as sensor position and attitude) are stored for multiple instances, at some time or line/framelet interval. These values are referenced to those specific image pixels (or times), either using Modules 13 (or 12) or—as shown in Figure Z.4-12 (right side)—using multiple instantiations of SENSRB. In the illustrated case with multiple SENSRB TREs, the contents of several modules are provided only in the first instantiation of SENSRB (See Z.4.9.3 Multiple SENSRB TREs and Module Existence). Subsequent instantiations of SENSRB provide only the needed dynamic values and the required flags.

The flexibility to use multiple instantiations of the SENSRB TRE, some of which might be stored in the TRE-OVERFLOW Data Extension Segment, overcomes any potential problems due to the size restriction on a single TRE. (See Z.4.9.2 TRE Size Limitations.)

Z.4.10 DYNAMIC VALUE INTERPOLATIONS

As introduced previously, SENSRB allows multiple field values to be stored for a dynamic (changing) parameter. These values can be stored either in looping fields (as Time-Stamped and/or Pixel-Referenced Data Sets—Modules 12 and/or 13, respectively) or across multiple SENSRB TREs. It is expected that these values will be stored in chronological order. It is also expected that the SENSRB user may interpolate between these values to determine appropriate parameter values for non-specified times or pixel locations. If possible, sufficient data should be provided to allow reasonable value approximations with linear interpolations.

Figure Z.4-13 provides four illustrative cases where dynamic parameter values (left column) are represented with discrete values (middle column). Possible interpretations are shown in the figure's right column. For the shown cases, the linear interpolation produces adequate value approximations. The more complex cubic spline may be needed in some cases, but it should be used with care—especially if discontinuities may exist in the data.

Discontinuities in dynamic values (such as the step or saw-tooth functions shown in the bottom two rows of Figure Z.4-13) can be represented by providing two separate values, each referenced to the same time or pixel location. In such cases, the ordering of the looping fields or the TREs is especially significant (see Z.4.8.3 Value Precedence or Z.4.9.1 TRE Placement, respectively). Furthermore, special care must be used in the interpolation when such discontinuities are represented.

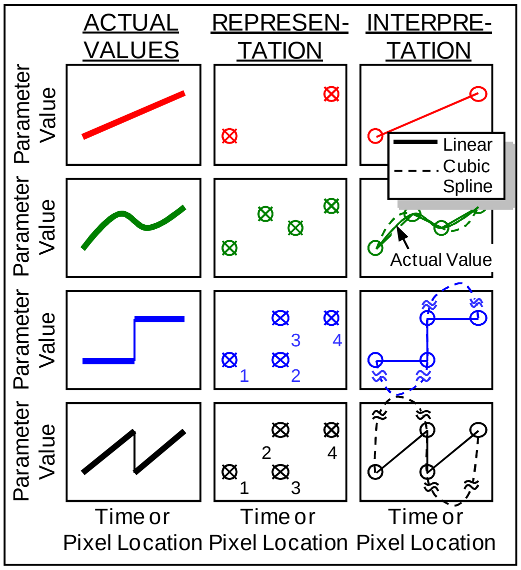


Figure Z.4-13. Dynamic Parameter Value Representations and Interpolated Interpretations.

Z.5. MODULE-SPECIFIC IMPLEMENTATION NOTES

This section provides guidance and clarification specific to each of the SENSRB modules. This guidance supplements but does not supplant the information in Z.3 Field Specifications. The *SENSRB Data Conversion Tool* (see Z.6.4 SENSRB Data Conversion Tool) provides an interactive reference to further aid in the interpretation of this guidance as it applies to each of the individual models.

Note: The subsection numbering in this section corresponds with the module numbers.

Z.5.1. GENERAL DATA MODULE (MODULE 01)

The General Data Module (Module 01) must exist in the first SENSRB TRE associated with any image segment (see Z.2.1 Data Modules). It contains data that typically does not change throughout a collection (see Static Data in Z.4.9.3 Multiple SENSRB TREs and Module Existence). The values in this module aid in the discovery of appropriate imagery for exploitation. They also define references for the values in subsequent modules and/or SENSRB TREs associated with that image segment.

Field values in the General Data Module identify and characterize the sensor and platform. They indicate which modules exist in association with the image segment and declare the datums and units used for those subsequent modules. They also establish the imaging date and time and state how many times and when the subsequent data has been adjusted or regenerated.

Z.5.1.1. Sensor and Platform Identification and Characterization

Sensor and Platform Names (SENSOR and PLATFORM, indices 01a and 01c) are NTB-registered, common names. See Z.4.1 NTB-Registered Field Values. Additional and more specific information regarding the sensor and platform can be indicated in their respective URIs.

Sensor and Platform URIs (SENSOR_URI and PLATFORM_URI, indices 01b and 01d) provide uniform resource identifiers using any applicable and acceptable scheme. These might include a sensor's serial number or an aircraft's tail number, but an "internet address" (uniform-resource-locator scheme) where additional and more specific information can be accessed is desirable. Internet Protocol version 6 addresses (eight groups of hexadecimal quartets separated by colons) can be accommodated in these fields—if either the colons or null quartets are omitted. Metadata values in the latest SENSRB TRE generation are assumed to take precedence over any conflicting values found at the provided URI.

Operation Domain (OPERATION_DOMAIN, index 01e) provides the user with an indication of the sensor's vantage point or perspective. This NTB-registered value (Z.4.1 NTB-Registered Field Values) enables queries for imagery with the appropriate vantage point for specific uses.

Z.5.1.2. Application-Required Content Level

This appendix does not attempt to define which modules must exist and what quality of data is required for specific applications. However, the content level field value (CONTENT_LEVEL, index 01f) does give some indication of which modules are available for the image segment. Thus, content level might be used to query archives for imagery with certain application-required levels of metadata.

To determine or interpret the content level value, one must refer to Table Z.5-1, which lists the modules required for the various values. Increasing content levels imply suitability for additional exploitation. For example, a content level "1" implementation of SENSRB indicates the presence of image temporal data and sensor position data; thus, a SENSRB user may infer this minimal content to be sufficient only for situational awareness. On the other hand, a content level "6" implementation indicates the presence of image temporal data, sensor position data, sensor array data, image formation data, sensor or image attitude data, and uncertainties (also synchronization and velocity data, depending on the sensor method); therefore, a user may infer this increased content to be sufficient for geopositioning.

Table Z.5.1-1. Minimum Data Requirements for SENSRB TRE Content Level. "X" indicates that data is required.

	Content Level Value									
SENSRB Content (Applicable Module)	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
Image Temporal Data (1 & 5)	X	X	X	X	X	X	X	X	X	X
Sensor Position Data (6)	X	X	X	X	X	X	X	X	X	X
Sensor Array & Image Formation Data (2 & 4)			X	X	X	X	X	X	X	X
Sensor or Image Attitude Data (7, 8, or 9)					X	X	X	X	X	X
Metadata Synchronization[a] (12 or 13)					X[a]	X[a]	X[a]	X[a]	X[a]	X[a]
Sensor Velocity Data[a] (10)					X[a]	X[a]	X[a]	X[a]	X[a]	X[a]
Uncertainties (14)							X	X	X	X
Sensor Calibration Data (3)									X	X
Image Geospatial Data (11)		X		X		X		X		X

[a] Modules 10 and (12 or 13) are required for content level 4 and higher but only for multiple frame, pushbroom, or whiskbroom imaging methods (METHOD, index 04a).

The SENSRB implementer has a significant responsibility to understand the limits of the provided metadata and to appropriately characterize the data with the CONTENT_LEVEL field value to reflect those limits. This proper use of the CONTENT_LEVEL field increases the confidence with which a user can infer the data's exploitability. As an example, uncertainty estimates for geopositioning typically require uncertainty values for several sensor parameters. Providing only some of these values may not support the user's exploitation needs. In such a case, indicating the presence of the uncertainty data (with a "6" or higher content level value) may not be in the user's best interest. Considering the inherent challenge in appropriately characterizing content sufficiency, the SENSRB implementer is encouraged to understand the various exploitation requirements and exercise good judgment in assigning a content level. In the future, this assignment may become more obvious as data requirements are documented and standardized across the imagery exploitation community.

Z.5.1.3. Datum and Unit Declarations

Geodetic Reference System (GEODETIC_SYSTEM, index 01g) explicitly declares the geodetic system (such as WGS-84) used for all subsequent geospatial coordinates in the SENSRB TREs associated with the image segment. See Z.4.6.1 Geospatial Coordinate Systems.

Geodetic Coordinate Type (GEODETIC_TYPE, index 01h) defines whether the sensor's location, attitude,[1] and velocity are specified relative to the geodetic (and local-level) or geocentric coordinate systems (see Z.4.6.1 Geospatial Coordinate Systems). Sections Z.5.6 Sensor Position Data Module through Z.5.10 Sensor Velocity Data Module include guidance on how the geodetic coordinate type affects those respective modules.

Elevation and Altitude Datum (ELEVATION_DATUM, index 01i) declares from which surface geodetic heights in SENSRB are measured. (See Z.4.6.1.2 Elevation and Altitude Datums.) Height above ellipsoid (HAE) is preferred.

Note: Even when a geocentric coordinate type is defined in GEODETIC_TYPE, the declared elevation datum still applies to any elevations in the Point Data Sets Module (P_ELEVATION_NNN, index 11g).

Length and Angular Units (LENGTH_UNIT and ANGULAR_UNIT, indices 01j and 01k) indicate which unit systems will be used in the subsequent modules. The exception being that geodetic coordinates are always provided in decimal degrees. (See Z.4.4 Measurement Units.) The metric system (SI) and angles in degrees (DEG) are preferred.

[1] The platform (and the non-platform-relative sensor) Euler angles of Module 7 are always defined to be relative to the local-level (NED) coordinate system, regardless of GEODETIC_TYPE (index 01h); thus, GEODETIC_TYPE only affects attitudes specified in Modules 8 or 9 (Attitude–Unit Vectors or Attitude–Quaternions).

Z.5.1.4. Imaging Times and Dates

General guidance regarding the use of times and dates in SENSRB is given in Z.4.2 Times and Dates. The imaging start and end dates and times (START_DATE, START_TIME, END_DATE, and END_TIME; indices 01l to 01o) reflect absolute UTC times. These time values may help uniquely identify the imagery and will support queries for imagery collected during a specific time period. Subsequent time references (REFERENCE_TIME and TIME_STAMP_TIME_NNNN, indices 05a and 12c) are relative to START_TIME.

For near instantaneous imaging processes, the start and end times can be identical; otherwise, the end time should follow the start time. Likewise, the end date should always be the same as or later than the start date.

For systems incapable of measuring absolute times, an approximation of the start and end times shall be made by the most accurate method reasonably available. For example, an image start time might be estimated using a known absolute epoch (such as aircraft launch time) plus the most accurate available elapsed time from that epoch time to the time of the imaging.

Z.5.1.5. Image Parameter Post-Collection Adjustments

Several parameters within the SENSRB TRE can be adjusted through the use of resection or other photogrammetric methods for improved accuracy. Other field values might also be updated over time. If adjusted values are to be stored with the image file, they shall be placed in a new instantiation of the TRE. The preceding instantiations of SENSRB should also be included in the file with more recent “generations” following the older ones, as described in Z.4.9.1 TRE Placement.

Generation Count (GENERATION_COUNT, index 01p) indicates how many times the field values have been adjusted. This value is incremented by one when a new TRE is created with adjusted values. Thus, a non-zero count is used to portend the existence of adjusted parameters within the TRE; in which case, the generation date and time must be present with valid values.

Generation Date and Time (GENERATION_DATE and GENERATION_TIME, indices 01q and 01r) are provided to help distinguish between files at the same (non-zero) generation level. If the generation count value is zero, the date and time values are ignored and may be set to the unspecified indicator (hyphen filled). Otherwise, since multiple realizations of a first generation can be made from a single original file and multiple second generation files can be made from each of these; the generation date and time are used to uniquely identify the various files. To this end, the generation time shall be filled out to the full precision allowed in the field. If the time resolution allowed is not available, the remaining characters should be filled with arbitrary[1] digits to increase the likelihood of a unique TRE identification.

The generation date uses the format described in section Z.4.2 Times and Dates.

The generation time is in the HHMMSS.sss format relative to the start of the day—where HH represents hours within the day (00 to 23), MM represents minutes within the hour (00 to 59), SS.sss represents seconds (and decimal portions thereof) within the minute (00.000 to 59.999). Single digit values shall include a leading zero. For example, the times 8:05:04.321 AM and 5:09:10.234 PM would be respectively represented as “080504.321” and “170910.234”.

[1] If unknown, the integer seconds, minutes, or hours shall be respectively set to zeros. The decimal portions of a second may be random (or incrementally increasing, if that helps preserve the chronology of the adjustments).

Z.5.2. SENSOR ARRAY PARAMETERS (MODULE 02)

For the purpose of determining geolocation, the flag for module 02 must be "Y", which indicates the presence of the fields 02a thru 02i. This module must be present for content level "2" or higher. The parameters in this module are required for photogrammetric applications and contain information regarding the image collecting hardware and process.

Z.5.2.1. Detection Type

Field 02a specifies the detection spectrum of the sensor array by using NTB approved values. The Detection Type parameter is meant to capture the concepts of the number of collected frequency bands and their widths and locations within the electromagnetic spectrum. For example, a value of 'VNIR+SWIR' represents the visible, near-infrared and short-wave infrared portion of the EO spectrum. See the NTB website for the currently defined list of parameter values and their descriptions. (Z.4.1. NTB Registered Field Values)

Z.5.2.2. Sensor Detector Array Data

The sensor detector array data relates to the physical sensor array used to detect the image; see Z.4.5.1. Physical Sensor Array. To provide the photogrammetric data typically required for geopositioning, this module must contain, for each dimension (row and column), meaningful values (neither 0 nor the unspecified indicator) in either: (ROW/COLUMN_FOV—indices 02g and 02h) – or – (both ROW/COLUMN_METRIC—indices 02d and 02e—and FOCAL_LENGTH—index 02f); see Z.5.2.5 Sensor Array Relationships for further clarification of the required fields.

Z.5.2.2.1. Row and Column Detectors

The pixel count of the detector used in the "instantaneous" collection process is stored in ROW_DETECTORS and COLUMN_DETECTORS (indices 02b and 02c). These values shall correspond with the used sensor array's physical dimensions—ROW_METRIC and COLUMN_METRIC (indices 02d and 02e), respectively. See Figure Z.5.2-1 below.

> ROW_DETECTORS is the number of rows in the detector (*m* in Figure Z.5-1). (This value is equivalent to the number of detectors in each column of the detector array.)
>
> COLUMN_DETECTORS is the number of columns in the detector (*n* in Figure Z.5-1). (This value is equivalent to the number of detectors in each row of the detector array.)

Z.5.2.2.2. Row and Column Metrics

The physical size of the detector used in the "instantaneous" collection process is stored in ROW_METRIC and COLUMN_METRIC (indices 02d and 02e). These dimensions shall correspond with the used sensor array's pixel count—ROW_DETECTORS and COLUMN_DETECTORS (indices 02b and 02c), respectively. See Figure Z.5.2-1 below.

> ROW_METRIC is the physical length of the sensor array as measured along the direction of increasing row indices.
>
> COLUMN_METRIC is the physical length of the sensor array as measured along the direction of increasing column indices.

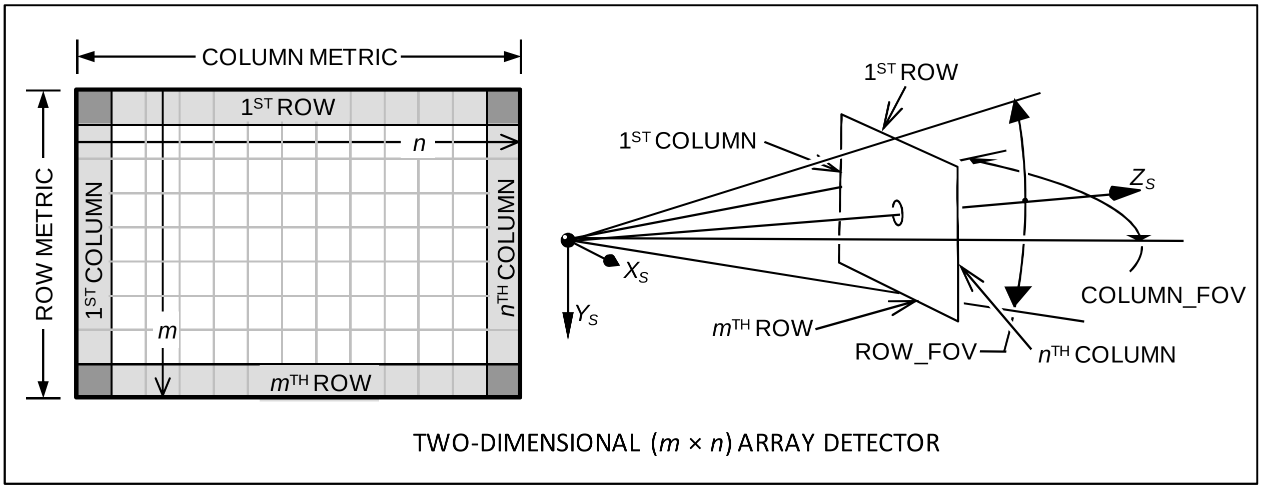


Figure Z.5.2-1. Row and Column Size and Field of View Example. For the shown two-dimensional detector array (*m* × *n*), ROW_DETECTORS is *m*; COLUMN_DETECTORS is *n*; ROW_METRIC and COLUMN_METRIC are as shown. The ROW_FOV and COLUMN_FOV are as shown on the right. It should be noted that although the full detector size might be larger than the *m* × *n* array shown, the *m* × *n* detectors are those used to make the instantaneous framelet.

Z.5.2.3. Focal Length

The best known focal length is index 02f. It is the best known value of the effective focal length. If the sensor's focal length varies with band, provide a nominal value and use the BANDSB TRE to provide the per-band focal lengths.

Z.5.2.4. Fields of View

Indices 02g and 02h are the field of view along the sensor array row and column. The angle measuring the effective field-of-view projected onto the sensor array center row and column (i.e. Sensor Horizontal Field of View and Sensor Vertical Field of View, respectively—not the half-angle FOV).

Z.5.2.5. Sensor Array Relationships

Figure Z.5.2-2 below shows the relationship between the sensor array parameters described in the previous sub-sections. These parameters are required if the TRE is intended to be used for geopositioning. The following symbols will be used in discussing these parameters:

$\mathbf{d_y}$ = Row_Metric $\mathbf{d_x}$ = Column_Metric $\mathbf{f}$ = Focal_Length $\theta_\mathbf{y}$ = Row_FOV $\theta_\mathbf{y}$ = Column_FOV

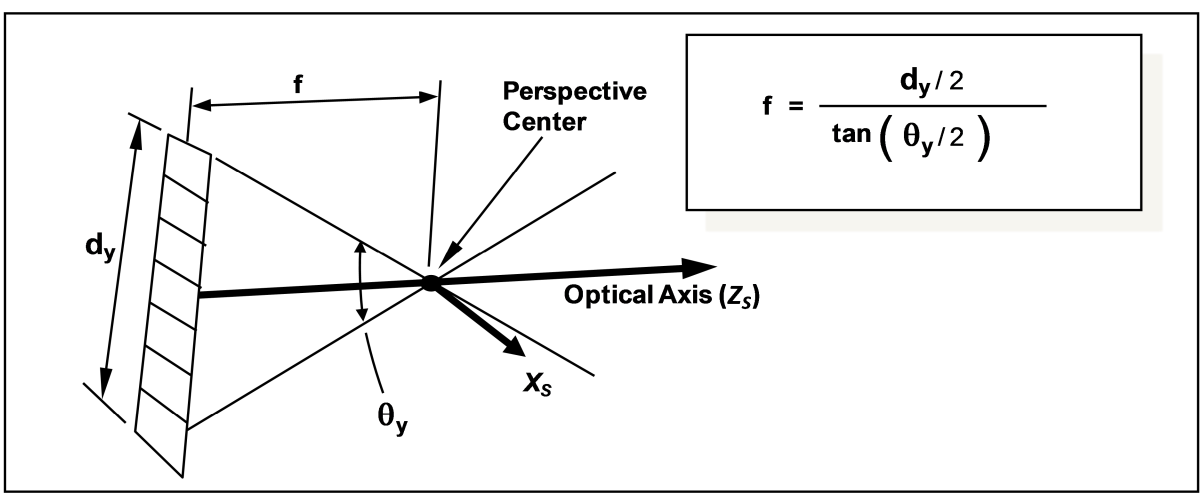


Figure Z.5.2-2. Sensor Array Row Parameter Definitions. Column parameters would be analogous.

There are two key equations that relate these values, which can be expressed relative to the focal length:

$$\mathrm{f} = \frac{d_y / 2}{\tan(\theta_y / 2)}$$

$$\mathrm{f} = \frac{d_x / 2}{\tan(\theta_x / 2)}$$

Because of the relationship between the parameters, through **f**, all five parameters need not be entered. In fact, as few as two and as many as five parameters may be entered. The values entered must define both dimensions, that is, the row and column dimensions. The input values, if over-defined, must also be consistent.

In one approach, a minimum of two parameters are required, namely θ_y and θ_x. This is a unique case which leaves the focal length indefinite such that any value, except 0, can be assumed for **f**.

Alternatively three parameters may be input. Any of the following combinations could be used to properly define the row and column dimensions:

{ $\mathbf{d_y}$, $\mathbf{d_x}$, and **f** }	{ $\mathbf{d_y}$, $\mathbf{d_x}$ and θ_y }	{ $\mathbf{d_y}$, $\mathbf{d_x}$, and θ_x }
{ θ_y, θ_x, and **f** }	{ θ_y, θ_x, and $\mathbf{d_y}$ }	{ θ_y, θ_x, and $\mathbf{d_x}$ }
{ $\mathbf{d_y}$, θ_x and **f** }	{ $\mathbf{d_x}$, θ_y, and **f** }	

The two other possible combinations of three elements are insufficient, as they do not define both the row and column dimensions. These are:

{ $\mathbf{d_y}$, θ_y, **f** }	{ $\mathbf{d_x}$, θ_x, **f** }

If more than three values are provided, the relationship between the elements must be consistent. For example, a user-specified focal length must be approximately the same as the focal length value(s) calculated using the above relationship equations.

Z.5.2.6. Focal Length Calibration Flag

Calibrated, index 02i, is the focal length calibration flag. It indicates if the provided focal length and/or fields of view are based on a geometric calibration process. ‘Y’ in this field will indicate that the focal length and/or the fields of view with the detector metrics are based on a geometric calibration. ‘N’ in this field will indicate that they are not.

Z.5.3. SENSOR CALIBRATION DATA MODULE (MODULE 03)

SENSRB can provide geometric sensor (or camera) calibration parameters.[1] These parameters are stored in Module 3 and are assumed to apply to the physical sensor array and the image coordinate system (see Z.4.5 Array Types and Z.4.6.4 Image Coordinate System). Using the calibration parameters, observed image coordinates can be corrected for various lens and detector distortions. These correcting adjustments are computed using the sequence of equations given in this section.

For a content level value (CONTENT_LEVEL, index 01f) of “8” or “9” (see Z.5.1.2 Application-Required Content Level) this module must exist and provide adequate data to apply appropriate calibration corrections. This data is typically static—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

Because the calibration parameters apply to the physical sensor array, the image coordinates measured in the NITF-stored array might first need to be adjusted using the image formation data provided in Module 4 (see also Z.4.5.4 Array Type Relationships). Once the observed image coordinates are in terms of the physical sensor array's row and column coordinates, the following equations apply.

Z.5.3.1. Center-Relative Measured Image Location

To apply the calibration equations, the user-observed image coordinates, (see Z.4.6.5 Image Array Common Coordinate System) which are measured as pixel row and column indices (or fractions thereof), must first be converted into the appropriate calibration units and referenced to the center of the image framelet using Equations Z.5-1.

$$x_{obs} = S_x{*}(col_{obs} - n/2) \qquad \text{(Z.5-1)}$$

$$y_{obs} = S_y{*}(\,m/2 - row_{obs})$$

where row_{obs} and col_{obs} (*line* and *sample*, respectively) are the user-observed image array common coordinates (origin at upper-left corner of upper-left detector) in units of pixels; and m and n are the number of rows and columns, respectively, collected in each framelet in units of pixels (ROW_SET and COLUMN_SET, indices 04e and 04f). The values S_x and S_y are scale factors to convert from units of pixels to the calibration unit system (pixel or mm, as determined by CALIBRATION_UNIT, index 03a). If the calibration unit is pixels, the scale factors are both unity; whereas, if the unit is millimeters the scale factors will be the detector pitches (in mm) along the X_I and Y_I directions (see Z.4.6.4 Image Coordinate System). These pitches are the ratios between the row or column dimension of the used detector (ROW_METRIC or COLUMN_ METRIC, indices 02d or 02e) and the number of detectors used in the row or column (ROW_DETECTORS or COLUMN_ DETECTORS, indices 02b or 02c), as discussed in Z.5.2.2.1 Row and Column Detectors. Thus, x_{obs} and y_{obs} are the observed image coordinates relative to the center of the image framelet.

Z.5.3.2. Principal Point Offset

Next, the center-relative observed image coordinates are offset to be relative to the calibrated principal point[2] using Equations Z.5-2 and the principle point offsets (PRINCIPAL_POINT_OFFSET_X (x_0) and PRINCIPAL_POINT_OFFSET_Y (y_0), indices 03b and 03c).

$$x = x_{obs} - x_0 \qquad \text{(Z.5-2)}$$

$$y = y_{obs} - y_0$$

[1] These parameters are consistent with those generated by the *Australis* software system.

[2] For digital sensors the principal point is assumed to be equivalent to both the point of autocollimation and symmetry. If calibrated fiducial coordinates are provided for a calibrated film camera, these could be provided as registered additional parameters, and the principal offsets in rows and columns (indices 11c and 11d) would locate the principal point of symmetry relative to the origin computed using those fiducial coordinates.

where x_0 and y_0 are the principal point offsets in the positive X_I and Y_I directions, respectively. The relationships between the elements of Equations Z.5-2 are illustrated in Figure Z.5.3-1.

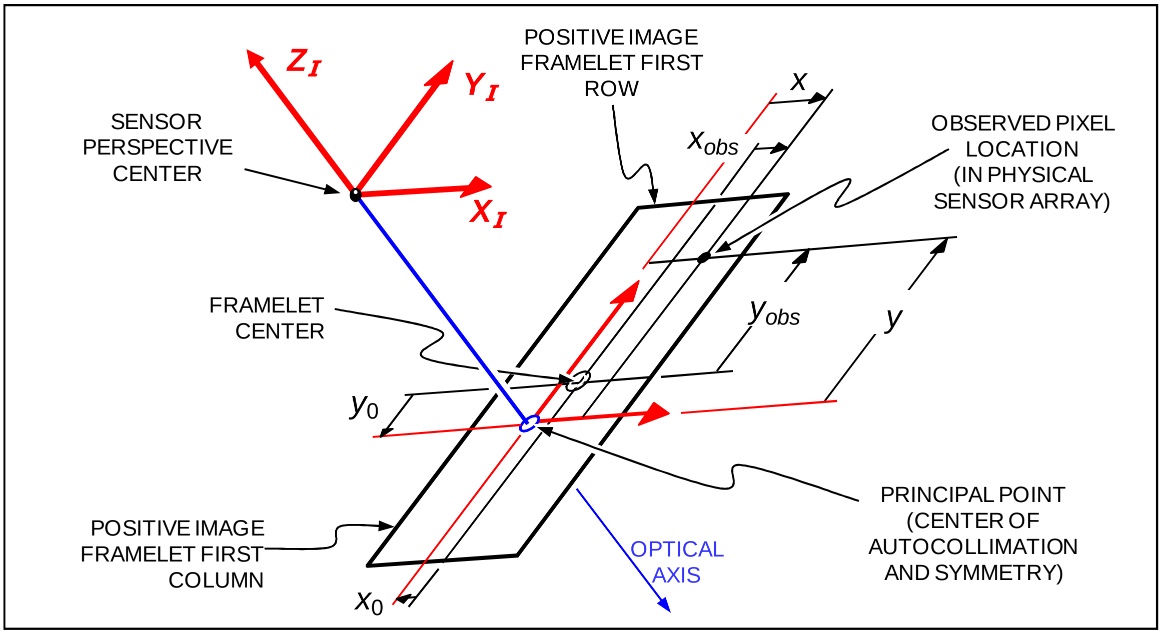


Figure Z.5.3-1. Principal-Point Offset Relationships. The optical axis may intercept the framelet's positive image plane at a location different from the framelet's center. This point of intersection is assumed to be both the center of autocollimation and symmetry. It is about this principal point that adjusted pixel location measurements are made. (Equations Z.5-1 and Z.5-2 offset row and column measurements to the desired values—*x* and *y*.) In the illustrated case, the values of both offsets (x_0 and y_0) are negative as the principal point is located in the negative X_I and Y_I directions from the framelet center.

Z.5.3.3. Radial Distortion

Using the values of *x* and *y* derived from Equations Z.5-2, the intermediate value, $r = \sqrt{x^2 + y^2}$ (the radial distance from the image principal point to the measured coordinate), is computed. Using *r*, the radial distortion, Δr, can be computed using Equation Z.5-3 and the radial distortion coefficients (RADIAL_DISTORT_1 (k_1), RADIAL_DISTORT_2 (k_2), RADIAL_DISTORT_3 (k_3); indices 03d, 03e, and 03f).[1]

$$\Delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 \qquad \text{(Z.5-3)}$$

Note: If the value of *r* is larger than the limit of radial distortion fit (RADIAL_DISTORT_LIMIT, index 03g), then the radial distortion computed using Z.5-3 is not valid. Radial distortion corrections to image locations measured beyond this limit shall not be applied, making pixel measurements in this region unreliable. As a means of last resort, the distortion (Δr) at the limit (computed using Equation Z.5-3 with *r* set to the limit) might be applied to these pixel measurements—again with uncertain results.

Z.5.3.4. Additional Distortions

Finally, the calibration-corrected image coordinates (x_{corr} and y_{corr}, relative to the principal point) can be computed using the previously computed values of *x*, *y*, *r*, and Δr in Equations Z.5-4.

[1] These radial distortion coefficients have the same eventual effect as but are different from the balanced radial distortion coefficients (K'_0, K'_1, K'_2, K'_3, K'_4), which are derived from the Simultaneous Multiframe Analytical Calibration (SMAC) system and which must be used with a "balanced" focal length.

$$x_{corr} = x + x\frac{\Delta r}{r} + p_1\left(r^2 + 2x^2\right) + 2p_2xy + b_1x + b_2y$$
$$y_{corr} = y + y\frac{\Delta r}{r} + p_2\left(r^2 + 2y^2\right) + 2p_1xy \qquad \text{(Z.5-4)}$$

where p_1 and p_2 are the first and second decentering distortion coefficients (DECENT_DISTORT_1 and DECENT_DISTORT_2, indices 03h and 03i), respectively; and b_1 and b_2 are the first and second affinity distortion coefficients (AFFINITY_DISTORT_1 and AFFINITY_DISTORT_2, indices 03j and 03k), respectively.

Z.5.3.5. Calibration Results

The adjusted values—x_{corr} and y_{corr}—resulting from Equations Z.5-4 can now be used as the image coordinates relative to the center of autocollimation or symmetry for any subsequent photogrammetric applications.

Z.5.3.6. Calibration Notes

If values for any of the above nine calibration parameters (x_0, y_0, k_1, k_2, k_3, p_1, p_2, b_1, or b_2) are not provided they can be assumed to be zero. Furthermore, some systems may provide additional registered calibration terms with their uncertainties in Module 15 (Additional Parameter Data).

Uncertainties associated with the above described calibration parameters, if available, shall be provided in Module 14 (Uncertainty Data).

It should be understood that calibrations may become less reliable with the passage of time and/or with extreme environmental conditions. The consideration of the calibration report date (CALIBRATION_DATE, index 03l) along with an understanding of the sensor's subsequent or routine operating environment may provide some insight into the reliability of the calibration parameters included in this module.

Z.5.4. IMAGE FORMATION DATA MODULE (MODULE 04)

The metadata in the Image Formation Data Module (Module 04) is used to describe the methodology used to combine one or more images from the physical sensor array to form the Initial Image and then effect a possible transformation to form the Original Stored NITF image. This is shown in the diagram below, Figure Z.5.4-1.

The metadata in this module can be used to derive basic information regarding each pixel that is eventually stored in the resulting NITF array. This would include information such as the time when each pixel would have been captured and how the original information may have been adjusted or moved in creating the final image.

For additional discussion, see also Z.4.5, Sensor and Image Arrays. For detailed information regarding image formation, see the "NGA Standardization Document, Pushbroom/Whiskbroom Sensor Model Metadata Profile Supporting Precise Geopositioning" NGA.SIG.0003_1.0.

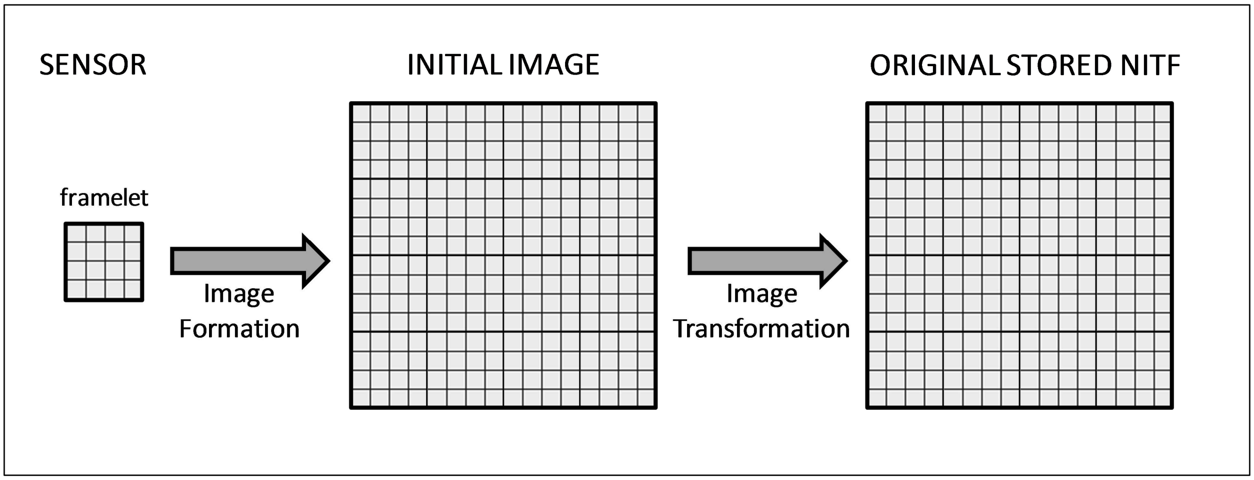


Figure Z.5.4-1. The Sensor to NITF "Pipeline". The process that takes a set of images captured by the physical sensor and constructs the initial image, which then can potentially be transformed and stored as the original NITF image file.

The following sections describe the metadata elements that represent these formation and transformation processes. The first section discusses the Imaging Methodology (the "Method" and the "Mode" of the sensor system). The next sections define the elements of the first (or left-most) arrow shown in Figure Z.5.4-1 above, representing the Image Formation parameters, which define how successive sensor images are combined to form the Initial Image. This is followed a description of how the "scanning directions" used in Image Formation are represented in various systems, such as pushbroom sensors. Subsequent sections provide a set of illustrative examples showing representative types of sensor systems. Finally, the characteristics of the right-most arrow in the above figure are discussed. These are the "Image Transformation" parameters that make final adjustments to the image prior to storing the results as the Original Stored NITF file.

For a content level value (CONTENT_LEVEL, index 01f) of "2" or higher (see Z.5.1.2 Application-Required Content Level) this module must exist and provide adequate image formation and transformation data. The image transformation parameters (TRANSFORM_PARAMs, indices 04l to 04s) may be dynamic for a collection, but the other parameters are expected to be static (see Z.4.7 Static and Dynamic Data).

Z.5.4.1. Imaging Methodology

The two fields, Image Formation Method (METHOD, index 04a) and Image Formation Mode (MODE, index 04b), describe the methodology used by the sensor to capture data and to create the Initial Image array. These two parameters describe the type of sensor and the mode of operation within that type. In the most basic case, the method would be “Single Frame”, for example a simple photographic or “framing” camera, and the mode would be an operational mode appropriate for that camera system.

Z.5.4.1.1. Image Formation Method

The Image Formation Method (METHOD, Index 04a) specifies the imaging method that was utilized and represents the relationship between the physical sensor and the initial image. The methods “Single Frame”, Multi-Frame”, “Single MIDSI”, and “Multi MIDSI” use one or more sensors simultaneously to form a single Initial Image. The methods “Pushbroom” and “Whiskbroom” use a single sensor that is used to serially populate various regions of a single image. The method parameter value can be any of the following NTB approved names:

Value	Description
Single Frame	The use of a single frame sensor to detect a single sensor array and to generate a single Initial Image. Sensor metadata is expected to be synchronous with the image collection time.
Multi-Frame	The use of multiple frame sensors to simultaneously detect a set of sensor arrays which are then combined to generate a single Initial Image. Sensor metadata is expected to pertain to the individual sensors and be synchronous with the image collection time.
Single MIDSI	Single Motion Imagery Derived Still Image. The use of a single frame sensor to detect a time-sequence of single image arrays (frames), one of which is extracted resulting in a single Initial Image. Because motion imagery is generally compressed using an approach such as MPEG, the final image may contain pixels or shared pixels from earlier images. The sensor metadata is not expected to be synchronous with any particular frames’ image collection time.
Multi-MIDSI	Multi Motion Imagery Derived Still Image. The use of multiple frame sensors to detect single image arrays which are then time-sequenced as motion imagery. One of these frames is extracted resulting in a single Initial Image. Because motion imagery is generally compressed using an approach such as MPEG, the final image may contain pixels or shared pixels from earlier images. The sensor metadata is not expected to be synchronous with any particular frames’ image collection time.
Pushbroom	The use of a single frame sensor having a defined number of detectors in the row and column directions of the sensor. This single frame is saved as a “framelet” in the Initial Image array and is then moved (such as by way of a rotating mirror or platform motion) to effectively move that sensor array so as to complete a full row or column of the Initial Image and then to move back to the beginning of that row or column to begin another capture sequence. (See section Z.5.4.3., Scanning Direction and Time of Pixel Capture).
Whiskbroom	Similar to Pushbroom except that when the sensor array has completed one pass across the Initial Image, the subsequent row or column is captured in the reverse direction. This continues to completion of the image resulting in a “serpentine” path of sequential “framelet” captures. (See section Z.5.4.3., Scanning Direction and Time of Pixel Capture).

Z.5.4.1.2. Image Formation Mode

The Image Formation Mode (MODE, index 04b) is a registered three-digit code indicating the actual operating mode within the sensor system that is used to collect or detect the image data. These three-digit codes are the same codes as used in ACFTB's MPLAN, which relates a physical sensor type with its various modes of operation. MODE will generally be the same as the "planned mode" given in ACFTB's MPLAN. It would differ from MPLAN if the actual mode (MODE) were different from the planned mode (MPLAN).

Z.5.4.2. Image Formation Fields

The first two image arrays shown earlier in Figure Z.5.4-1 represent the creation of the Initial Image from the physical sensor array. This section will focus on the intermediate process of Image Formation as represented by the middle image array shown in Figure Z.5.4-2 below.

There are four pairs of values used to characterize the imaging parameters used to create the Initial Image. These are ROW_COUNT and COLUMN_COUNT, indices 04c and 04d, ROW_SET and COLUMN_SET, indices 04e and 04f, ROW_RATE and COLUMN_RATE, indices 04g and 04h, and FIRST_PIXEL_ROW and FIRST_PIXEL_COLUMN, indices 04j and 04k. Figure Z.5.4-2 begins the process of showing these values for the various types of image formation methods.

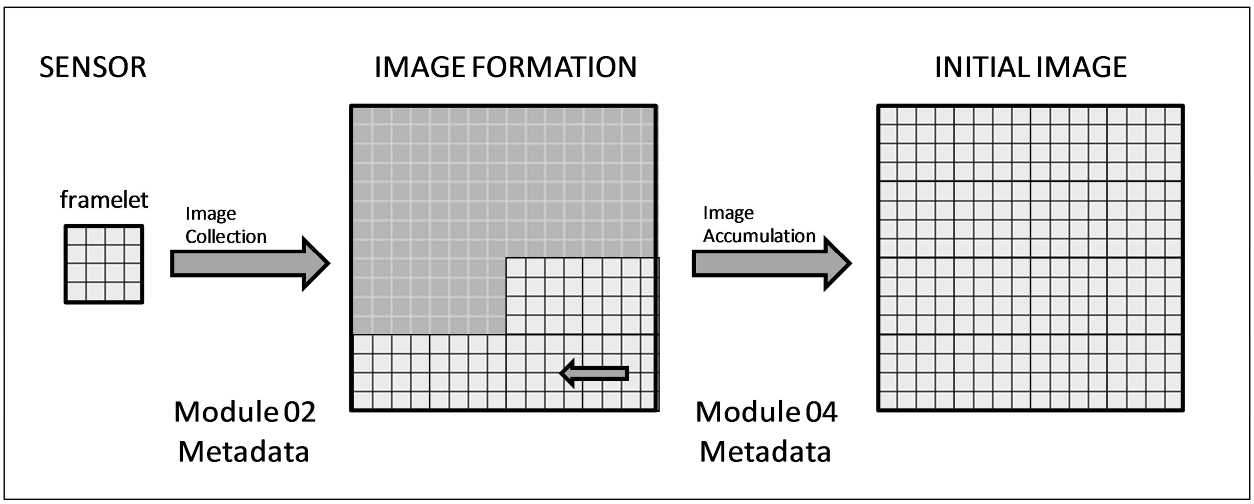


Figure Z.5.4-2. Image Collection and Image Accumulation. The process that takes a set of images captured by the physical sensor and consecutively accumulates these images to form the "Initial Image".

The figure below, Figure Z.5.4-3 is an expansion of the middle array above, referred to as "Image Formation." This figure brings together the essential elements of the Module 04 metadata in that it describes the relationship between the physical sensor and how the framelets from that sensor are combined together over a short period of time to form an image array. Pixel locations are defined in the Pixel Coordinate system, (see Z.4.6.6 Pixel Coordinate System).

In this figure, the sensor is represented as a p by q array of detectors that will be used to fill an identically sized region of the Initial Image starting at the First Pixel location. The sensor is then "moved" as required so as to capture the next sub-image and populate the next framelet of the Initial Image. The ratio of the number of pixels in the final image to the number of pixels in the physical sensor will be the number of framelets or successive capture steps in the creation of the image.

In some cases, the sensor is the same size as the final image. In this case, all of the pixels in this one image are captured at the same time. In other cases, such as when field "Method" is equal to "Pushbroom" or "Whiskbroom", the sensor array is used successively to populate various framelets of the Initial Image.

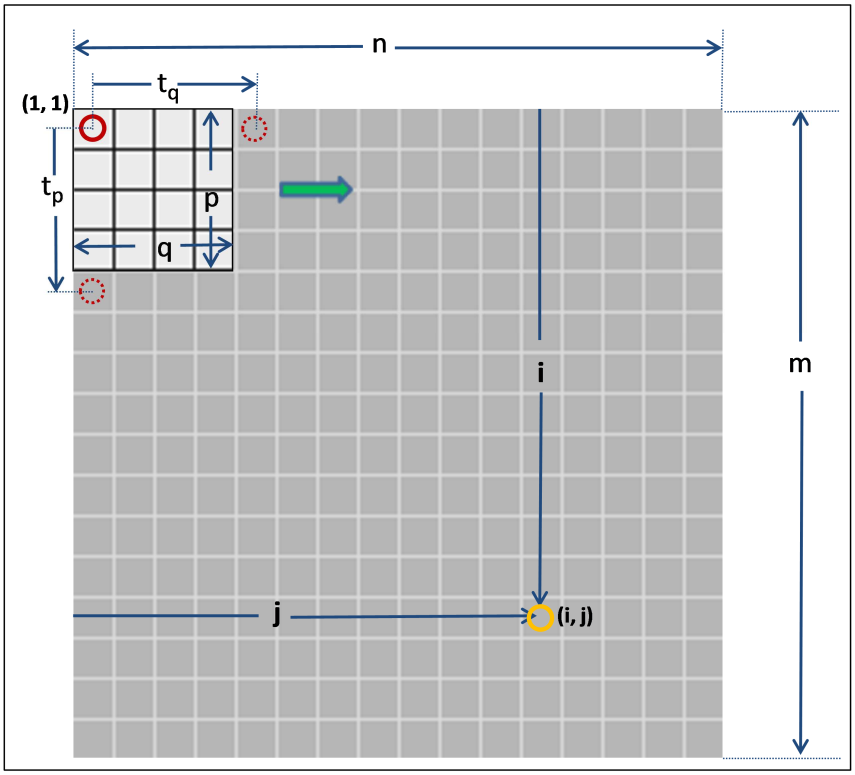


Figure Z.5.4-3. Image Formation Parameters. A representation of the Module 04 parameters defining the Row and Column imaging methodology. The parameters p and q relate to the type of sensor.

TABLE Z.5.4-1 MODULE 04 INDICES AND MEANINGS

Index	Field	Symbol
01k	Start Time	t_0
04c	Row Count	m
04d	Column Count	n
04e	Row Set	p
04f	Column Set	q
04g	Row Rate	t_p
04h	Column Rate	t_q
04i	1st Pixel Row	i based
04j	1st Pixel Column	j based

In the figure, the 1st Pixel Row and Column is the top-left corner, pixel (1, 1). A generalized pixel is shown as (i, j) meaning that the first index is in the row index while the second index, j, is the column index.

Consider the following four examples of different values for p and q:

- p = m and q = n. A **Single Frame or Single MIDSI** in which the entire image is captured simultaneously by the physical sensor.
- p is an integer fraction of m and/or q is an integer fraction of n. A **Multi-Frame or Multi-MIDSI** image formation in which a set of framelets (p by q pixels) are simultaneously captured to form the total Initial Image array.
- p = m or q = n. A **Pushbroom** sensor in which an entire row or column is captured by the sensor and then that array is moved perpendicular to the array to capture the next row or column in the final image.

- p = q = 1. A “flying dot”. For a **Whiskbroom,** this would indicate a single pixel being captured as a mirror moves the pixel, in this instance, in a serpentine pattern (back and forth) with the initial motion defined by the sign of the row or column rate with the lowest magnitude.

The following sections detail the four pairs of parameters representing the image Formation Fields.

Z.5.4.2.1. Row and Column Count

Row Count and Column Count, (ROW_COUNT and COLUMN_COUNT, indices 04c and 04d), are the number of rows (m) and columns (n) in the Initial Image array.

When there are no Image Transformation Parameters, these values are also equal to the Image Subheader values NROWS and NCOLS of the resulting NITF image segment.

Z.5.4.2.2. Row and Column Detection Set

Row and Column Detection Set, (ROW_SET and COLUMN_SET, indices 04e and 04f), are the number of rows and columns collected simultaneously to form the physical sensor array. Typically these values will be equal to the module 02 fields ROW_DETECTORS and COLUMN_DETECTORS, (indices 02b and 02c). These values can also be less than the module 02 detectors count when only a subset of the physical sensor is used to populate each framelet of the Initial Image array. It is generally the case that the ratio of the row count to the row set will be an integer, and similarly for the column count to the column set, meaning that an integer number of sets (framelets) are required to populate each direction of the Initial Image.

For instantaneous framing sensors, these values shall be equal to the row and column counts (ROW_COUNT and COLUMN_COUNT, indices 04c and 04d). As another example, for a single-line pushbroom, one field shall have a meaningful value (equal to the row or column count) and the other shall be one.

Z.5.4.2.3. Row and Column Detection Rate

Row and Column Detection Rates, (ROW_RATE and COLUMN_RATE, indices 04g and 04h), are the durations in time, the “step time”, between starting the capture of the first detection set and starting the capture of the second detection set in that same row or column when forming the image array. For framing sensors, both of these values shall be zeros (or optionally the shutter speed or integration time). For pushbroom sensors, one field (in the direction of the scanning) shall have a meaningful value and the other shall be zero (or the shutter speed/integration time).

In figure Z.5.4-3 above, consecutive sets are being captured in the direction of the thick arrow. This is the column direction, which has increasing indices for the columns. In this instance, the COLUMN_RATE would be the increment in time between starting the capture of the first set, pixel (1, 1), and starting capture of the second set with its top-left pixel at (1, q) (the dotted red circle in the figure). After the final set in that row has been captured, the set would begin the capture of the next set in the image, starting at pixel (p, 1) (the dotted circle below the p by q set). The time between the capture of the (1, 1) set and the (p, 1) set is t_p, the Row_Rate.

The magnitude (absolute value) of the detection rate will be equal to the time differential between initiating the capture of two consecutive sets within the current row or column. When the Column_Rate magnitude is less than the Row_Rate magnitude, it means that the initial motion is along the current row (increasing column indices). This motion will continue until the end of the row is reached, then the set moves to the next column. The time between the original set capture and this new capture in the next column is the Column_Rate.

To indicate a positive or negative scan direction, this field value shall be positive when FIRST_PIXEL_ROW is stored within Initial Image’s first row and negative when stored within last image row. Or this field value shall be positive when FIRST_PIXEL_COLUMN is stored within first Initial Image column and negative when stored within last image column.

For a Formation Method of “Pushbroom” the set moves in the “positive” direction (for a positive Rate) after completing its first pass across the array and moving to the next row or column. For a Formation Method of “Whiskbroom” the set moves first in the positive direction and then in the negative direction on the next row or column creating a “serpentine” or “back-and-forth” pattern.

Z.5.4.2.4. Row and Column of First Collected Pixel

The row and column within the Initial Image array which receives the first collected pixel or line set of the image array is the Row and Column of First Collected Pixel, (FIRST_PIXEL_ROW and FIRST_PIXEL COLUMN, indices 04i and 04j).

In Figure Z.5.4-3, the FIRST_PIXEL_ROW and FIRST_PIXEL_COLUMN would be (1, 1) where the first index represents the direction of increasing row indices (i and m in the figure), while the second index represents the direction of increasing column indices (j and n in the figures).

For framing sensors, these values shall be the top left pixel (1, 1) of the Initial Image array. For the pushbroom example shown below in Figure Z.5.4-4 the first pixel would be (1, 1) while for the Pushbroom example shown in Figure Z.5.4-5 the first pixel would be (m, 1).

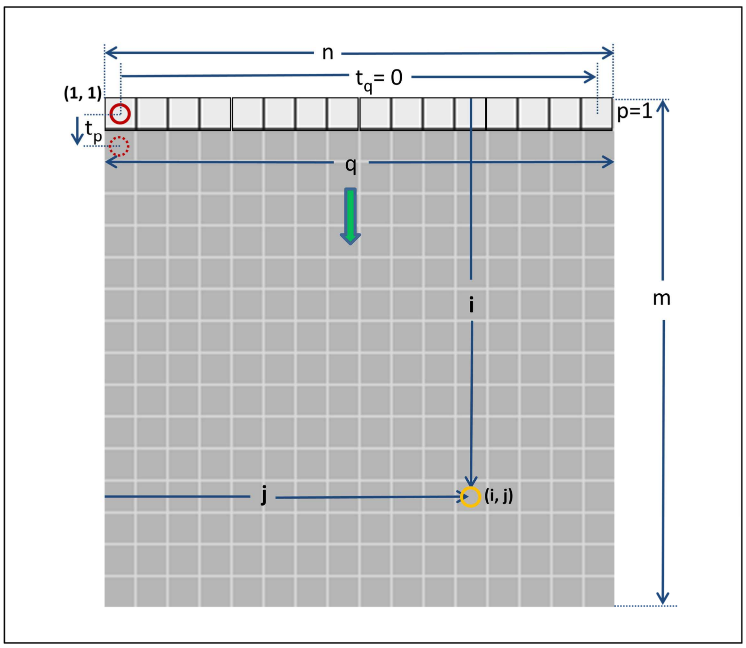


Figure Z.5.4-4 Pushbroom Image Formation

Z.5.4.3. Scanning Direction and Time of Pixel Capture

There are three principal classes of Image Formation processes that affect the calculation of the time when each pixel was captured:

- “Single Frame” and “Multi-Frame” sensors capture the entire image at one time and so all pixels were captured at time t_0, (START_TIME, index 01k). Optionally, one can enter the shutter speed of the camera into both the Row and Column Detection Rates (index 04g and 04h).
- “Single MIDSI” and “Multi-MIDSI” sensors generally create a single image from pixels captured at various times and pixel locations because of the video compression method. As a result one can only approximate the time of capture of each pixel sent to the Initial Image array.

- “Pushbroom” and “Whiskbroom” sensors create an Initial Image array from a time sequence of sub-images captured by the physical sensor array. For these kinds of sensors there is a straightforward calculation that can be used to determine the time of capture of each pixel as described in the following paragraphs.

Figure Z.5.4-4, above, shows one example of a Pushbroom sensor. In this case, the entire top row is captured simultaneously and then the mirror or device is rotated perpendicular to the captured row so as to be ready to capture the next row of the image. The time required to move the sensor to the second row is represented as t_p. All of the pixels in the first row are captured at time t_0,and all of the pixels in the second row are captured at time $t_0 + t_p$. Thus, if the First Pixel, (1,1), is captured at time t_0, as specified in Module 01, (index 01k), then a generalized pixel (i, j) would be captured at $t_0 + (i - 1)*t_p$.

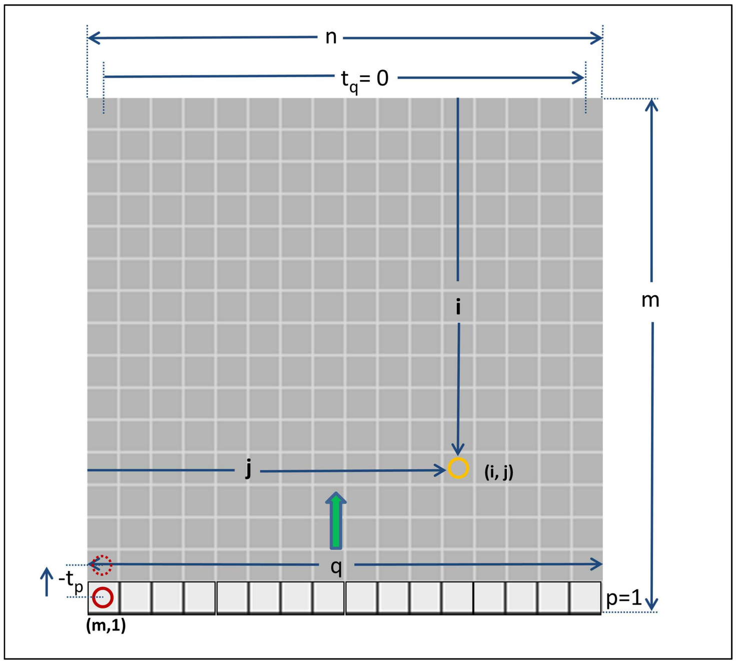


Figure Z.5.4-5 Scanning Direction for a Pushbroom

As a separate example, Figure Z.5.4-5 shows another type of Pushbroom sensor. In this diagram, the entire bottom row is captured at time t_0 while the row above that pixel starting at (m-1,1) would be captured at time t_p.

Another example of a Pushbroom sensor would be the case when only one pixel is captured at a time, a “flying spot”, such that p = q = 1. In this case, each pixel in each row requires t_q seconds and the first pixel in the second row would be captured at time t_p and therefore the time required to capture a complete row would be

$$t_n = n*t_q.$$

In this case, pixel (i, j) would be captured at time

$$t = t0 + (i - 1)*t_p + (j - 1)* t_q$$

A similar analysis would be appropriate when the sensor captures a *column* and then scans to the right (or left as discussed later in this section).

The **direction** of scanning has been represented in the figures as the broad arrow, which in Figures Z.5.4-3 and Z.5.4-4 pointed to the right or down. In some instances of a Pushbroom or Whiskbroom scanner, the scanning direction goes from the bottom of the image to the top, as shown in Figure Z.5.4-5, or as a column starting on the right and scanning to the left.

In Figure Z.5.4-5, the First Pixel Row and First Pixel Column is the bottom left corner of the Initial Image (m,1). In order to specify that the subsequent scans are moving up from the bottom, place a minus sign in front of the Row Rate parameter, $-t_{p,}$ and set the Column Rate $t_q = 0$ (or the shutter speed which should be less than the magnitude of t_p.)

If this example were of a single pixel scanner starting at the indicated pixel (m, 1) moving to the right along a row, and then moving up to the next higher Row, enter a positive number for Column Rate (the move to the right) and a negative number (of higher magnitude) for the Row Rate (the upward motion).

If the First Pixel was the bottom right, (m, n), and the scans moved to the left and then up, both the Column Rate and the Row Rate would be negative numbers.

Z.5.4.4. Image Formation Examples

Through the use of the parameter pairs Row_Rate / Column_Rate, Row_First_Pixel / Column_First_Pixel, and the Formation Method type, current known sensor collection geometries can be represented in Module 04 of the SENSRB TRE. In the following sections, we provide a few examples for some common camera systems.

Z.5.4.4.1. Framing Sensor

The most straightforward cases are the framing sensors for still and motion imagery systems. In these systems, the collection of imaging rows and columns are considered simultaneous and essentially instantaneous (or at least captured within the shutter speed or integration time) and are typically also stored as the resulting NITF array.

The table below illustrates an example for a 512 row detector by 1024 column detector sensor array resulting directly as a 512-row by 1024-column Initial Image array and Original Stored NITF array. The imaging array in this case is considered simultaneous and instantaneous, which equates to a 512-row "set" by 1024-column "set" with all pixels captured at the end of the shutter period, 1/250th of a second, (0.004). Table Z.5.4-2 and subsequent tables in this section demonstrate the locations and values in the NITF file and in the SENSRB TRE the various image formation parameters would be defined.

TABLE Z.5.4-2 FRAMING SENSOR IMAGE FORMATION

NITF Image Subheader Parameters

NROWS	00000512
NCOLS	00001024

SENSRB Parameters

02	SENSOR_ARRAY	Y
02a	ROW_DETECTORS	00000512
02b	COLUMN_DETECTORS	00001024
…		
04	IMAGE_FORMATION_DATA	Y
04a	METHOD	Single Frame
04b	MODE	*NTB Value*
04c	ROW_COUNT	00000512
04d	COLUMN_COUNT	00001024
04e	ROW_SET	00000512
04f	COLUMN_SET	00001024
04g	ROW_RATE	0.00400000
04h	COLUMN_RATE	0.00400000
04i	FIRST_PIXEL_ROW	00000001
04j	FIRST_PIXEL_COLUMN	00000001

Z.5.4.4.2. An Image Transformation to Create the Original Stored NITF Array

For many purposes, automated or manual modifications may occur at a variety of points along the image creation "pipeline" and as a result, the Original Stored NITF array may differ from the Initial Image array. Such image modification processes should be evident in the metadata to inform photogrammetrists when translating applicable sensor array data to the NITF array.

One example of such a modification would be an image transformation prior to the creation of the Original Stored NITF so as to create a "natural look" for downstream analysts, or perhaps scaling to increase or decrease the size of the NITF image segment.

Table Z.5.4-3 below illustrates the first case of the framing sensor discussed in Z.5.4.4.1 but downsized by half and rotated 90 degrees about the image center.

TABLE Z.5.4-3 MODIFIED FRAMING SENSOR IMAGE FORMATION

SENSRB Parameters

02	SENSOR_ARRAY_DATA	Y
02a	ROW_DETECTORS	00000512
02b	COLUMN_DETECTORS	00001024
…		
04	IMAGE_FORMATION_DATA	Y
…		
04c	ROW_COUNT	00000512
04d	COLUMN_COUNT	00001024
04e	ROW_SET	00000512
04f	COLUMN_SET	00001024
04g	ROW_RATE	0.00400000
04h	COLUMN_RATE	0.00400000
04i	FIRST_PIXEL_ROW	00000001
04j	FIRST_PIXEL_COLUMN	00000001
04k	TRANSFORM_PARAMS	4
04l	TRANSFORM_PARAM1	0.5000000000
04m	TRANSFORM_PARAM2	90.000000000
04n	TRANSFORM_PARAM3	256.00000000
04o	TRANSFORM_PARAM4	512.00000000

As will be discussed in the next section, four parameters (specified by entering a "4" in index 04k) required to effect this transformation are the scale factor, 0.5, the angle, positive 90.0, and the center of the image (256, 512), which are entered into indices 04l, 04m, 04n and 04o.

(Note that the image transformation as shown in this example is different than a modification of the Original Stored NITF to create an output NITF image defined using the ICHIPB TRE (STDI-0002, Appendix B), which is outside of the scope of this TRE.)

Z.5.4.4.3 Scanning Sensors

The image formation process for scanning sensors involves the aggregation of multiple rows and/or columns each being detected instantaneously in most cases. The resulting NITF array will appear as a single image, but accumulated over time. As a result, additional parameters are needed to account for the time-based variables. Below are two cases to distinguish between pushbroom (unidirectional aggregation) and whiskbroom sensors (bidirectional aggregation).

Figure Z.5.4-6 on the left is an example of a 1-row by 1024-column pushbroom sensor array resulting in a 512-row by 1024-column Initial Image array.

Figure Z.5.4-7 on the right is an example of a 1-row by 1-column flying-spot whiskbroom sensor array resulting in a 512-row by 1024-column NITF array.

TABLE Z.5.4-4 SCANNING SENSOR IMAGE FORMATION

Pushbroom Example
NITF Image Subheader Parameters

NROWS	00000512
NCOLS	00001024

SENSRB Parameters

02	SENSOR_ARRAY	Y
02a	ROW_DETECTORS	00000001
02b	COLUMN_DETECTORS	00001024
…		
04	IMAGE_FORMATION_DATA	Y
04a	METHOD	Pushbroom
04b	MODE	*NTB Value*
04c	ROW_COUNT	00000512
04d	COLUMN_COUNT	00001024
04e	ROW_SET	00000001
04f	COLUMN_SET	00001024
04g	ROW_RATE	0.00200000
04h	COLUMN_RATE	0000000000
04i	FIRST_PIXEL_ROW	00000001
04j	FIRST_PIXEL_COLUMN	00000001
…		

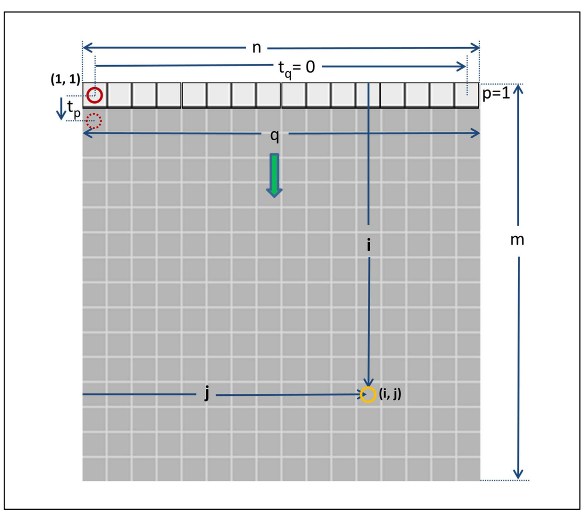


Figure Z.5.4-6 Pushbroom Example

Whiskbroom Example
NITF Image Subheader Parameters

NROWS	00000512
NCOLS	00001024

SENSRB Parameters

02	SENSOR_ARRAY	Y
02a	ROW_DETECTORS	0000001
02b	COLUMN_DETECTORS	0000001
…		
04	IMAGE_FORMATION_DATA	Y
04	METHOD	Whiskbroom
04	MODE	*NTB Value*
04c	ROW_COUNT	00000512
04d	COLUMN_COUNT	00001024
04e	ROW_SET	00000001
04f	COLUMN_SET	00000001
04g	ROW_RATE	-0.5120000
04h	COLUMN_RATE	0.00100000
04i	FIRST_PIXEL_ROW	00000512
04j	FIRST_PIXEL_COLUMN	00000001
04k	TRANSFORM_PARAMS	0
…		

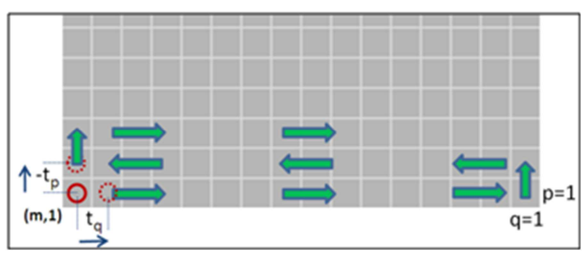


Figure Z.5.4-7 Whiskbroom Example

Z.5.4.5. Image Transformations

An Image Transformation is a process which applies a mathematical transformation to an image to produce another image. These parameters are applied to the image using the Image Coordinate System, having its origin at the center of the image.

Z.5.4.5.1 Image Transformation Fields

A transformation is defined using the following set of fields in Module 04, as follows:

- The index 04k (TRANSFORM_PARAMS). This field has a value between zero and eight with zero indicating that there are no image transformation parameters, thus the Image Transformation "H" is the Identity Matrix. Therefore, if no transformation parameters are required, field 04k shall contain a zero. In that case, none of the subsequent fields in the remainder of this Module should exist.

  This is not unlike the use of a "Y" or "N" to define the subsequent existence of an optional Module, or similarly the use of a "0" in a looping module to specify that no information regarding that module exists.

  If transformation parameters are required, place into field 04k the number of transformation parameters required (a number between 1 and 8) and then enter precisely that number of subsequent fields (the coefficients of the transformation).

  The value in index 04k is also used to specify the class of the transformation, that is, the nature and order of the coefficients associated with that number of terms (see Table 5.4.4-4 below, Image Transformation Parameters).

- Indices 04l to 04s (TRANSFORM_PARAMi for "i" equal to 1 thru 8). These fields provide the coefficients of the transformation matrix between the image coordinates associated with the Initial Image array represented in this TRE and those corresponding to the NITF image segment's data array.

  Fields 04l to 04s shall have as many values specified as defined by the parameter 04k, (TRANSFORM_PARAMS) – and no more. A meaningful value for each provided parameter is required.

Z.5.4.5.2 Image Transformation Overview

In the image "pipeline" that has been discussed, there are two places where an image transformation can take place:

- A "pre-processing" transformation, which effects a transformation from the physical sensor array to a framelet or sub-segment of the Initial Image array. This can be done, for example, to effect "image stabilization". In this instance, the image Transformation Parameters would be required for *each* sub-segment transformation, if known. In this case, the implementer would need to define these parameter values in Module 12, TIME_STAMPED_DATA_SETS, or in Module 13, PIXEL_REFERENCED_DATA_SETS, as an array of the up to 8 Transformation Parameters for each of the (n * m) / (p * q) framelets of the Initial Image.

- A "post-processing" transformation, which performs a transformation of the Initial Image array to form the Original Stored NITF array. These are typically transformations such as scaling, rotation, stretching, and similar mathematical transformations of the entire image. This second type of transformation can be completely defined using the TRANSFORM_PARAMs, indices 04k through 04s.

In both of these instances, the TRANSFORM_PARAMS will be used to effect their respective image adjustments as discussed in the following sections.

Z.5.4.5.3. Image Transformation Equations

The Image Transformation Parameters define the coefficients of the transformation equations that convert the Initial Image into the Original Stored NITF array, see Figure Z.5.4-8. The input to the transformation is the set of all pixels in the Initial Image, any one of which is expressed as $\begin{bmatrix} Xo \\ Yo \end{bmatrix}$. The output of the transformation is the set of all pixels in the resulting Original Stored NITF array, any one of which is expressed as $\begin{bmatrix} Xn \\ Yn \end{bmatrix}$.

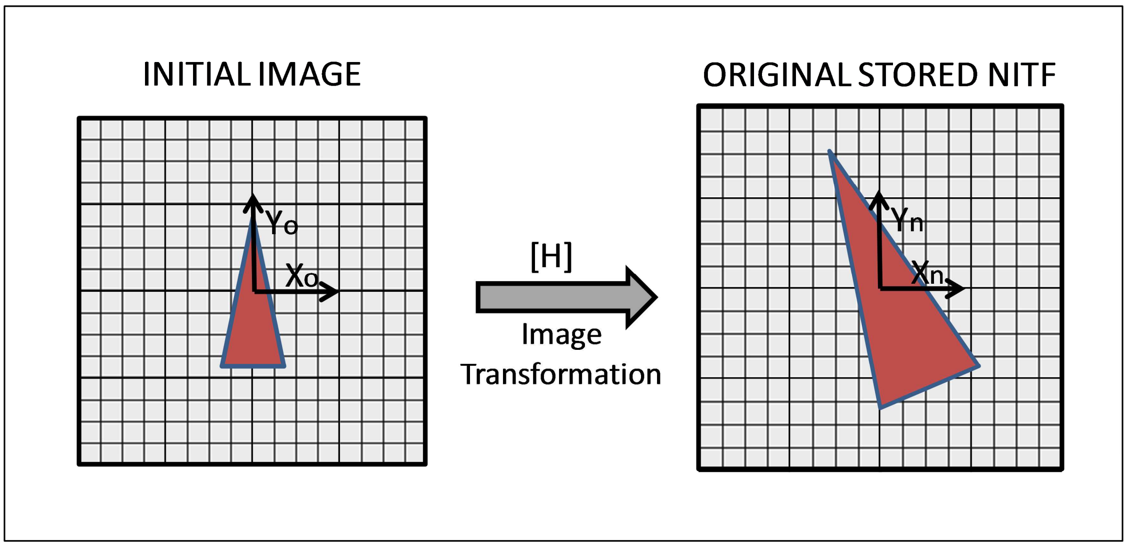


Figure Z.5.4-8 Transformation (Rotation and Scaling) of the Initial Image to create the Original Stored NITF

First, it is important to define the coordinate system used in these transformations. As a matter of convention, we define the origin of the coordinate system as the center of the Initial Image and the corresponding center of the Original Stored NITF as the center of that image. Despite the fact that pixels are generally referenced from the top left and with increasing row indices going down, for the purposes of transformations, we utilize the conventional right-hand coordinate system with X to the right and Y going up, all measured from the center of the image in units of pixels.

To simplify the input and to save space in the TRE, the parameters in the transformation can be input in six different ways corresponding to the nature of the various types of transformations. The first is the identity or “No Transformation” case (which requires no data at all) and can be expressed as:

$$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Xo \\ Yo \end{bmatrix}.$$

The most basic transformation is a pure translation, shown in Figure Z.5.4.9 below, in which every pixel is shifted by a fixed amount in X and Y represented as Dx and Dy. The equation for this would be:

$$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}.$$

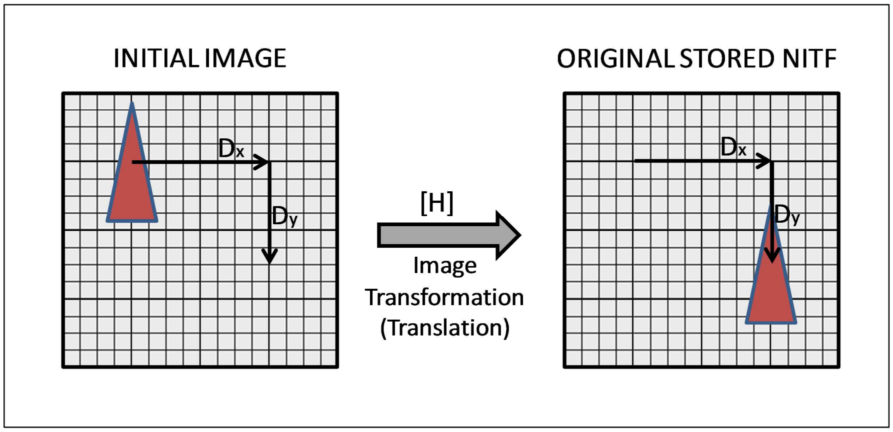


Figure Z.5.4-9 Translation Transformation

To input this, enter “2” for the number of Transform Parameters, index 04k, followed by the values for Dx and Dy in indices 04l and 04m. No other values should be entered.

The next class of transformation is the Isogonal Affine Transformation which requires four parameters referred to as S, α, Dx, and Dy. These parameters represent a uniform scaling (S), a rotation of α degrees about the image origin, followed by a translation of (Dx, Dy). The matrix representation of this transformation is:

$$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = S \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$$

This transformation first rotates the Initial Image about the origin by α degrees followed by a uniform scaling of magnitude “S” relative to the image origin, and finally followed by a translation to the final location as a result of a (Dx, Dy) translation, as shown in Figure Z.5.4.10, below.

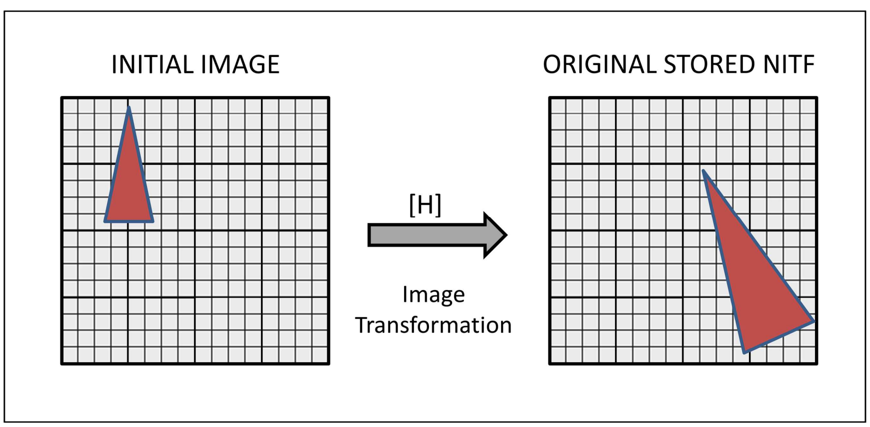


Figure Z.5.4-10 Rotation, Scaling, Translation: An Isogonal Affine Transformation

To input this transformation, enter “4” for the number of Transform Parameters, index 04k, followed by the values for S, α, Dx and Dy in that order in indices 04l, 04m, 04n, and 04o. No other values should be entered.

The next transformation class simply permits the scaling to be different in the x and y directions, which means that there are now five parameters and the transformation equation is:

$$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$$

The next transformation accepts 6 parameters and is referred to as an affine transformation, which generalizes the orthogonal affine transformation by replacing the rotation and scaling with a 2 by 2 matrix so that the transformation equation becomes:

$$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$$

As a result, the affine transformation requires six parameters, namely {a, b, c, d, Dx, Dy} in that order.

The most complete transformation, the Projective Transformation, uses a defined set of eight coefficients that are used in a rational polynomial to effect the transformation according to the following equations:

$$Xn = \frac{a_1 Xo + a_2 Yo + a_3}{d_1 Xo + d_2 Yo + 1} \qquad Yn = \frac{b_1 Xo + b_2 Yo + b_3}{d_1 Xo + d_2 Yo + 1}$$

To input this transformation, enter “8” for the number of Transform Parameters, index 04k, followed by the values for {a1, a2, a3, b1, b2, b3, d1, d2} in that order in indices 04l to 04s.

Note that the 8 parameter Projective Transformation becomes the 6 parameter Affine Transformation when d1 = d2 = 0 (no projection/warping) and we replace the numerator terms {a1, a2, b1, b2} with {a, b, c, d} and {a3, b3} are replaced with {Dx, Dy}.

The following table, Z.5.4-5, shows available transformations each having various numbers of terms as well as the order of the terms and their corresponding equations.

TABLE Z.5.4-5. IMAGE TRANSFORMATION PARAMETERS

# Parameters	Meaning	Equations
0	No Transformation	$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Xo \\ Yo \end{bmatrix}$
2	Translation {Dx, Dy}	$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$
4	Isogonal Affine Transformation {S, α, Dx, Dy}	$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = S\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$
5	Orthogonal Affine Transformation {Cx, Cy, α, Dx, Dy}	$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} Cx & 0 \\ 0 & Cy \end{bmatrix}\begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$
6	Affine Transformation {a, b, c, d, Dx, Dy}	$\begin{bmatrix} Xn \\ Yn \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} Xo \\ Yo \end{bmatrix} + \begin{bmatrix} Dx \\ Dy \end{bmatrix}$
8	Projective Transformation {a1, a2, a3, b1, b2, b3, d1, d2}	$Xn = \frac{a_1 Xo + a_2 Yo + a_3}{d_1 Xo + d_2 Yo + 1}$ $Yn = \frac{b_1 Xo + b_2 Yo + b_3}{d_1 Xo + d_2 Yo + 1}$

Z.5.4.5.4 Transformation Example

A common transformation would be the four-parameter Isogonal Affine Transformation, which might be applied as a pre-processing transformation to correct “platform jitter” between sensor frame captures. As an example, this might be nothing more than a pair of coordinate translations, D_x and D_y, with no scaling, S, or rotation, α. Thus the equation would be simply:

$X_n = X_o + D_x$ and
$Y_n = Y_o + D_y$.

Although this only requires a 2 parameter specification, for illustrative purposes we can also use the 4 parameter representation. Given that the ordering of the 4-parameter class of transformation is {S, α, D_x, D_y} as shown in Table Z.5.4-5, the field 04k (TRANSFORM_PARAMS) would be 4 and the first four TRANSFORM_PARAMs would be

{1.0, 0.0, D_x, D_y}.

No other TRANSFORM_PARAMs would be entered.

Z.5.5. REFERENCE TIME OR PIXEL MODULE (MODULE 05)

The Reference Time or Pixel Module (Module 05) does not include a flag to indicate its existence. This module is required for each and every SENSRB TRE.

All other field values within the TRE will be associated with the reference time or pixel location of applicability indicated in Module 5.[1] The references in this module are expected to be dynamic, such that if multiple SENSRB TREs are provided, each is likely to have different references of applicability—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

Either the reference time of applicability (REFERENCE_TIME, index 05a) or the reference pixel location of applicability (both REFERENCE_ROW and REFERENCE_COLUMN, indices 05b and 05c) must be provided with a meaningful value (not unspecified). With one reference of applicability (time or pixel location) provided, the other means of reference (pixel location or time) can contain the unspecified indicator. If both a reference time and a pixel location are provided and are in conflict (internally inconsistent), the reference pixel location will take precedence over the reference time.

It is expected that some image exploitation methods may use interpolation or filtering schemes to derive parameter values for times or pixel locations not included as the references of applicability—see Z.4.10 Dynamic Value Interpolations.

Z.5.5.1. Reference Times

The reference time (REFERENCE_TIME, index 05a) is indicated in seconds relative to the absolute start time from Module 1 (START_TIME, index 01m). If the data in a TRE pertains to an instant in time before the imaging began, the reference time of applicability for that TRE will be negative. A reference time of zero (0) indicates that the TRE data is applicable to the same instant in time that the imaging began (imaging start time); whereas, a positive, non-zero reference time denotes that the data pertains to an instant some time after the imaging start time. Reference times of applicability might extend beyond the imaging time. Reference times before and after the image collection time can facilitate interpolation schemes so that values can be estimated for various times or pixel locations other than those specified—see Z.4.10 Dynamic Value Interpolations.

The value range for this field allows for reference times up to nearly 28 hours before or after the imaging start time with a resolution of 10 microseconds. Much finer resolutions are possible for reference times closer to the imaging start time. For example, nanosecond resolutions are possible up to 100 seconds before or after the imaging start time. At the same time, the potential resolution of this field should not be confused with accuracy; the guidance of Z.4.3.3 Numerical Precision should be followed. Uncertainties in the reference time can be provided in the Uncertainty Data Module (Module 14).

Z.5.5.2. Reference Pixel Locations

The reference pixel location (REFERENCE_ROW and REFERENCE_COLUMN, indices 05b and 05c) are provided using the image array common coordinate system as it applies to the NITF-stored array (see Z.4.5.3 NITF-Stored Image Array and Z.4.6.5 Image Array Common Coordinate System). The reference pixel location field values indicate which image pixels were being collected at the time when the rest of the TRE values are applicable. Such a reference is useful for scanning sensors, so that collection geometry values can be related to a particular line or framelet in the resulting NITF image.

As with reference times, it is expected that interpolation can be used to obtain parameter values associated with framelets, lines, or pixel locations that are not explicitly provided in the SENSRB TREs—see Z.4.10 Dynamic Value Interpolations. At the same time, the information in the Image Formation Module (Module 4) should be considered to account for discrete changes in the parameters (see Z.5.4.3

[1] The exception is for field values in the Time-Stamped and Pixel-Referenced Data Sets Modules (Modules 12 and 13), which are explicitly related to their own time or pixel location references of applicability—see Z.5.12 Time-Stamped Data Module (Module 12) and Z.5.13 Pixel-Referenced Data Module (Module 13).

Image Formation Counts, Sets, Rates, and First Pixels). For example, a sensor might sequentially image a number of framelets (a rectangular array of pixels), which are then combined to form the Initial Image (a step and stare sensor). In this case, the dynamic parameters might not appropriately be assumed to be continuously changing between two reference pixel locations of applicability. Instead they would be assumed to change in discrete steps as each framelet is collected with a given set of parameter values applying across all pixels in the associated framelet (set).

In relating metadata to particular pixel references, any appropriate pixel coordinate can be used to represent a set of pixels that are collected simultaneously. If needed pixel locations that do not fall in the valid NITF image (including negative pixel coordinates) are allowed. These locations make possible the recording of metadata before and/or after the imaging time—further facilitating some interpolation schemes. Figures Z.5.4-3 and Figure Z.5.4-4 show how pixel location references might be made for both a pushbroom and a step and stare (focal plane array) sensor.

Z.5.6. SENSOR POSITION DATA MODULE (MODULE 06)

The Sensor Position Data Module (Module 06) is the second of two SENSRB modules that do not include a flag to indicate their existence. As with Module 5, the Sensor Position Data Module is required for each and every SENSRB TRE. The module values indicate the absolute geoposition of the sensor at the time of applicability (or when the pixel location of applicability was collected)—see Z.5.5 Reference Time or Pixel Module. The absolute three-dimensional geocoordinates given in the first three fields in this module (indices 06a to 06c) may provide either:

1. The sensor location (the sensor's effective perspective center) or
2. Another nearby location (typically on the platform) relative to which the sensor location may be defined using a set of three-dimensional offsets

The offsets are provided in the last three fields of the module (indices 06d to 06f).

Unless the sensor is stationary, the data in this module is expected to be dynamic, with each SENSRB TRE having different values—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

Uncertainties for the absolute position and the offsets should be provided in the Uncertainty Data (Module 14). These uncertainties are needed to support various exploitation applications. These uncertainties can be fixed values based on engineering estimates or specifications, or—ideally—*they can be taken from the relevant variance and cross-variance data produced and made available by most modern navigational systems*. See Z.5.14.4 Geodetic Coordinate Uncertainties regarding the reporting of uncertainties, if geodetic coordinates are provided in this module.

Z.5.6.1. Sensor or Platform Position

The absolute location (LATITUDE_OR_X, LONGITUDE_OR_Y, and ALTITUDE_OR_Z; indices 06a to 06c) of the sensor or platform will pertain to either the sensor's perspective center (preferred) or to another appropriate location (such as the sensor's or the platform's center of navigation).

If the location is the sensor perspective center (the sensor coordinate system origin), the offsets to the sensor's position (indices 06d to 06f) will be zero. If any other location is used, the offsets will be non-zero (see Z.5.6.2 Sensor Position Offsets below). In either case, the absolute location shall be reported using the method specified by the geodetic coordinate type (GEODETIC_TYPE, index 01h)—see Z.5.1.3 Datum and Unit Declarations.

<u>Geodetic Coordinates</u>. If geodetic coordinates are used, the location is reported in latitude, longitude, and altitude. In this case, the guidance in Z.4.6.1.1 Geodetic Coordinates shall apply. The guidance in Z.4.6.1.2 Elevation and Altitude Datums shall apply to the sensor's or platform's altitude or elevation. The unit of measurement for the altitude or elevation is meters (m) or feet (ft) depending on the length unit system (LENGTH_UNIT, (index 01j). Again, height above ellipsoid (HAE) is strongly recommended as the preferred elevation and altitude datum (ELEVATION_DATUM, index 01i).

<u>Geocentric Coordinates</u>. If geocentric coordinates are used, the location is reported in geocentric Cartesian coordinates (X_E, Y_E, and Z_E). In this case, the guidance in Z.4.6.1.3 Geocentric Coordinates shall apply. The unit of measurement for the three-dimensional location components is meters (m) or feet (ft) depending on the length unit system (LENGTH_UNIT, (index 01j). For a surface-based or airborne sensor, the expected values for each of the three components would be between ±6,400,000 m (±21,000,000 ft). The square root of the sum of the three component values squared ($\sqrt{X_E^2 + Y_E^2 + Z_E^2}$) should never be less than 6,350,000 m (20,840,000 ft)[1].

[1] This is the approximate distance from the center of the earth to the floor of the Arctic Ocean, which is the minimum distance from the center of the ellipsoid to the outside edge of the earth's crust.

Z.5.6.2. Sensor Position Offsets

The sensor position offsets (SENSOR_X_OFFSET (ΔX_p), SENSOR_Y_OFFSET (ΔY_p), and SENSOR_Z_OFFSET (ΔZ_p); indices 06d to 06f) define the location of the sensor's effective perspective center relative to the absolute geoposition defined in the previous three fields (indices 06a to 06c)—see Z.5.6.1 Sensor or Platform Position above.

Offsets of zero (0) indicate that the provided absolute geoposition (defined by indices 06a to 06c) is that of the sensor's perspective center. Non-zero offsets (sometimes called "lever arms") are aligned with the platform coordinate system—see Z.4.6.2 Platform Coordinate System, each offset being along its respective platform axis direction. This is illustrated in Figure Z.5.6-1. The units of measurement for the offsets are meters (m) or feet (ft) depending on the length unit system (LENGTH_UNIT, index 01j).

For non-zero offsets, the absolute sensor location can only be determined correctly if the platform attitude is provided in the Euler Angles Module (Module 7) as defined by the platform Euler angles (PLATFORM_HEADING, PLATFORM_PITCH, and PLATFORM_ROLL; indices 07f to 07h)—see Z.5.7.2 Platform Euler Angles. If the platform attitude is not provided, the total length of the offsets ($\sqrt{\Delta_{X_P}^2 + \Delta_{Y_P}^2 + \Delta_{Z_P}^2}$) becomes an additional contributor to the overall sensor location uncertainty.

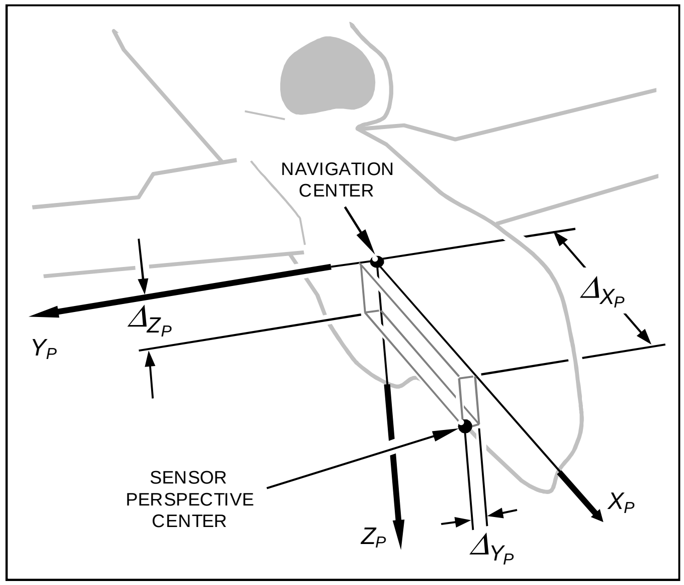


Figure Z.5.6-1. Sensor Position Offsets. The geolocation of the senor's perspective center can be defined relative to another absolute geoposition (such as the platform's navigation center, as shown here). The offsets (ΔX_p, ΔY_p, and ΔZ_p) are always aligned with their respective platform axes (X_P, Y_P, and Z_P).

Z.5.7. EULER ANGLES MODULE (MODULE 07)

The sensor coordinate system's attitude (or orientation) can be provided in a SENSRB TRE by any of three methods[1]:

1. Euler Angles—Module 7
2. Unit Vectors (or direction cosines)—Module 8[2]
3. Quaternions—Module 9

For a content level value (CONTENT_LEVEL, index 01f) of "4" or higher (see Z.5.1.2 Application-Required Content Level) one of the three above-listed modules (Module 7, 8, or 9) must exist and provide the necessary data to define the sensor's attitude. Furthermore, unless the sensor is stationary, the metadata values in these modules are likely to be dynamic. In other words, if multiple SENSRB TREs exist for an image segment, it would not be unexpected for the field values in these modules to be different in the various TREs—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

The Euler angles method implemented in Module 7 is commonly used to define the sensor's attitude. It has the advantage of being compact and intuitively consistent with the gimballed systems used by many electro-optical sensors. Euler angles are also in harmony with various existing international standards. On the other hand, the use of Euler angles can lead to ambiguities and misinterpretations, which would not be as prevalent if using the unit vectors or quaternions methods. (Unit vectors are preferred.)

The sensor Euler angles (SENSOR_ANGLE_1, SENSOR_ANGLE_2, and SENSOR_ANGLE_3; indices 07b to 07d) are defined differently depending on the type of sensor angle rotations (SENSOR_ANGLE_ MODEL, index 07a). Furthermore, they can be reported relative to either the local-level (NED)[3] or the platform coordinate system—depending on the sensor-angles-relative-to-platform flag (PLATFORM_ RELATIVE, index 07e). Z.5.7.3 Sensor Euler Angles provides more details.

The platform Euler angles (PLATFORM_ HEADING, PLATFORM_PITCH, and PLATFORM_ROLL; indices 07f to 07h) are *always* reported relative to the local-level coordinate system, even when the geodetic coordinate type (GEODETIC_TYPE, index 01h) has the value of "C" for geocentric Cartesian—see Z.5.1.3 Datum and Unit Declarations. Z.5.7.2 Platform Euler Angles provides more details.

Even when other methods are used to define the sensor's attitude (Module 8 or 9), providing the equivalent Euler angles[4] in Module 7 has several benefits. For example, the Euler angles provide an intuitive picture of the collection geometry. Fields in Module 7 can provide the platform attitude, which is not available elsewhere but may be needed for various reasons (for example, to determine the correct sensor location—see Z.5.6.2 Sensor Position Offsets). The redundancy of information can also provide confidence in the attitude data. Finally, attitude uncertainties may be more readily available in terms of platform Euler angles and/or the gimbal angles, which are related to the sensor Euler angles.

The angular units used in this module depend on the value of the angular unit type (ANGULAR_UNIT index 01k)—see Z.4.4.2 Angular Units. The value range limits for the fields are defined in degrees—the preferred units. Equivalent limits apply in the alternative units of radians or semi-circles.

Uncertainties for the platform and sensor Euler angles should be provided in the Uncertainty Data Module (Module 14). These uncertainties are needed to support various exploitation applications. These

[1] If field values are provided in two or more of the attitude modules and the values are in conflict (inconsistent between modules), precedence will be given to the values in the modules using the following order: Module 8 taking precedence over the other two, and Module 9 taking precedence over Module 7.

[2] The unit vectors of Module 8 define the attitude of the image coordinate system rather than the sensor coordinate system—see Z.4.6.3 Sensor Coordinate System and Z.4.6.5 Image Coordinate System.

[3] See the footnote associated with "Geodetic Coordinate Type" in Z.5.1.3 Datum and Unit Declarations.

[4] Equations for computing equivalent Euler angles, unit vectors, and quaternions are available in Z.6.3 Attitude Representation Conversions. The attitude conversions can also be made using the SENSRB Data Conversion Tool (see Z.6.4 SENSRB Data Conversion Tool).

uncertainties can be fixed values based on engineering estimates or specifications, or they may be extracted from covariance data produced by many modern inertial measurement units.

The SENSRB Data Conversion Tool provides conversions between the various sensor orientation modules. It also provides various visualization tools to further aid with the interpretation of the Euler angle definitions used in this module—see Z.6.4 SENSRB Data Conversion Tool.

Z.5.7.1. Positive Rotation Definition (Right-Hand Rule)

Euler angles are often defined using rotations and angular measurements. Figure Z.5.7-1 illustrates the definition of a positive rotation or angle measurement about an axis or vector according to the right-hand rule. With the thumb of the right hand pointing in the positive axis or vector direction, the fingers curl in the positive rotation or angle measurement direction. This positive-direction definition is used consistently within Module 7 (and elsewhere in SENSRB)

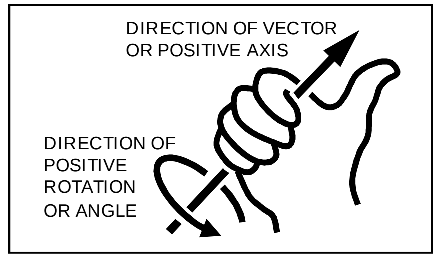


Figure Z.5.7-1. Right-Hand Rule for Positive Rotations or Angles.

Z.5.7.2. Platform Euler Angles

In SENSRB, the attitude of the platform coordinate system is reported explicitly only in the Euler Angles Module (Module 7). The Euler angles used to describe the platform attitude are PLATFORM_HEADING, PLATFORM_PITCH, and PLATFORM_ROLL (indices 07f to 07h). These are defined in the same way for both SENSRA and SENSRB. Regardless of the datum type (GEODETIC_TYPE, index 01h), these angles are always reported relative to the local-level or NED coordinate system[1] (see Z.4.6.1.4 Local-Level Coordinate System).

If the Sensor-Angles-Relative-To-Platform flag is set to "Y," the Euler angles of the sensor will be defined relative to the platform. In that case, the platform Euler angles must be provided with meaningful values. Otherwise (if flag 07e is set to "N"), the platform Euler angle fields can be unspecified (dashes), however this would limit the usability of the TRE.

The three platform Euler angles can be thought of in either of two equivalent ways: in terms of three rotations or in terms of three static angles.

<u>Sequential Rotations.</u> The three sequential rotations used to describe the platform Euler angles are illustrated in Figure Z.5.7-2 on the following page. The figure also introduces an auxiliary set of axes (*P*, *Q*, and *R*) useful in defining the angles and rotations. The *P*, *Q*, and *R* axes are aligned with the local-level NED coordinate system. They form the *P′-Q′-R′* system after being rotated about the +*R* (or +*D*) axis by the platform heading angle (ψ_P)—defined to be zero when the platform pitch is ±90°. A second and a third rotation subsequently define the *P″-Q″-R″* and *P‴-Q‴-R‴* systems. The second rotation is about the +*Q′* axis by the platform pitch angle (θ_P). The third is about the +*P″* (or +X_P) axis by the platform roll angle (ϕ_P). The final system is aligned with the platform coordinate system (X_P, Y_P, and Z_P)—see Z.4.6.2 Platform Coordinate System.

<u>Static Angles.</u> Alternatively, the three platform Euler angles used in SENSRA and SENSRB can be defined as three static angles as delineated here. These definitions use the auxiliary axes system (*P′-Q′-R′*) introduced in Figure Z.5.7-2. The *P′-Q′-R′* system is said to be formed from the P-Q-R system (which is aligned with the *N-E-D* system) by rotating it about the +*D* (or +*R*) axis until +*P′* is aligned with the projection of the platform's positive roll axis (+X_P) onto the local horizontal (*N-E* or *P-Q*) plane (see Figure Z.5.7-2 (c and d)). If the local-level projection of X_P is of zero length (because X_P is pointing straight up or down), the *P′-Q′-R′* system is defined as remaining the same as the *P-Q-R* (and also the *N-E-D*) axes system. The Euler angles are defined positive according to the right-hand rule—see Z.5.7.1 Positive Rotation Definition (Right-Hand Rule).

[1] See the footnote associated with "Geodetic Coordinate Type" in Z.5.1.3 Datum and Unit Declarations.

Platform Heading (ψ_P): The angle from the positive local-level north axis (+*N*) to the +*P′* axis; measured about the local down (+D) axis, positive in the direction from north toward east. Heading values range from zero—True North—to almost 360°. When the platform pitch is ±90°, the platform heading is defined to be zero.

Platform Pitch (θ_P): The angle from the +*P′* axis to the positive roll axis (+X_P); measured about the +*Q′* axis, positive when the positive platform roll axis (+X_P) points above the horizontal plane. Pitch values range from +90° (straight up) to -90° (straight down).

Platform Roll (ϕ_P): The angle from the +*Q′* axis to the positive pitch axis (+Y_P); measured about the positive platform roll axis (+X_P), positive when the positive pitch axis (+Y_P) points below the horizontal plane (provided that platform pitch is not ±90°). Roll values range from -180° to +180°. For platform pitch of +/-90, the roll will be defined as the angle between the local east direction and the transverse axis (positive toward north for +90 and positive toward south for -90).

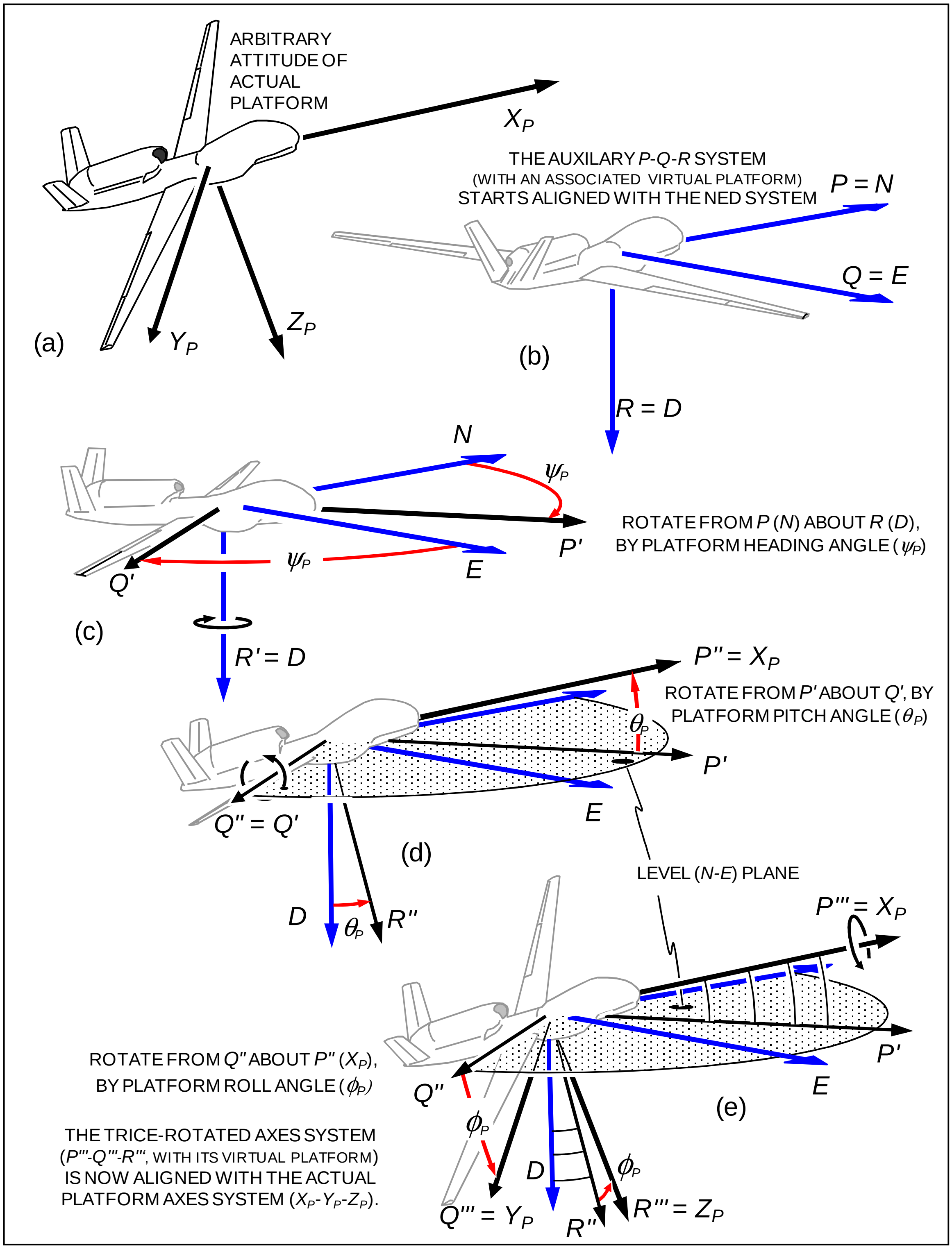


Figure Z.5.7-2. Platform Attitude. An arbitrary platform attitude (a) is defined in SENSRA or SENSRB relative to the local-level coordinate system by three angles or rotations—heading (ψ_P), pitch (θ_P), and roll (ϕ_P). Here, the attitude is realized by rotating a virtual platform associated with an auxiliary axes system (*P-Q-R*, initially aligned with the NED system) (b) by the heading (c), the pitch (d), and the roll (e) angles. In this illustration all the axes and rotations shown are in the positive direction—see Z.5.7.1 Positive Rotation Definition (Right-Hand Rule).

Z.5.7.3. Sensor Euler Angles

The Euler angles used to define the attitude of the sensor coordinate system have the potential to be chosen according to the physical mounting of the sensor and its gimbal system. The three generic angles are the first (α), second (β), and third (γ) sensor rotation angles—SENSOR_ANGLE_1, SENSOR_ANGLE_2, and SENSOR_ANGLE_3 (indices 07b to 07d), respectively.

The SENSRB sensor Euler angles are not the same as the sensor angles defined in SENSRA. The SENSRA angles, as defined for that TRE, cannot completely characterize the attitude of a sensor coordinate system. This inadequacy was the initial impetus for the development of SENSRB.

The sensor Euler angles in SENSRB are defined according to the type of sensor angle rotations (SENSOR_ANGLE_MODEL, index 07a)—as described below in Z.5.7.3.1 Sensor Angle Model. Rotation-order-independent static angle definitions for the sensor Euler angles are given in Z.5.7.3.2 Sensor Angle Definitions.

Depending on the Sensor-Angles-Relative-To-Platform flag (PLATFORM_RELATIVE, index 07e), the sensor Euler angles can be reported in one of two ways:

1. Relative to the platform coordinate system, when the flag (index 07e) is set to “Y”, or
2. Relative to the local-level coordinate system,[1] when the flag (index 07e) is set to “N”

as described in Z.5.7.3.3 Sensor Angle Reference.

Z.5.7.3.1. Sensor Angle Model

Various sensor angle rotation types can be accommodated in the Euler Angles Module. These sensor angle models are meant to facilitate the use of the readily available gimbal angles associated with the most common sensor mounting systems. As illustrated in Figure Z.5.7-3, the different models reflect the differences in gimbal orderings (outermost to innermost) in various systems. The numerically-coded value stored in SENSOR_ANGLE_MODEL (index 07a) indicates which model is used for the subsequent sensor Euler angles.

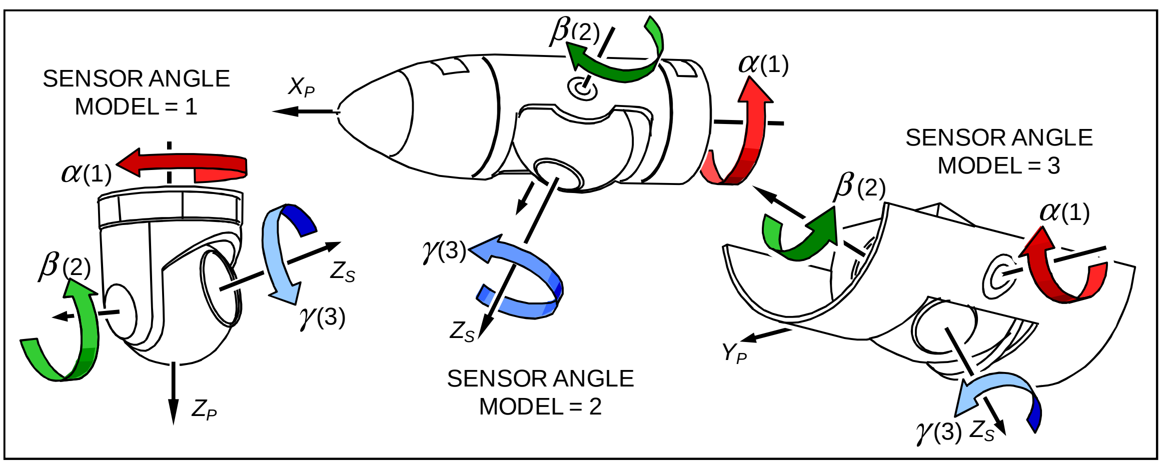


Figure Z.5.7-3. Three Sensor Angle Models. The Euler angle rotations are shown relative to the platform coordinate systems. Rotations could be made relative to the local-level coordinate system by replacing X_P, Y_P, and Z_P with N, E, and D; respectively. In the illustrations all the axes and rotations shown are in the positive directions—see Z.5.7.1 Positive Rotation Definition (Right-Hand Rule).

The sensor angle models are further defined in Table Z.5-2 on the following page. The table indicates the initial (zero-rotation) orientation of the sensor coordinate system as well as the axes for the rotation sequence. The table makes use of the auxiliary coordinate systems shown in Figure Z.5.7-4. The

[1] See the footnote associated with “Geodetic Coordinate Type” in Z.5.1.3 Datum and Unit Declarations.

auxiliary system (*P*-*Q*-*R*) corresponds to the zero-rotation sensor coordinate system (X_{S0}-Y_{S0}-Z_{S0}). Its initial alignment to the reference system (platform or local-level coordinate system) is as specified in the table. Subsequent auxiliary systems—(P'-Q'-R'), (P''-Q''-R''), and (P'''-Q'''-R''')—are created by rotating the auxiliary systems as indicated in the table. The final auxiliary system (P'''-Q'''-R''') is aligned with the attitude of the sensor coordinate system ($P''' = X_S$; $Q''' = Y_S$, and $R''' = Z_S$).

Table Z.5.7-1. Sensor Angle Model Initial Axis Alignments and Axes of Rotation.

SENSOR_ ANGLE_ MODEL	Zero-Rotation Alignment[a]			Axis of Rotation		
	$+X_P$[b]	**$+Y_P$**[c]	**$+Z_P$**[d]	**1st Angle (α)**	**2nd Angle (β)**	**3rd Angle (γ)**
1	$+Z_{S_0}$ (+*R*)	$+X_{S_0}$ (+*P*)	$+Y_{S_0}$ (+*Q*)	$+Z_P$[d] (+*Q* or $+Y_{S_0}$)	$+P'$	$+R''$ ($+Z_S$)
2	$-Y_{S_0}$ (-*Q*)	$+X_{S_0}$ (+*P*)	$+Z_{S_0}$ (+*R*)	$+X_P$[b] (-*Q* or $-Y_{S_0}$)	$+P'$	$+R''$ ($+Z_S$)
3	$-Y_{S_0}$ (-*Q*)	$+X_{S_0}$ (+*P*)	$+Z_{S_0}$ (+*R*)	$+Y_P$[c] (+*P* or $+X_{S_0}$)	$-Q'$	$+R''$ ($+Z_S$)

[a] Because the platform and the sensor coordinate systems may have different origins, alignment means parallel to and in the same direction as; it does not necessarily mean that the axes are collinear.
[b] For an attitude relative to the local-level coordinate system, substitute *N* (true north) for the platform's X_P.
[c] For an attitude relative to the local-level coordinate system, substitute *E* (east) for the platform's Y_P.
[d] For an attitude relative to the local-level coordinate system, substitute *D* (down) for the platform's Z_P.

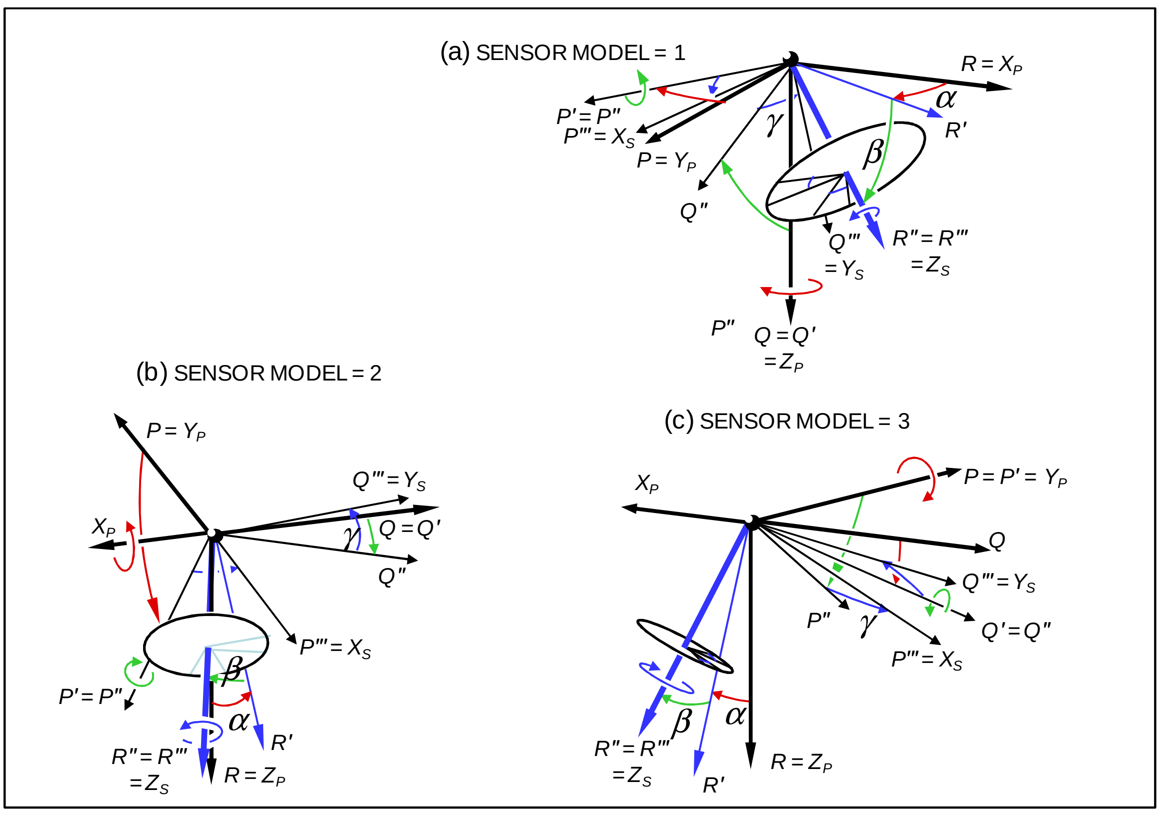


Figure Z.5.7-4. Sensor Euler Angles.

The sensor angle rotation types described in Table Z.5-2 and shown in Figure Z.5.7-4 are analogous to the following human views relative to an airborne platform:

<u>Model 1:</u> The view is like that of a pilot looking forward from the cockpit. The first rotation (α) is like the pilot looking left or right. The second rotation (β) is like the pilot nodding his or her previously-turned head up or down. The final rotation (γ) is like the pilot cocking his or her turned and nodded head sideways. This is usually referred to as yaw, pitch, and roll.

Model 2: The view is like that of a prone, downward-looking bombardier with his or her feet toward the aircraft tail and the top of his or her head toward the aircraft nose. The first rotation (α) is like the bombardier swiveling his or her neck left or right. The second rotation (β) is like the bombardier nodding his or her previously-swiveled head foreword or aftward. The final rotation (γ) is like the bombardier cocking his or her swiveled and nodded head sideways.

Model 3: The view is like that of the same bombardier described for Model 2. However, the first rotation (α) is like the bombardier nodding his or her initially downward-looking head foreword or aftward. The second rotation (β) is like the bombardier swiveling his or her now-nodded head left or right (about the nodded-head axis not the prone-body axis). Again, the final rotation (γ) is like the bombardier cocking his or her nodded and swiveled head sideways.

The first model works well for various turreted and some podded systems, such as that on the MQ-1 Predator unmanned air vehicle (see Figure Z.5.7-5) and some FLIR (forward-looking infrared) pods. The

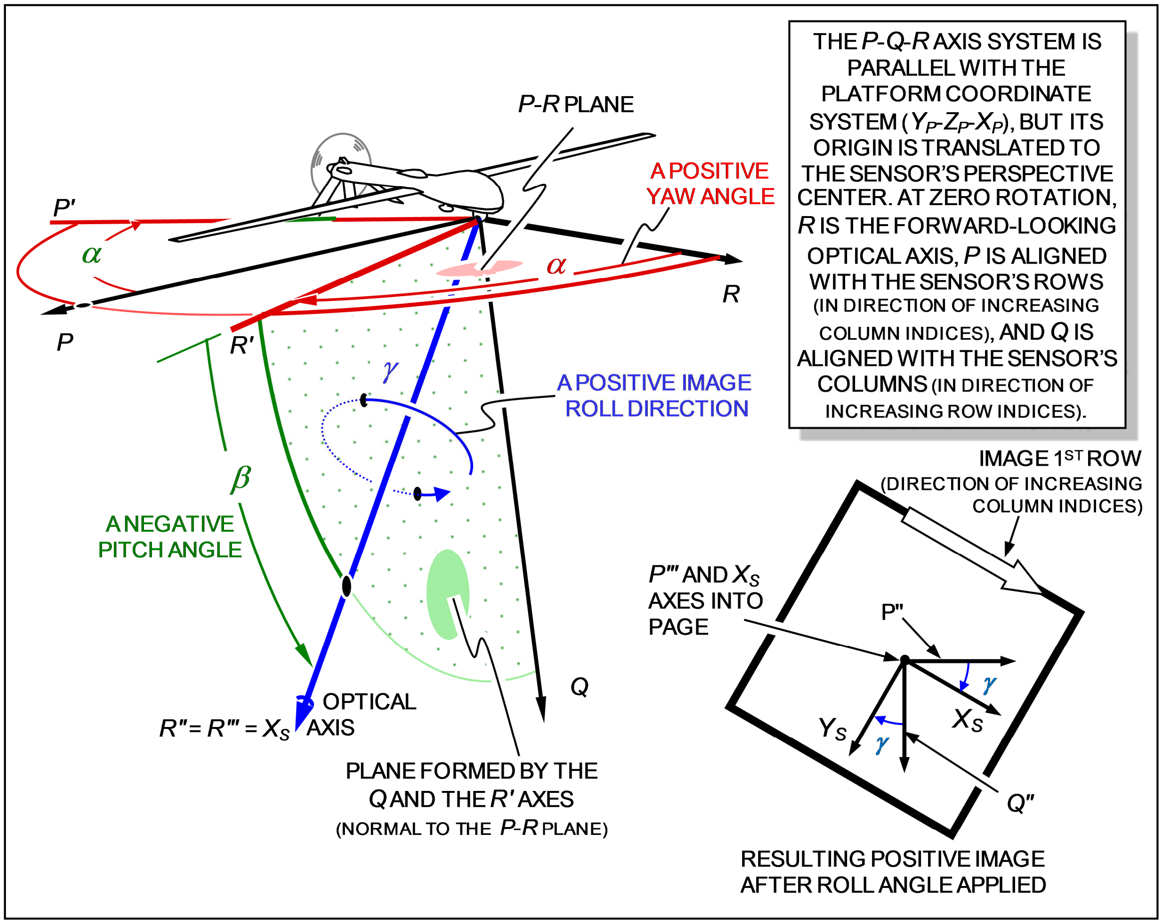


Figure Z.5.7-5. Application of the Sensor Euler Angles (Model 1). A series of three Euler angle rotations define the sensor coordinate system's orientation relative to the platform coordinate system. A zero-rotation, optical axis (+*R*) starts parallel with the aircraft +X_P axis; the sensor's row-aligned (increasing column indices) axis (+*P*) parallel with the platform's +Y_P axis; and the sensor's column-aligned (increasing row indices) axis (+*Q*) parallel with the platform's +Z_P axis. The first rotation (α) is about the +*Q* axis (in the *P-R* plane)—positive toward the right wing. The second rotation (β) is about the new +*P'* axis resulting from the first rotation (and is in the indicated plane formed by the *Q* and *R'* axes), such that a rotation from the *P-R* plane toward the platform belly is *negative*. The third rotation (γ) is about the resulting optical axis (+Z_S) or in the resulting image plane, such that a positive rotation is as shown (bottom right).

second model works well for various podded and nose-mounted systems; such as the SHARP (Shared Airborne Reconnaissance Pod), the RQ-4 Global Hawk's electro-optical sensor (see Figure Z.5.7-6), and the U-2's SYERS-2 (Senior Year Electro-optical Reconnaissance System). The third model was included to accommodate the U-2's OBC (Optical Bar Camera) (see Figure Z.5-11 on the following page).

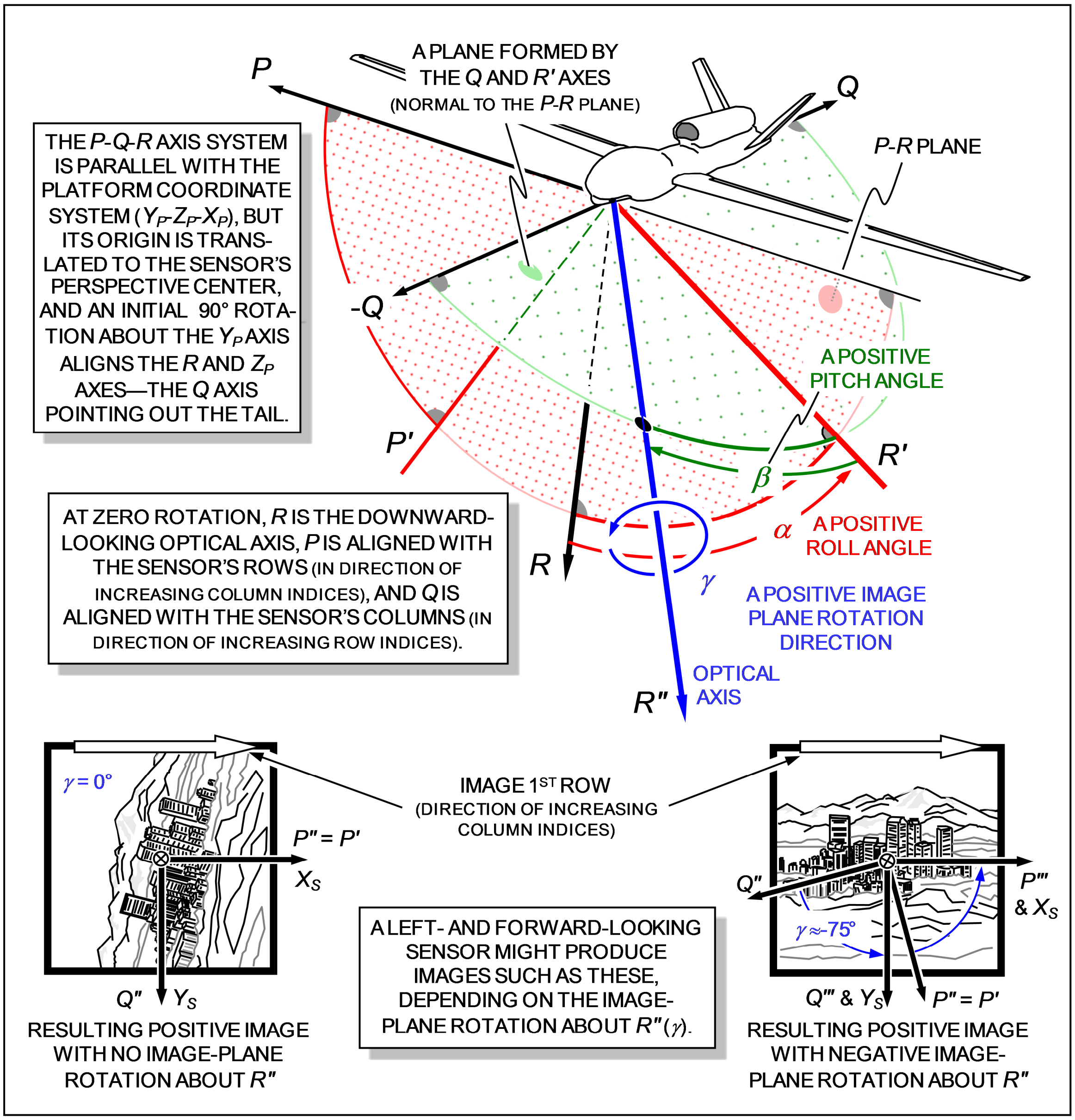


Figure Z.5.7-6. Application of the Sensor Euler Angles (Model 2). A series of three Euler angle rotations define the sensor coordinate system's orientation relative to the platform coordinate system. A zero-rotation, optical axis (+*R*) starts parallel with the aircraft +Z_P axis; the sensor's row-aligned (increasing column indices) axis (+*P*) parallel with the platform's +Y_P axis; and the sensor's column-aligned (increasing row indices) axis (+*Q*) parallel with but in the opposite direction of the platform's +X_P axis. The first rotation (α) is about the -*Q* axis (in the *P-R* plane)—positive toward the left wing. The second rotation (β) is about the +*P′* axis resulting from the first rotation (and is in the indicated plane formed by the *Q* and *R′* axes), such that a rotation from the *P-R* plane forward toward the platform nose is positive. The third rotation (γ) is about the resulting optical axis (Z_S) or in the resulting image plane, such that a positive rotation is as shown for the images in the lower left and right.

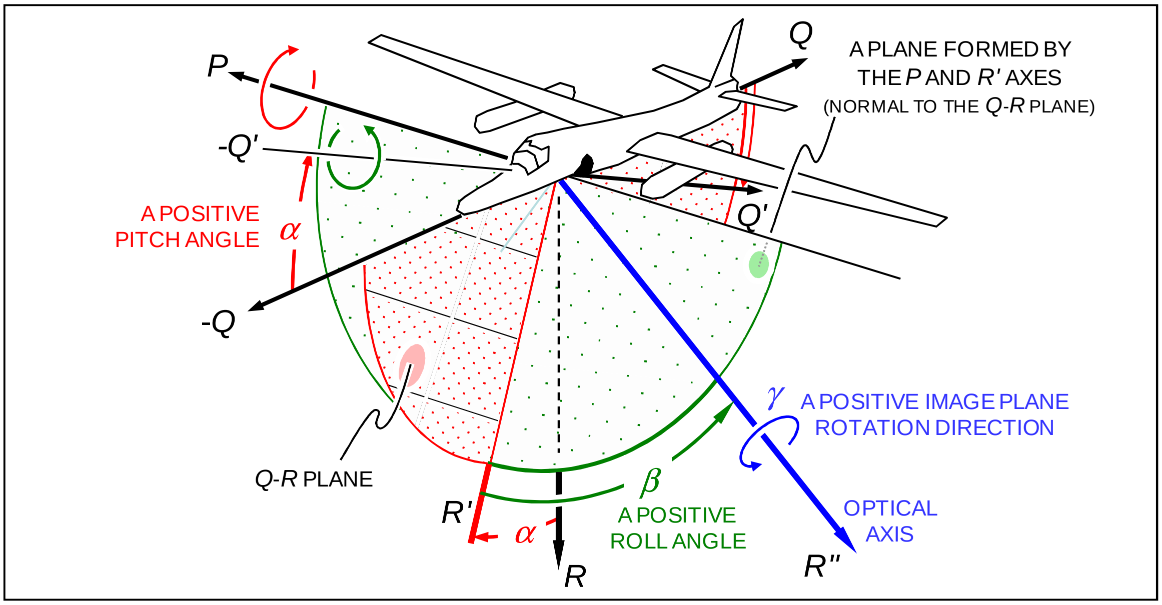


Figure Z.5.7-7. Application of the Sensor Euler Angles (Model 3). A series of three Euler angle rotations define the sensor coordinate system's orientation relative to the platform coordinate system. A zero-rotation, optical axis (+*R*) starts parallel with the aircraft $+Z_P$ axis; the sensor's row-aligned (increasing column indices) axis (+*P*) parallel with the platform's $+Y_P$ axis; and the sensor's column-aligned (increasing row indices) axis (+*Q*) parallel with but in the opposite direction of the platform's $+X_P$ axis. The first rotation (α) is about the +*P* axis (in the *Q-R* plane)—positive forward toward the nose. The second rotation (β) is about the -*Q′* axis resulting from the first rotation (and is in the indicated plane formed by the *P* and *R′* axes), such that a rotation from the *P-R* plane toward the left wing is positive. The third rotation (γ) is about the resulting optical axis (Z_S) or in the resulting image plane, such that a positive rotation is as [will be] shown.

If a sensor system has an initial (zero-rotation) alignment different from that specified in Table Z.5-2, adjustments must be made to the gimbal angle values to account for those differences.

Some sensor systems might have additional Euler angles associated with other intermediate coordinate systems. In these cases the rotations can be aggregated. The resulting aggregate Euler angles are acceptable as long as they result in a set of rotations equivalent to one of the above-listed models.

Alternatively, the sensor coordinate system's attitude can be provided in the preferred Unit Vectors Module (Module 8). However, even if Module 8 is used, it is desirable to provide the nominal Euler angles as computed using the equations in Z.6.3 Attitude Representation Conversions.

It is believed that the provided models will facilitate most common gimbal systems. If additional models are needed, approval can be sought to register these models with the NTB—see Z.4.1 NTB-Registered Field Values.

Z.5.7.3.2. Sensor Angle Definitions

The three sensor Euler angles used in SENSRB can be defined as three static angles as delineated in this sub-section. These definitions depend on the type of sensor angle rotation (SENSOR_ANGLE_MODEL, index 07a). The static angle definitions facilitate the use of the Euler angles even when a gimbaled system or a mounting does not fit one of the subscribed types of sensor angle rotations. These definitions are especially useful when the sensor Euler angles are referenced to the local-level coordinate system—see Z.5.7.3.3 Sensor Angle Reference.

The sensor Euler angle definitions given here use the auxiliary axes systems (*P′-Q′-R′*) introduced in Figure Z.5.7-4. The *P′-Q′-R′* system is formed from the *P-Q-R* system, which is aligned with the reference (platform or local-level) coordinate system according to the zero-rotation columns in Table Z.5.7-1. The origin of the auxiliary coordinate systems is the sensor's perspective center.

The rotation used to form the *P′-Q′-R′* coordinate system depends of the specified type of sensor angle rotation (SENSOR_ANGLE_MODEL, index 07a) as defined below. In every case, if the second sensor rotation angle (SENSOR_ANGLE_2, index 07c) is ±90°, the *P′-Q′-R′* coordinate system is defined to be the same as the original *P-Q-R* coordinate system.

Model 1: If the SENSOR_ANGLE_MODEL is set to "1", the following definitions apply. The *P′-Q′-R′* system is formed by rotating the *P-Q-R* system about the +*Q* axis (aligned with either the +*D* or $+Z_P$ axis) until the +*R′* is aligned with the projection of the sensor's optical axis ($+Z_S$) onto the *P-R* plane (see Figures Z.5-8 (a) and Z.5-9).

First Sensor Rotation Angle (azimuth): The angle from the +*R* to the +*R′* axis; measured about the sensor's zero-rotation column-aligned (+*Q*) axis, positive in the direction from +*R* toward +*P*. Azimuth values range from nearly -180°—southward to the west (or backward to the left)—to +180°—southward to the east (backward to the right). When the sensor's second rotation angle (elevation) is ±90°—looking parallel to the *D* axis—straight up or down (or parallel to the platform's Z_P axis—straight through the platform belly or top), the first sensor rotation angle is defined to be zero.

Second Sensor Rotation Angle (elevation): The angle from the +*R′* axis to the optical axis ($+X_S$); measured about the +*P′* axis, positive when the optical axis ($+X_S$) points above the horizontal (or platform-level) *P-R* plane. Elevation values range from -90°—straight down (or straight through the platform's belly) to +90°—straight up (or straight through the platform's top).

Third Sensor Rotation Angle (roll): The angle from the +*P′* axis (in the horizontal or platform-level, *P-R* plane) to the sensor's row-aligned, increasing-column-indices direction ($+X_S$); measured about the optical axis ($+Z_S$), positive when the positive row-aligned axis ($+X_S$) points below the horizontal (or platform-level) plane. Roll values range from -180° to +180°. When the sensor's second rotation angle (elevation) is ±90° (the sensor's first rotation angle (azimuth) being defined as 0°), the sensor's third rotation angle (roll) is the angle from the +*P* to the sensor's row-aligned ($+X_S$) axis about the optical axis ($+Z_S$), with the sense determined by the previously-described right-hand rule.

Model 2: If the SENSOR_ANGLE_MODEL is set to "2", the following definitions apply. The *P′-Q′-R′* system is formed by rotating the *P-Q-R* system about the -*Q* axis (aligned with either the +*N* or $+X_P$ axis) until the +*R′* lies in the plane formed by the sensor's optical axis ($+Z_S$) and the *Q* axis (see Figures Z.5-8 (b) and Z.5-10).

First Sensor Rotation Angle (roll): The angle from the +*R* to the +*R′* axis; measured about the sensor's zero-rotation column-aligned (-*Q*) axis, positive in the direction from +*R* toward -*P*—westward (or left) looking. Roll values range from nearly -180°—upward to the east (right)—to +180°—upward to the west (left). When the sensor's second rotation angle (elevation) is ±90°—looking parallel to the *N* axis—due north or south (or parallel to the platform's X_P axis—straight through the platform nose or tail), the first sensor rotation angle is defined to be zero.

Second Sensor Rotation Angle (pitch): The angle from the +*R′* axis to the optical axis ($+X_S$); measured about the +*P′* axis, positive when the optical axis ($+X_S$) points north (forward) of the east-west (or flight-normal) *P-R* plane. Pitch values range from -90°—due south (or straight past the platform's tail) to +90°—due north (or straight past the platform's nose).

Third Sensor Rotation Angle (image roll): The angle from the +*P′* axis (in the east-west or flight-normal, *P-R* plane) to the sensor's row-aligned, increasing-column-indices direction ($+X_S$); measured about the optical axis ($+Z_S$), positive when the positive row-aligned axis ($+X_S$) points south (aft) of the *P-R* plane. Image roll values range from -180° to +180°. When the sensor's second rotation angle (pitch) is ±90° (the sensor's first rotation angle (roll) being defined as 0°), the sensor's third rotation angle (image roll) is the angle from the +*P* to the sensor's row-aligned ($+X_S$) axis about the optical axis ($+Z_S$), with the sense determined by the previously-described right-hand rule.

Model 3: If the SENSOR_ANGLE_MODEL is set to "3", the following definitions apply. The *P′-Q′-R′* system is formed by rotating the *P-Q-R* system about the +*P* axis (aligned with either the +*E* or $+Y_P$ axis) until the +*R′* lies in the plane formed by the sensor's optical axis ($+Z_S$) and the *P* axis (see Figures Z.5-8 (c) and Z.5-11).

First Sensor Rotation Angle (pitch): The angle from the +*R* to the +*R′* axis; measured about the sensor's zero-rotation row-aligned (+*P*) axis, positive in the direction northward (or forward) from +*R* toward -*Q*. Pitch values range from nearly -180°—upward from the south (tail)—to +180°—upward from the north (nose). When the sensor's second rotation angle (roll) is ±90°—looking parallel to the *E* axis—due west or east (or parallel to the platform's Y_P axis—straight through the platform left or right wing), the first sensor rotation angle is defined to be zero.

Second Sensor Rotation Angle (roll): The angle from the +*R′* axis to the optical axis ($+X_S$); measured about the -*Q′* axis, positive when the optical axis ($+X_S$) points west (left) of the *Q-R* plane. Roll values range from -90° (due east or straight through the platform's right wing) to +90° (due west or straight through the platform's left wing).

Third Sensor Rotation Angle (image roll): The angle from the +*Q′* axis (in the *Q-R* plane) to the sensor's column-aligned, increasing-row-indices direction ($+Y_S$); measured about the optical axis ($+Z_S$), positive when the positive column-aligned axis ($+Y_S$) points west (left) of the *P-R* plane. Image roll values range from -180° to +180°. When the sensor's second rotation angle (roll) is ±90° (the sensor's first rotation angle (pitch) being defined as 0°), the sensor's third rotation angle (image roll) is the angle from the +*Q* to the sensor's row-aligned ($+Y_S$) axis about the optical axis ($+Z_S$), with the sense determined by the previously-described right-hand rule.

Z.5.7.3.3. Sensor Angle Reference

In the Euler Angles Module (Module 7), the attitude of the sensor coordinate system can be provided relative to either the platform or the local-level (NED) coordinate system[1] (see Z.4.6.2 Platform Coordinate System and Z.4.6.1.4 Local-Level Coordinate System). The reference system is indicated by the Sensor-Angles-Relative-To-Platform flag (PLATFORM_RELATIVE, index 07e). If the flag is set to "Y", the sensor Euler angles are relative to the platform coordinate system. If the flag is set to "N", the sensor Euler angles are relative to the local-level coordinate system.

The platform-relative angles are useful for systems where the sensor pointing information is determined by a fixed mounting on a platform or is derived from gimbal angles (with the gimbaled system mounted to the platform). The concluding paragraphs of Z.5.7.3.1 Sensor Angle Model should be considered in these cases.

An increasing number of sensor systems employ their own inertial measurement units and are capable of reporting their own attitude relative to the local-level coordinate system. If such a system is used, the Euler angles may not be relative to the platform. The sensor's inertial measurement unit is still likely to provide Euler angles in terms of one of the models described in the previous sub-sections. The application of the local-level reference system can be made by substituting *N*, *E*, and *D* for X_P, Y_P, and Z_P; respectively, throughout the previous sub-sections.

[1] See the footnote associated with "Geodetic Coordinate Type" in Z.5.1.3 Datum and Unit Declarations.

Z.5.8. UNIT VECTORS MODULE (MODULE 08)

The image coordinate system's (and therefore the sensor coordinate system's) attitude can be provided in a SENSRB TRE by any of three equivalent methods:

1. Euler Angles—Module 7[1]
2. Unit Vectors (or direction cosines)—Module 8
3. Quaternions—Module 9[1]

For a content level value (CONTENT_LEVEL, index 01f) of "4" or higher (see Z.5.1.2 Application-Required Content Level) one of the three above-listed modules (Module 7, 8, or 9) must exist and provide the necessary data to define the sensor's attitude. Furthermore, unless the sensor is stationary, the metadata values in these modules are likely to be dynamic. In other words, if multiple SENSRB TREs exist for an image segment, it would not be unexpected for the field values in these modules to be different in the various TREs—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

The unit vector method implemented in Module 8 is the preferred method for providing the image coordinate system (and sensor coordinate system) orientation. The image coordinate system (rather than the sensor coordinate system) is commonly used in photogrammetric applications. The image coordinate system is related to the sensor coordinate system by a simple permutation of the axes—see Z.4.6.3 Sensor Coordinate System and Z.4.6.4 Image Coordinate System. The use of the unit vector method and the image coordinate system is consistent with the abstract formulations contained in the Community Sensor Model Working Group's implementation guidance documents for geopositioning from frame, pushbroom, and whiskbroom sensors. Unit vectors offer intuitive pointing information relative to the reference coordinate system. They also provide built-in redundancy for error checking.[2]

If field values are provided in two or more of the attitude modules and the values are in conflict (inconsistent between modules), precedence will be given to the values in the modules using the following order—Module 8 taking precedence over the other two, and Module 9 taking precedence over Module 7.

The unit vector coordinates in Module 8 define the attitude of the image coordinate system relative to the sensor's local-level coordinate system or the geocentric (earth-centered, earth-fixed) coordinate system, depending on the value of the GEODETIC_TYPE (index 01h)—see Z.5.1.3 Datum and Unit Declarations. These two reference systems are described in Z.4.6.1.4 Local-Level Coordinate System and Z.4.6.1.3 Geocentric Coordinates, respectively.

The nine coordinates provided in the module are for the unit vectors aligned with the three positive image axes (X_I, Y_I, and Z_I) and are given in terms of the reference coordinate system. Figure Z.5-12 on the following page illustrates the three unit vectors defining the image coordinate system axes in relationship to the local-level (*N-E-D*) coordinate system. The coordinates of these unit vectors in the reference system provide the direction cosines for the image system attitude. As such they are the elements of the 3×3 rotation matrix used to transform between the sensor's local-level and image coordinate systems.

The first three fields (ICX_NORTH_OR_X, ICX_EAST_OR_Y, and ICX_DOWN_OR_Z; indices 08a to 08c) give the coordinates (or direction cosines) of the image coordinate system's $+X_I$ axis in terms of either the *N-E-D* or X_E-Y_E-Z_E system. The next three fields (ICY_NORTH_OR_X, ICY_EAST_OR_Y, and ICY_DOWN_OR_Z; indices 08d to 08f) give the coordinates (direction cosines) of the image coordinate system's $+Y_I$ axis in terms of either the *N-E-D* or X_E-Y_E-Z_E system. The final three fields (ICZ_NORTH_

[1] The Euler angles and quaternions of Modules 7 and 9, respectively, define the attitude of the sensor coordinate system rather than the image coordinate system—see Z.4.6.3 Sensor Coordinate System and Z.4.6.5 Image Coordinate System.

[2] The values of the nine unit vector coordinates should form a 3×3 orthonormal matrix, where each row and column has a magnitude of unity and is orthogonal to the other rows or columns.

OR_X, ICZ_EAST_OR_Y, and ICZ_DOWN_OR_Z; indices 08g to 08i) give the coordinates (direction cosines) of the image coordinate system's $+Z_I$ axis in terms of either the *N-E-D* or X_E-Y_E-Z_E system.

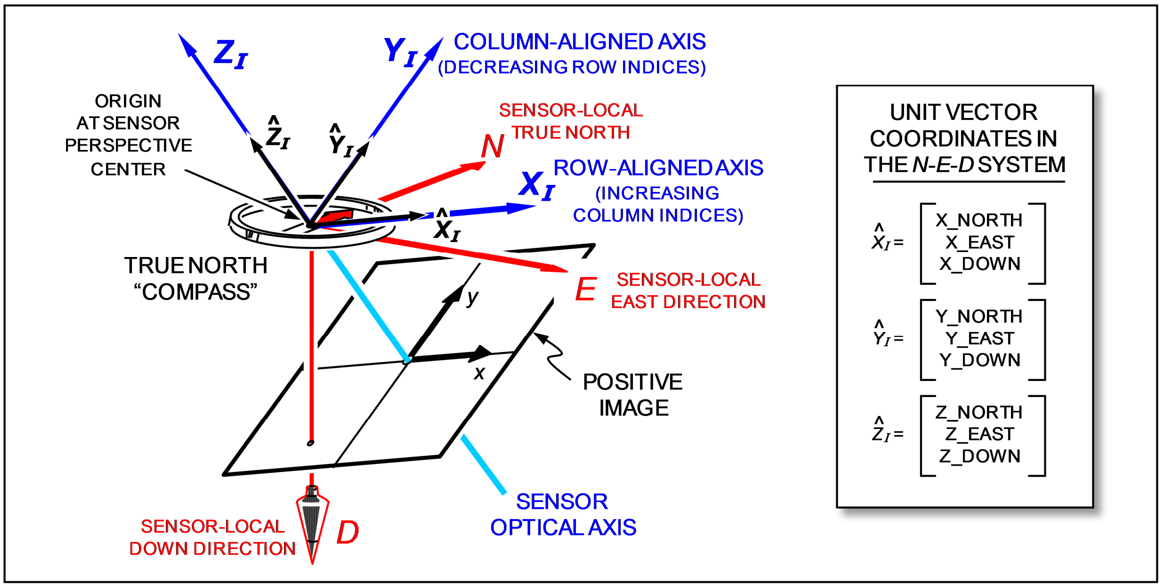


Figure Z.5.8-1. Image Coordinate System Relative to the Local-Level (NED) Coordinate System. The unit-length vectors defining the line-of-sight, row-aligned, and column-aligned axes of the image coordinate system (left) can be defined by the *N-E-D* coordinates of those unit vectors (right). These coordinates define the image system's absolute attitude (relative to the sensor-local-level coordinate system).

The values for each of the nine unit-vector coordinates (direction cosines) are unitless and must fall between ±1, inclusive. The square root of the sum of the squares of each set of three coordinates (each unit vector's magnitude) should be one. Furthermore, each unit vector should be orthogonal to the other two.[1]

Even when Module 8 is used to define the sensor's attitude, providing the equivalent Euler angles[2] in Module 7 has several benefits—see Z.5.7 Euler Angles Module.

Visualization options in the SENSRB Data Conversion Tool are also available to further aid with the interpretation of the unit vector definitions used in this module—see Z.6.4 SENSRB Data Conversion Tool.

The uncertainties for the image coordinate system unit vector coordinates (or direction cosines) should be provided in the Uncertainty Data Module (Module 14). These uncertainties are needed to support various exploitation applications. These uncertainties can be fixed values based on engineering estimates or specifications, or they may be extracted from covariance data produced by many modern inertial measurement units.

The unit vector (or direction cosines) uncertainties are stored in Module 14 using a unique method. The uncertainties are formed in a covariance matrix associated with the absolute uncertainties relative to the reference frame, rather than as the uncertainties associated with the individual coordinates (or direction cosines). This storage method is outlined in greater detail in Z.5.14.5 Attitude Unit Vector Uncertainties.

[1] The orthogonality of the unit vectors can be tested using either vector dot- or cross-products: the dot-product of any two unit vectors should be zero, and the appropriate cross-product of any two unit vectors should produce the third vector.

[2] Equations for computing equivalent Euler angles, unit vectors, and quaternions are available in Z.6.3 Attitude Representation Conversions. The attitude conversions can also be made using the SENSRB Data Conversion Tool (see Z.6.4 SENSRB Data Conversion Tool).

Z.5.9. QUATERNIONS MODULE (MODULE 09)

The sensor coordinate system's attitude (or orientation) can be provided in a SENSRB TRE by any of three equivalent[1] methods:

1. Euler Angles—Module 7[1]
2. Unit Vectors (or direction cosines)—Module 8
3. Quaternions—Module 9[2]

For a content level value (CONTENT_LEVEL, index 01f) of "4" or higher (see Z.5.1.2 Application-Required Content Level) one of the three above-listed modules (Module 7, 8, or 9) must exist and provide the necessary data to define the sensor's attitude. Furthermore, unless the sensor is stationary, the metadata values in these modules are likely to be dynamic. In other words, if multiple SENSRB TREs exist for an image segment, it would not be unexpected for the field values in these modules to be different in the various TREs—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

Quaternions are the predominate method used to describe the attitude of spacecraft and space-based sensors. Quaternions have the advantage of being compact, and they avoid the discontinuities that can occur when using Euler angles.[3]

The quaternion in Module 9 defines the attitude of the sensor coordinate system relative to the sensor's local-level coordinate system or the geocentric (earth-centered, earth-fixed) coordinate system, depending on the value of the GEODETIC_TYPE (index 01h)—see Z.5.1.3 Datum and Unit Declarations. These two reference systems are described in Z.4.6.1.4 Local-Level Coordinate System and Z.4.6.1.3 Geocentric Coordinates, respectively.

The four quaternion elements provided in the module are normalized (the magnitude of the four values should be unity). The values in the first three fields (ATTITUDE_Q1, ATTITUDE_Q2, and ATTITUDE_Q3; indices 09a through 09c) can be used to define a rotation unit vector ($\hat{\mathbf{j}}$) in the reference system (either *N-E-D* or X_E-Y_E-Z_E). The value in the last field (ATTITUDE_Q4, index 09d) can be used to define the rotation angle (θ_j) about the previously defined vector. The quaternion elements (q_1, q_2, q_3, and q_4—indices 09a through 09d) are defined in terms of $\hat{\mathbf{j}}$ and θ_j in Equation Z.5-5.

$$[q_1 \quad q_2 \quad q_3 \quad q_4] = \left[\sin\left(\frac{\theta_j}{2}\right)\cdot j_x \quad \sin\left(\frac{\theta_j}{2}\right)\cdot j_y \quad \sin\left(\frac{\theta_j}{2}\right)\cdot j_z \quad \cos\left(\frac{\theta_j}{2}\right)\right] \qquad \text{(Z.5-5)}$$

where j_x, j_y, and j_z are the X_E or *N*, Y_E or *E*, and Z_E or *D* components of $\hat{\mathbf{j}}$, respectively.

Even when Module 9 is used to define the sensor's attitude, providing the equivalent Euler angles[4] in Module 7 has several benefits—see Z.5.7 Euler Angles Module.

Visualization options in the SENSRB Data Conversion Tool are also available to further aid with the interpretation of the quaternion definitions used in this module—see Z.6.4 SENSRB Data Conversion Tool.

---

1 If field values are provided in two or more of the attitude modules and the values are in conflict (inconsistent between modules), precedence will be given to the values in the modules using the above-shown order—Module 8 taking precedence over the other two, and Module 9 taking precedence over Module 7.

2 The Euler angles and quaternions of Modules 7 and 9, respectively, define the attitude of the sensor coordinate system rather than the image coordinate system—see Z.4.6.3 Sensor Coordinate System and Z.4.6.5 Image Coordinate System.

3 These discontinuities exist, using the SENSRB definitions, when the second Euler angle becomes ±90°.

4 Equations for computing equivalent Euler angles, unit vectors, and quaternions are available in Z.6.3 Attitude Representation Conversions. The attitude conversions can also be made using the SENSRB Data Conversion Tool (see Z.6.4 SENSRB Data Conversion Tool).

The uncertainties for the quaternion elements should be provided in the Uncertainty Data Module (Module 14). These uncertainties are needed to support various exploitation applications. These uncertainties can be fixed values based on engineering estimates or specifications, or they may be extracted from variance and cross-variance data produced by many modern inertial measurement units.

Z.5.10. SENSOR VELOCITY DATA MODULE (MODULE 10)

Some sensors (such as pushbroom or whiskbroom) may be able to utilize (or may even require) sensor velocity data. This data can be stored in the Sensor Velocity Data Module (Module 10).

If sensor velocity data is needed to perform photogrammetric functions with an image segment then this module must exist and provide the necessary data to define the sensor's velocity in order to indicate a content level value (CONTENT_LEVEL, index 01f) of "4" or higher (see Z.5.1.2 Application-Required Content Level). Furthermore, it is likely that the metadata values in this module are dynamic. In other words, if multiple SENSRB TREs exist for an image segment, it would not be unexpected for the field values in this module to be different in the various TREs—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

The velocity data values in this module are defined to be the sensor's linear velocity components. As such, they (and any reported uncertainties) should also account for any sensor velocity due to the platform's angular rates. The velocity components are relative to either the sensor's local-level coordinate system (N-E-D) or the geocentric (earth-centered, earth-fixed) coordinate system (X_E-Y_E-Z_E), depending on the value of the GEODETIC_TYPE (index 01h)—see Z.5.1.3 Datum and Unit Declarations. These two reference systems are described in Z.4.6.1.4 Local-Level Coordinate System and Z.4.6.1.3 Geocentric Coordinates, respectively.

The fields (VELOCITY_NORTH_OR_X, VELOCITY_NORTH_OR_X, and VELOCITY_NORTH_OR_X; indices 10a to 10c) provide the velocity components in the $+N$ or $+X_E$, $+E$ or $+Y_E$, and $+D$ or $+Z_E$ directions; respectively. Negative component velocities are valid. The square root of the sum of the three components squared should equal the total sensor speed relative to the earth. The units for the component velocities are either meters per second (m/s) or feet per second (ft/s), depending on the value of LENGTH_UNIT (index 01j)—see Z.4.4.1 Linear Units.

The uncertainties associated with the velocity components should be provided in the Uncertainty Data Module (Module 14). These uncertainties can be fixed values based on engineering estimates or specifications, or they may be extracted from variance and cross-variance data produced by many modern navigation systems.

Z.5.11. POINT DATA SETS MODULE (MODULE 11)

The Point Data Sets Module (Module 11) allows for a variety of point data to be stored in the SENSRB TRE. Some sensing systems make ranging measurements to single or multiple locations in the scene. Others systems process data[1] to estimate the range to or the location of the scene's corner points or its center. Module 11 accommodates the storage of this kind of data and other point data sets.

It is envisioned that a structure like that of Module 11 may become its own NITF TRE. If and when this happens, the use of the new TRE would be preferred unless the point data is specifically and directly pertinent to the sensor.

[1] This processing often involves a terrain model. These models have varying degrees of fidelity—flat earth models being common. Any added uncertainties introduced by these models should be accounted for in the associated values provided in the Uncertainty Data Module (Module 14).

For an odd-numbered content level value (CONTENT_LEVEL, index 01f—see Z.5.1.2 Application-Required Content Level), Module 11 must exist and provide at least sufficient data to define the image's footprint on the earth. The metadata in this module is expected to be unchanging (static), unless adjustments are made during post processing—see Z.5.1.5 Image Parameter Post-Collection Adjustments.

Uncertainties should be provided in the Uncertainty Data Module (Module 14) for any of the point data values provided in this module—see the last paragraph of Z.5.11.2 Point Data Fields for additional guidance.

Z.5.11.1. Looping Field Structure

This module uses the looping field concept introduced in Z.4.8 Looping Field Concepts. The first field in the module (POINT_SET_DATA, index 11) is a count indicating how many sets of point data exist in the module. This value will be denoted here as *M*. Each of the *M* sets of data is then provided in turn. The first field in a set (POINT_SET_TYPE_MM, index 11a) uses a NTB-registered value to indicate what type of set the points make up—see Z.4.1 NTB-Registered Field Values. Several point set types are described in Z.5.11.3 Point Set Types below. The second field in each set (POINT_COUNT_MM, index 11b) indicates how many points are contained in that set. This value will be denoted here as *N*. Several values can now be provided for each of the *N* points, in turn.

Z.5.11.2. Point Data Fields

For each point in each point set, the following set of fields is available:

<u>Image Coordinates</u>. The values in the image coordinate fields (P_ROW_NNN and P_COLUMN_NNN, indices 11c and 11d) are related to the NITF-stored array (see Z.4.5.3 NITF-Stored Image Array). The subsequent field values pertain to the imaged feature (object) corresponding to these image array common coordinates. These two fields must have meaningful values and shall not contain the unspecified indicator. This image array common coordinate system location can be entered as a decimal number, if necessary, to represent a precise image location.

<u>Object Coordinates</u>. These three fields contain the geospatial coordinates, in geodetic coordinates (latitude, longitude, elevation), for the feature (object) located at the image coordinates given in the previous fields.

These three fields (P_LATITUDE_NNN, P_LONGITUDE_NNN, and P_ELEVA-TION_NNN, indices 11e, 11f, and 11g) are always geodetic (see Z.4.6.1.1 Geodetic Coordinates), even when the geodetic coordinate type (GEODETIC_TYPE, index 01h) has the value of “C” for geocentric Cartesian—see Z.5.1.3 Datum and Unit Declarations. The elevation will be relative to the datum specified by ELEVATION_DATUM (index 01i). As stated in Z.4.6.1.3 Elevation and Altitude Datums, heights above ellipsoid (HAE) are preferred. The elevation units will be either meters (m) or feet (ft), depending on the value of LENGTH_UNIT (index 01j)—see Z.4.4.1 Linear Units. Any of the three object coordinate fields may contain the unspecified indicator (dashes), when the values are not available.

<u>Object Range</u>. Finally, the slant range from the sensor's perspective center to the imaged feature can be provided in P_RANGE_NNN (index 11h). The range field is included in this module to accommodate sensor systems that include active ranging devices (such as laser range finders or LADAR), but range estimates can be provided from other sensor systems, when those estimates are available. The units for the range will be either meters (m) or feet (ft), depending on the value of LENGTH_UNIT (index 01j)—see Z.4.4.1 Linear Units. The object range field may be unspecified (dashes), when the value is not available.

The indexing of the point data fields for referencing purposes in the Uncertainty Data Module and elsewhere will be according to the guidance given in Z.4.8.2 Looping Field Indexing. See the following paragraph for the exception to this guidance.

The uncertainties associated with each of the above described values should be provided for each point using the points full index; however, if the same uncertainties apply to all the points in a set or to all the points in the module, the index reference can be truncated to indicate the level at which the data applies. For example, if the uncertainties associated with all the ranges in the third set of point data were identical, this uncertainty could be provided using the index “11h3” in the field UNCERTAINTY_FIRST_TYPE_NNN (index 14a). Furthermore, if all the elevations in the entire module had the same uncertainty, these could be provided using the index “11g” in the same uncertainty module field (index 14a). Section Z.5.14.5 Geodetic Coordinate Uncertainties provides additional guidance regarding the reporting of uncertainties associated with geodetic coordinates.

Z.5.11.3. Point Set Types

List of currently approved point set types.

Sensor Aimpoint: Position in the image domain and the estimated geographic position on the ground domain where the sensor’s line-of-sight vector intersects the imaged area. May not always be the same as image center for such cases as image chipping.

Image Center: This latitude, longitude, and elevation of the image center.

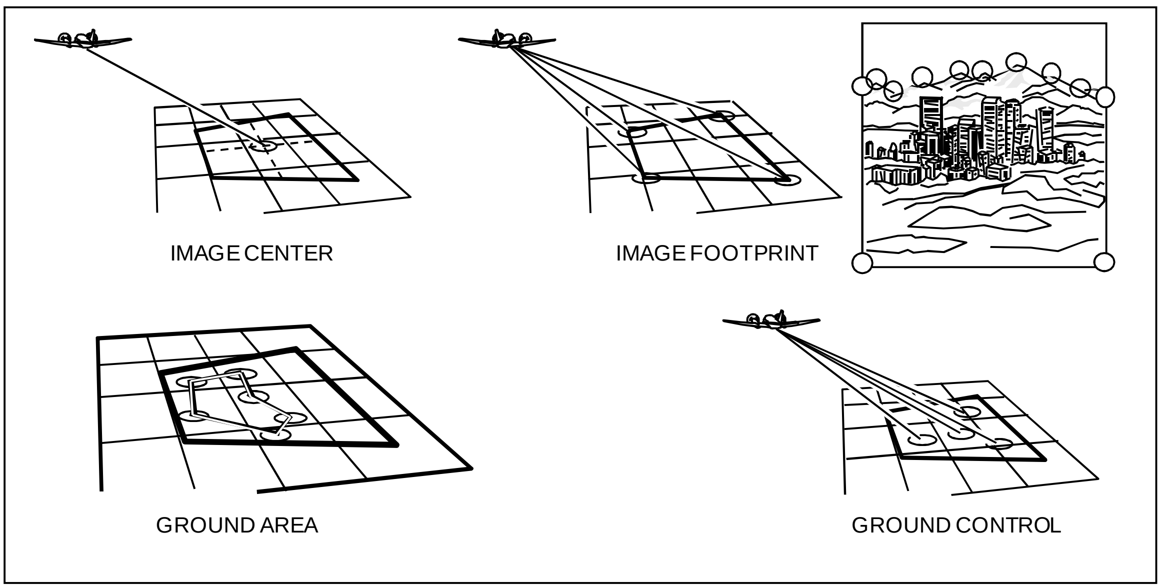


Figure Z.5.9-1. Point Set Type Examples. This graphic provides examples of four types of point sets contained within an image (displayed as a dark border): Image Center (top left), Image Footprint (top right) (The image footprint may be simple as shown with the four corner point collection or may be complicated as indicated on the more oblique image at the far right.), Ground Area (bottom left), and Ground Control (bottom right). The ground area allows any portion of the image to be called out. The ground control could be applicable to various kinds of points such as those illuminated by a laser range finder, other points of interest (such as tie points), or surveyed markers. The uncertainties associated with the points (Module 14) will indicate how they can be used.

Image Footprint: The geographic footprint of the NITF image segment. A minimum of 3 points must be defined. Only one Image footprint per NITF image segment is allowed. Generally, the footprint should follow the IGEOLO NITF Standard and therefore be defined in either a clockwise (preferred) or counterclockwise direction, starting at the upper-left (preferred) and set to the actual corner locations of the image when possible, namely (0.0, 0.0), (0.0, MaxCol), (MaxRow, MaxCol), and (MaxRow, 0.0), where MaxRow and MaxCol represent the number of rows and the number of columns.

Ground Area: A collection of points of special interest. For example, one might have tagged something post-process (not in the sensor) indicating special points on the ground such as accurate ground control coordinates or special points that need to be identified.

Ground Points: Latitude, longitude, and elevation of objects of special interest in the image

Point of Interest: Similar to an Aimpoint or Image Center, this is the coordinate (latitude, longitude, and elevation) of one or a set of reference points to be used for initialization of a CSM model.

Area of Interest: A connected set of points outlining an area of interest. These are the coordinates (latitude, longitude, and elevation) of a set of bounding points.

LRF Measurements: Measurement of range from image sensor position to illuminated point with the use of a laser range finders (LRF).

Z.5.12. TIME-STAMPED DATA MODULE (MODULE 12)

The Time Stamped Data Module (Module 12) makes use of the looping field concept introduced in Z.4.8 Looping Field Concepts. The outer loop consists of the *MM* parameters and each parameter in turn may have its own looping field for parameter values. The first field in the module (TIME_STAMPED_ DATA_SETS, index 12) is a count indicating how many sets of dynamic data exist in the module. The number of allowed dynamic parameters is between 0 and 99. The second field in the module (TIME_STAMP_ TYPE_MM, index 12a) uses is limited to reasonably applicable fields (dynamic parameters) in the value range 06a-10c. The third field in the module (TIME_STAMP_ TIME_NNNN, index 12b) is the reference time in seconds relative to START_ TIME (index 01m) associated with the subsequent value of the m^{th} parameter. The fourth field is (TIME_STAMP_ VALUE_NNNN, index 12c) is the value at that individual time.

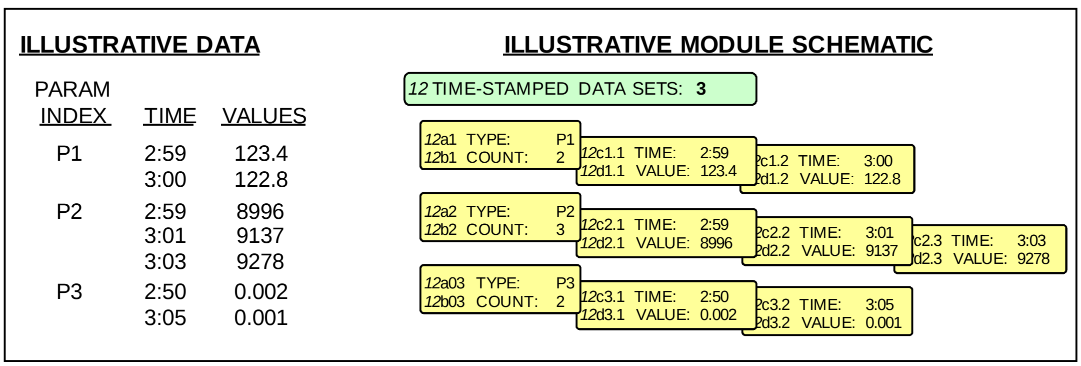


Figure Z.5.12-1. Illustrative Time-Stamped Data Example.

Z.5.13. PIXEL-REFERENCED DATA MODULE (MODULE 13)

The Pixel Referenced Data Module (Module 13) makes use of the looping field concept introduced in Z.4.8 Looping Field Concepts. The outer loop consists of the *MM* parameters and each parameter in turn may have its own looping field for parameter values. The first field in the module (PIXEL_REFERENCED_ DATA_SETS, index 13) is a count indicating how many sets of dynamic data exist in the module. The number of allowed dynamic parameters is between 0 and 99. The second field in the module (PIXEL_REFERENCED_ TYPE_MM, index 12a) uses is limited to reasonably applicable fields (dynamic parameters) in the value range 06a-10c. The third field in the module (PIXEL_REFERENCED_ ROW_NNNN, index 13b) is the reference row. The fourth field in the module (PIXEL_REFERENCED_ COLUMN_NNNN, index 13b) is the reference column and the fifth field is (PIXEL_REFERENCED_ VALUE_NNNN, index 13e) is the value at that individual pixel.

ILLUSTRATIVE DATA

PARAM INDEX	PIXEL LOCT'N	VALUES
P1	(0, 0)	123.4
	(0, 2000)	122.8
P2	(0, 0)	8996
	(0, 1000)	9137
	(0, 2000)	9278
P3	(0, 0)	0.002
	(0, 2000)	0.001

ILLUSTRATIVE MODULE SCHEMATIC

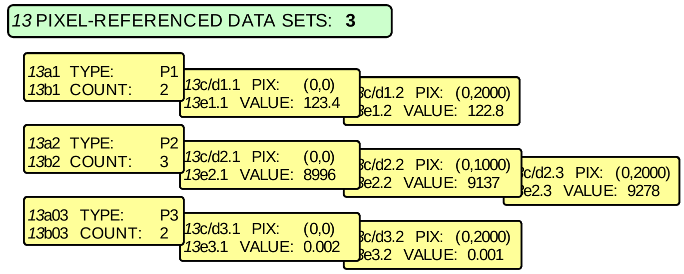


Figure Z.5.12-2. Illustrative Pixel-Referenced Data Example.

Z.5.14. UNCERTAINTY DATA MODULE (MODULE 14)

Module 14 allows for estimated or assumed uncertainties and correlation coefficients to be reported for many of the previously indexed parameters using looping fields—see Z.4.8 Looping Fields Concept. These uncertainties and correlation coefficients can be used to estimate the—sometimes application-required—final uncertainties in geocoordinates derived using those parameters. Each uncertainty is reported as a standard deviation value (1σ) and will always be positive. The correlation between the uncertainties in any two parameters will be expressed as a correlation coefficient and will always be a value between ±1, inclusive. The integer-value flag indicates the number of parameters or parameter pairs for which the uncertainty values and correlation coefficients are provided.

The uncertainties for the parameters that define the sensor location and attitude are expected for any application providing accurate geopositioning of imaged objects. The correlations between these parameters are also expected for such applications. These standard deviations and correlation coefficients can be computed from values typically available in the form of a covariance matrix from a navigational unit associated with the platform and/or sensor (such as an inertial measurement unit or an integrated GPS and inertial navigation system). The methodology for retrieving the standard deviation and correlation coefficient from the covariance data is given in Equations Z.5-6 for a generic two-dimensional case. This methodology can easily be extended to (or beyond) the expected six degrees of freedom (3 position and 3 attitude parameters).

$$\sigma_x = \sqrt{\sigma_x^2}\,, \quad \sigma_y = \sqrt{\sigma_y^2}$$
$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y} = \rho_{yx} = \frac{\sigma_{yx}}{\sigma_x \cdot \sigma_y} \qquad \text{(Z.5-6)}$$

where the positive values of the square root is assumed; ρ_{xy} is the correlation coefficient between x and y (which is the same as ρ_{yx}); and σ_x^2, σ_y^2, σ_{xy}, and σ_{yx} are the elements of the two-dimensional covariance matrix ($\sigma_{xy} \equiv \sigma_{yx}$):

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}.$$

Figure Z.5-16 gives an example of how a two-dimensional covariance matrix would be converted into a correlation matrix and then stored (schematically) in Module 14.

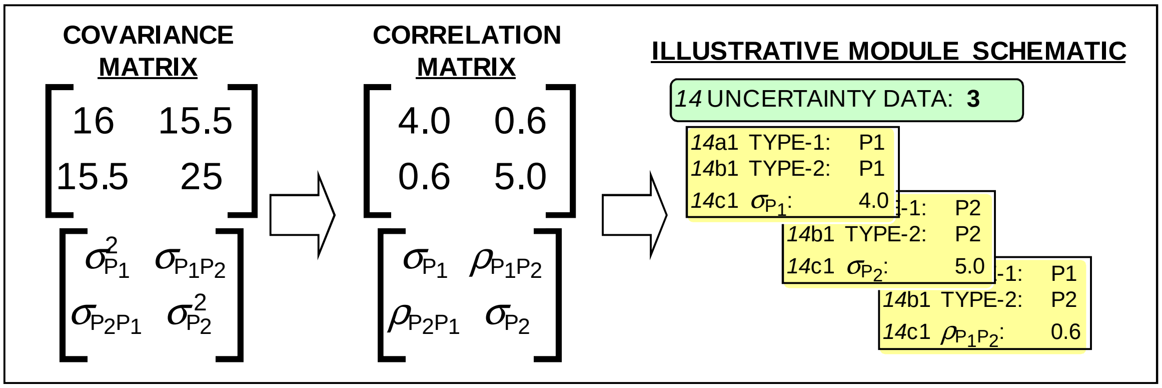


Figure Z.5.14-1. Illustrative Uncertainty Data Example.

Z.5.14.1. Uncertainty Values

For each reported uncertainty value, the associated parameter index (see Z.2.2.2 Field References and Z.4.8.2 Looping Field Indexing) is given as the UNCERTAINTY_FIRST_TYPE_NNN (index 14a). The index will then either be repeated in UNCERTAINTY_SECOND_TYPE_NNN (index 14b) or that field (index 14b) will contain the unspecified indicator. The uncertainty estimate for the indexed parameter is then given in UNCERTAINTY_VALUE_NNN (index 14c).

The uncertainties (with only the exceptions explained in Z.5.14.3 Geodetic Coordinate Uncertainties and Z.5.14.4 Attitude Unit Vector Uncertainties) are always in the same units as are used for the parameter itself. The fields for uncertainty values are a fixed length. The uncertainties are to be reported using exponential notation (see Z.4.3.2 Number Representations). In the absurd case that an uncertainty is larger than 9.99999e99 or smaller than 1.0000e-99 (1σ), the appropriate extreme value (9.99999e99 or 0.00000e00) should be substituted for the estimated uncertainty value. When no uncertainty estimate or assumption has been made for a parameter, that parameter's index should not appear in Module 14.

Z.5.14.2. Correlation Coefficients

Unless reported otherwise, the uncertainties in any two parameters are assumed to be independent and have a zero-valued correlation coefficient. For each reported correlation coefficient, the two associated parameter indices (see Z.2.2.2 Field References and Z.4.8.2 Looping Field Indexing) are given respectively in the UNCERTAINTY_FIRST_TYPE_NNN (index 14a) and the UNCERTAINTY_SECOND_TYPE_NNN (index 14b). The correlation coefficient is then given in UNCERTAINTY_VALUE_NNN (index 14c). The correlations are based on the uncertainties as described in this section (Z.5.14 Uncertainty Data Module). The fields for the coefficient values are as described above for the standard deviations, but the coefficient must have a value between -1.0000e00 and 1.00000e00, inclusive. When no correlation has been estimated or a zero correlation has been assumed between the uncertainties associated with a pair of parameters, that pair of parameters' indices should not appear together in module 14, such that the default assumption of independence (zero correlation) is made.

Z.5.14.3. Dynamic Uncertainties

If the uncertainty or correlation coefficient for a parameter or parameter pair were to change appreciably during a collection, a new instantiation of SENSRB (a new TRE) would be required to characterize that change, as the estimated values reported in the module are assumed to be constant. Reported uncertainty and correlation values are to be assumed valid from the reference time (or location) they are reported until they are updated—see Z.4.9.3 Multiple SENSRB TREs and Module Existence.

Z.5.14.4. Geodetic Coordinate Uncertainties

Uncertainties associated with geodetic coordinates (latitudes and longitudes) shall be reported in either feet or meters (depending on the setting of LENGTH_UNIT, index 01j) in the coordinate's local-level coordinate system—north-south (for latitude) or east-west (for longitude) directions.

Z.5.14.5. Attitude Unit Vector Uncertainties

The treatment in this section follows closely to “Strapdown Analytics: Part1” by Paul G. Savage which is a great reference on unit vectors and their uncertainties.

C_B^A Is the direction cosine matrix that transforms vectors from B_{frame} to A_{frame}

$\widetilde{C}_B^A$ is C_B^A as calculated by the system

$\partial \widetilde{C}_B^A = \widetilde{C}_B^A - C_B^A$

Where

$\partial \widetilde{C}_B^A = Error\ in\ \widetilde{C}_B^A$

Once the error in directional cosines is determined it is easy to find the error in Euler angles and quaternions.

Z.5.15. ADDITIONAL PARAMETER DATA MODULE (MODULE 15)

The Additional Parameter Data Module (Module 15) is designed to make the SENSRB TRE expandable to support evolving requirements. The module permits a user to specify data values or data arrays for a new parameter. It is envisioned that these additional parameters will be registered (see Z.4.1 NTB-Registered Field Values) although in some instances users may wish to utilize a new parameter for special internal studies to establish the utility of that additional parameter.

This module uses the double looping field concept introduced in Z.4.8 Looping Field Concepts. The outer loop consists of the set of *MMM* parameters, where *MMM* is a number of up to 3 digits. Each parameter will have:

- a “name” (a left justified alphanumeric with blank fill to the right),
- a “size” (the number of characters allocated to each of the values),
- a parameter “count” indicating the number of values for that parameter, and
- the set of “count” (*NNNN*) parameter values.

When used with the SENSRB Data Conversion Tool, any parameter name will be accepted and the parameter values are treated as floating point values. As a result, these additional parameters may also be referenced in Module 14 to provide uncertainties with respect to these additional parameters.

The intent of Module 15 is to provide for expansion. Users are encouraged not to use this module as a “catch-all” or to register names having a very narrow application.

Z.5.15.1 Currently Approved Additional Data Parameters

The currently defined (NTB approved) parameter names are listed below. Refer to the NTB website for an up-to-date list and for full description of the parameters.

- ASSOCIATED_DATA_MODULE
- DATE_STAMP
- DATE_TIME_STAMP
- LUNAR_AZIMUTH
- LUNAR_ELEVATION
- LUNAR_GLINT_LATITUDE
- LUNAR_GLINT_LONGITUDE
- LUNAR_PHASE
- MI_LDS_VERSION
- MI_SECURITY_VERSION
- MI_TARGET_WIDTH
- SENSOR_AZIMUTH
- SENSOR_ELEVATION
- SOLAR_AZIMUTH
- SOLAR_ELEVATION
- SOLAR_GLINT_LATITUDE
- SOLAR_GLINT_LONGITUDE
- TIME_STAMP

Z.6. ADVANCED TECHNICAL CONCEPTS

Z.6.1. SENSRA TO SENSRB MAPPING

SENSRB was created to address missing fields, dynamic parameters, and to better describe sensor orientation not found in SENSRA. It is possible to go from SENSRA to SENSRB, but depending on the content level the converse might not be possible due to the additional data contained in SENSRB.

Z.6.1.1. SENSRA TRE TO SENSRB TRE CONVERSION

Table Z.6-2 indicates how SENSRA metadata elements relate to SENSRB. Some elements are directly related, some require conversions or conditions to consider, and some elements are not supported by SENSRB. Although not recommended, in some instances a metadata element that is not supported may be registered with the NTB as an additional parameter in module 15 of SENSRB.

NOTE: This table provides only high level guidance to indicate where relationships do or do not exist between the metadata elements. It is the responsibility of the SENSRB implementer to give due diligence to the SENSRB specifications (Table Z.3-1 and Table Z.3-2) and to use Table Z.6.1 for informative guidance only.

TABLE Z.6-1. SENSRA TRE TO SENSRB TRE MAPPING

= DIRECT RELATION, ≈ PARTIAL RELATION, ≠ NO RELATION, - NO COMMENT

SENSRA FIELD	MAP	INDEX	SENSRB FIELD	RELATIONSHIP
REF_ROW	=	05a	REFERENCE_ROW	-
REF_COL	=	05b	REFERENCE_COLUMN	-
SENSOR_MODEL	≈	01a	SENSOR	- Use NTB approved name
SENSOR_MOUNT	≠	-	N/A	-
SENSOR_LOC	=	06a 06b	LATITUDE_OR_X LONGITUDE_OR_Y	SENSRA SENSOR_LOC is recorded in geodetic coordinates; SENSRB allows for geodetic or geocentric. To maintain SENSRA convention, SENSRB GEODETIC_TYPE field value must be “G”.
SENSOR_ALT_ SOURCE	=	01g	ELEVATION_DATUM	To convert SENSRA to SENSRB values, B = MSL, G = HAE, R = AGL. M is not applicable.
SENSOR_ALT	=	06c	ALTITUDE_OR_Z	Recommend the use of “HAE”
SENSOR_ALT_UNIT	=	01h	LENGTH_UNIT	Indicate “SI” or “EE” in SENSRB Module 01 for meters or feet, respectively. All of the SENSRA “LENGTH_UNIT” values must be the same in SENSRB.
SENSOR_AGL	≈	06c 01g	ALTITUDE_OR_Z ELEVATION_DATUM	
SENSOR_PITCH	≠	-	-	SENSRA definitions are not related to SENSRB definitions. See next section. Refer to SENSRB Module 7 (ATTITUDE EULER ANGLES) or preferably Module 8 (ATTITUDE UNIT VECTORS) for definitions and implementations.
SENSOR_ROLL	≠	-	-	
SENSOR_YAW	≠	-	-	
PLATFORM_PITCH	=	07b	PLATFORM_PITCH	-
PLATFORM_ROLL	=	07c	PLATFORM_ROLL	
PLATFORM_HDG	=	07a	PLATFORM_HEADING	
GROUND_SPD _ SOURCE	≠	-	-	-
GROUND_SPD	≈	09a 09b	VELOCITY_NORTH VELOCITY_EAST	SENSRA GROUND_SPD must be reported as the North and East velocity components calculated using PLATFORM_HDG field value.
GROUND_SPD_UNIT	≈	01h	LENGTH_UNIT	SENSRB does not support “k” for “knots” as a unit and must be converted to feet or meters per second.
GROUND_TRACK	≠	-	-	-
VERT_VEL	≈	09c	VELOCITY_DOWN	SENSRA has positive velocity upward. SENSRB has positive velocity downward, which must be accounted for. Conversion from per minute to per second is also required.
VERT_VEL_UNIT	=	01h	LENGTH_UNIT	

SENSRA FIELD	MAP	INDEX	SENSRB FIELD	RELATIONSHIP
SWATH_FRAMES	≠	-	-	Depending on the method of image construction, these parameters may map to fields in SENSRB Module 04, Image Formation.
N_SWATHS	≠	-	-	
SPOT_NUM	≠	-	-	

Z.6.1.2. SENSRA Angles to Euler angles Transformation

Converting the platform-relative SENSRA angles (the angles for the projections—ψ_A, θ_A, and ϕ_A) into the more commonly used platform-relative azimuth and depression Euler angles (ψ_S and ϕ_S):

By inspection, the platform-axis components for the line-of-sight vector are:

$z_P = \sin\theta_S$
$x_P = \cos\theta_S \ \cos\psi_S$
$y_P = \cos\theta_S \ \sin\psi_S$

The tangents of the SENSRA angles are defined by the following (see the three far-right diagrams):

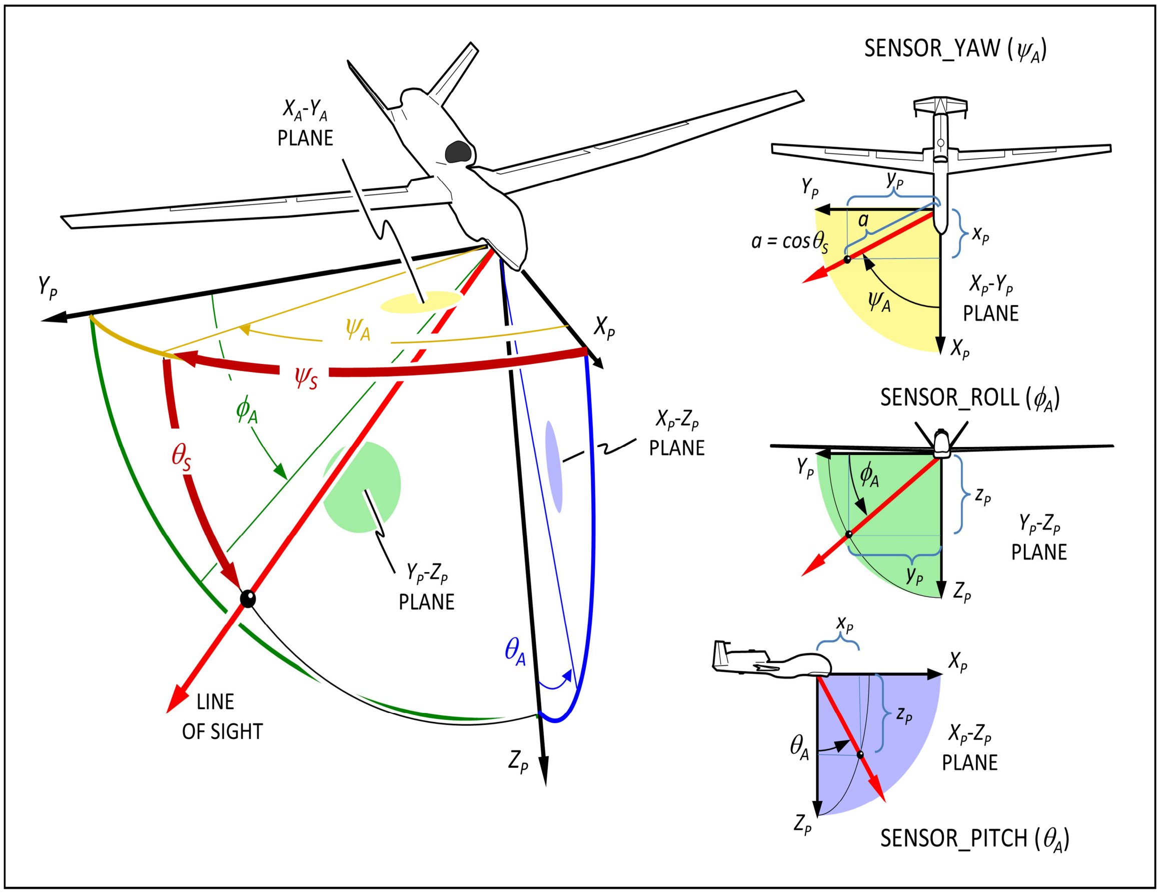


Figure Z.6-1: SENSRA (not SENSRB) angles definitions

$\tan\psi_A = \ y_P \,/\, x_P = (\cos\theta_S \sin\psi_S) \,/\, (\cos\theta_S \cos\psi_S) = \sin\psi_S \,/\, \cos\psi_S = \tan\psi_S$
$\tan\phi_A = \ z_P \,/\, y_P = (\sin\theta_S) \,/\, (\cos\theta_S \sin\psi_S) = \tan\theta_S \,/\, \sin\psi_S$
$\tan\theta_A = \ x_P \,/\, z_P = (\cos\theta_S \cos\psi_S) \,/\, (\sin\theta_S) = \cos\psi_S \,/\, \tan\theta_S$

Therefore …

$\psi_S = \psi_A$, and $\theta_S = \arctan(\sin\psi_A \tan\phi_A) = \arctan(\cos\psi_A \,/\, \tan\theta_A)$ (Eqns 1)

Note: the sensor roll angle—ϕ_S—for a two-gimbal system will always be 0°.

The following outlines the process for converting SENSRA sensor and platform angles into NED-relative Euler angles:

(1) Convert SENSRA angles to platform-relative Euler angles using Eqns 1.

(2) Build sensor rotation matrix using sensor Euler angles

(3) Build platform rotation matrix using platform (NED-relative) Euler angles from SENSRA

(4) Multiple two rotation angles to get a composite rotation matrix (test to check multiply order)

(5) Determine composite Euler angles by solving components of composite rotation matrix

Z.6.1.3. Euler angles to SENSRA angles Transformation

Given the platform-relative azimuth and depression Euler angles (ψ_S and ϕ_S) one would have to convert to the platform-relative SENSRA angles (the angles for the projections—ψ_A, θ_A, and ϕ_A). One could use the following methodology:

By inspection, the platform-axis components for the line-of-sight vector are:

$z_P = \sin\theta_S$
$x_P = \cos\theta_S \cos\psi_S$
$y_P = \cos\theta_S \sin\psi_S$

The tangents of the SENSRA angles are as shown previously in Figure Z.6-1 (see the three far-right diagrams):

$\tan\psi_A = y_P / x_P = (\cos\theta_S \sin\psi_S) / (\cos\theta_S \cos\psi_S) = \sin\psi_S / \cos\psi_S = \tan\psi_S$
$\tan\phi_A = z_P / y_P = (\sin\theta_S) / (\cos\theta_S \sin\psi_S) = \tan\theta_S / \sin\psi_S$
$\tan\theta_A = x_P / z_P = (\cos\theta_S \cos\psi_S) / (\sin\theta_S) = \cos\psi_S / \tan\theta_S$

Therefore…

$\psi_A = \psi_S$
$\theta_A = \arctan((\cos\theta_S \cos\psi_S) / (\sin\theta_S))$
$\phi_A = \arctan((\sin\theta_S) / (\cos\theta_S \sin\psi_S))$

Note, the traditional sensor roll angle—ϕ_S—for a two-gimbal system should always be 0°.)

Z.6.2 COORDINATE SYSTEM TRANSFORMATIONS

Central to various geopositioning-related applications are the transformations between the coordinate systems defined above. These systems with the respective transformations are summarized notionally in Figure Z.6-2. Illustrative equations for these transformations are provided in the following paragraphs.

A. Geodetic-to-Geocentric Transformations. Assuming the elevation or altitude term (h) is given as or is translated to a height above ellipsoid, geocentric coordinates (x_E, y_E, and z_E) can be derived from geodetic coordinates (ϕ, λ, and h) using Equation Z.6-5.

$$\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} (N+h)\cos\phi\cos\lambda \\ (N+h)\cos\phi\sin\lambda \\ \left(N\left(1-e^2\right)+h\right)\sin\phi \end{bmatrix} \qquad \text{(Z.6-5)}$$

where $N = \dfrac{a}{\sqrt{1-e^2\sin^2\phi}}$ is the prime-vertical radius of curvature, with a and e being the semi major axis length and the first eccentricity, respectively, of the ellipsoid.

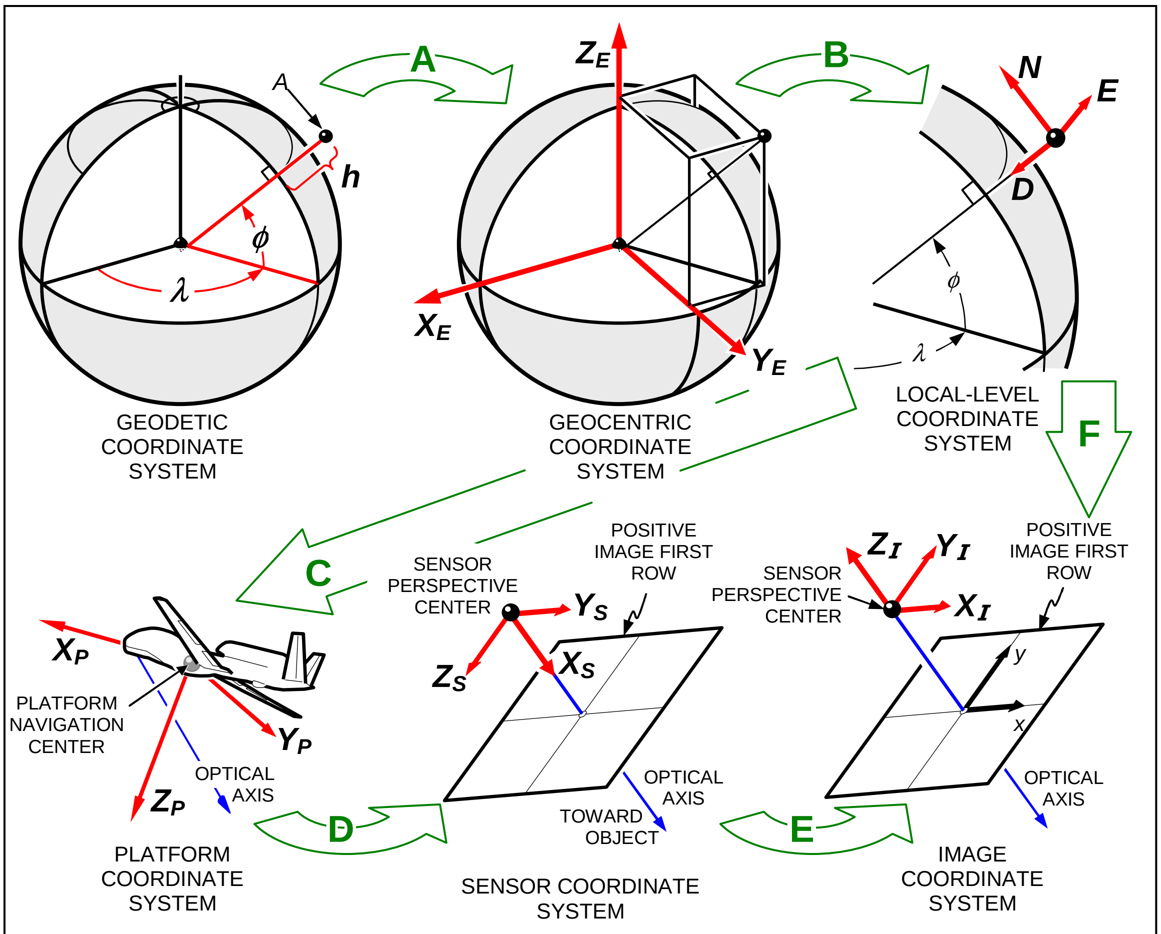


Figure Z.6-2 Coordinate System Transformations Summary. The location of point A (often a platform navigation center, but ideally a sensor perspective center) can be defined in geodetic coordinates (λ, ϕ, and h). These are transformed (represented by the arrow labeled **A**) to geocentric coordinates (X_E, Y_E, and Z_E). Other points can be located relative to A in a local-level (*NED*) coordinate system. The geocentric system can be transformed into the NED system with a translation and a rotation (represented by the arrow labeled **B**). The local-level system can then be rotated (arrow **C**) to obtain the platform coordinate system (X_P, Y_P, and Z_P). The transformation (arrow **D**) to the conceptual sensor system (X_S, Y_S, and Z_S) may require a translation (shifting the origin from the platform navigation center to the sensor perspective center) and a rotation. The final transformation (arrow **E**) to the image coordinate system* (X_I, Y_I, and Z_I) can be made with simple 90° rotations. If point *A* is co-located with the sensor perspective center a transformation from the local-level to the image coordinate system can be made directly (arrow **F**).

B. Geocentric-to-Local-Level Transformations. Geocentric coordinates for a location (x_E, y_E, and z_E) can be transformed into coordinates relative to a local-level (*NED*) system with origin at point *A*—with geocentric coordinates x_{E_A}, y_{E_A}, and z_{E_A}—using Equation Z.6-6.

$$\begin{bmatrix} n \\ e \\ d \end{bmatrix}_A = \mathbf{M}_E^{L_A}\left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - \boldsymbol{A}_E\right) = \left(\begin{bmatrix} -\sin\phi_A & 0 & \cos\phi_A \\ 0 & 1 & 0 \\ -\cos\phi_A & 0 & -\sin\phi_A \end{bmatrix} \cdot \begin{bmatrix} \cos\lambda_A & \sin\lambda_A & 0 \\ -\sin\lambda_A & \cos\lambda_A & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \cdot \left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - \begin{bmatrix} x_{E_A} \\ y_{E_A} \\ z_{E_A} \end{bmatrix}\right) \quad \text{(Z.6-6)}$$

where $\mathbf{M}_E^{L_A}$ is the rotation matrix* that rotates from the geocentric to the *A*-local, north-east-down coordinate system, $\boldsymbol{A_E}$ is the vector coordinates of the point *A* in geocentric coordinates, and ϕ_A and λ_A are point *A*'s geodetic latitude and longitude, respectively.

C. Local-Level-to-Platform Transformations. If the origin of the local-level and the platform system are the same, the transformation from the former (n, e, and d) to the latter (x_P, y_P, and z_P—relative to point *A*) can be accomplished by the three rotations defined in Equation Z.6-7.

$$\begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}_A = \mathbf{M}_{L_A}^{P}\begin{bmatrix} n \\ e \\ d \end{bmatrix}_A = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_P & \sin\phi_P \\ 0 & -\sin\phi_P & \cos\phi_P \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_P & 0 & -\sin\theta_P \\ 0 & 1 & 0 \\ \sin\theta_P & 0 & \cos\theta_P \end{bmatrix} \cdot \begin{bmatrix} \cos\psi_P & \sin\psi_P & 0 \\ -\sin\psi_P & \cos\psi_P & 0 \\ 0 & 0 & 1 \end{bmatrix}\right) \cdot \begin{bmatrix} n \\ e \\ d \end{bmatrix}_A \quad \text{(Z.6-7)}$$

where $\mathbf{M}_{L_A}^{P}$ is the rotation matrix that rotates from the *A*-local-level to the platform coordinate system.

If the platform Euler angles is not available, then the information shall be available to transform from the local-level coordinate system directly to either the conceptual-sensor or image coordinate system as described as part of the following paragraphs.

D. Platform-to-Conceptual Sensor Transformations. The transformation from the platform coordinate system (x_P, y_P, and z_P) to the conceptual sensor system (x_S, y_S, and z_S) typically involves the translation from the navigation center to the sensor's perspective center and the conceptual sensor rotations. These latter rotations will vary depending on the order of sensor rotation angles, as described by Equation Z.6-8 (again assuming that *A* is the location of the platform navigation center).

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \mathbf{M}_P^S\left(\begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix}_A - \begin{bmatrix} \Delta_X \\ \Delta_Y \\ \Delta_Z \end{bmatrix}\right) \quad \text{(Z.6-8)}$$

where $\mathbf{M}_P^S$, the rotation matrix that rotates from the platform to the sensor coordinate system, can be defined in either of the following ways, depending on SENSOR_ROTATION_ORDER (index 07d):

When order of sensor rotation angles = 1,

$$\mathbf{M}_P^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_S & \sin\phi_S \\ 0 & -\sin\phi_S & \cos\phi_S \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \cdot \begin{bmatrix} \cos\psi_S & \sin\psi_S & 0 \\ -\sin\psi_S & \cos\psi_S & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

When order of sensor rotation angles = 2,

* The transformation from row and column image-coordinate observations to the image coordinate system is described in Z.5.3 Sensor Calibration (see Equation Z.6-12). That section also provides the equations needed to derive corrected image coordinates using sensor calibration parameters.

* The rotation matrix notation used throughout this section uses the letter **M**, with a subscript indicating the original coordinate system and a superscript indicating the resulting coordinate system.

$$\mathbf{M}_P^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_S & \sin\psi_S \\ 0 & -\sin\psi_S & \cos\psi_S \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \cdot \begin{bmatrix} \cos\phi_S & -\sin\phi_S & 0 \\ \sin\phi_S & \cos\phi_S & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

where, using the auxiliary axes (*P*, *Q*, and *R*) of Z.5.7.3.1 Sensor Coordinate System Platform-Relative Attitude, the right-most matrix accomplishes the initial rotation about the *Q* axis to point the +*P* axis in the "belly-looking" direction. The second right-most matrix realizes a positive rotation about the -*R* matrix or a negative rotation about the +*R* matrix.

When the platform Euler angles is unspecified (all hyphens), it is assumed that the sensor Euler angles is relative to the local-level coordinate system. In this case, Equations Z.6-7 and Z.6-8 can be replaced by Equation Z.6-9*.

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \mathbf{M}_{L_S}^S \begin{bmatrix} n \\ e \\ d \end{bmatrix}_s \qquad \text{(Z.6-9)}$$

where $\mathbf{M}_{L_S}^S$, the rotation matrix that rotates from the sensor-local-level to the conceptual-sensor coordinate system, is defined as was $\mathbf{M}_P^S$ for Equation Z.6-8—depending on the order of the sensor rotation angles:

When order of sensor rotation angles = 1,

$$\mathbf{M}_{L_S}^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_S & \sin\phi_S \\ 0 & -\sin\phi_S & \cos\phi_S \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \cdot \begin{bmatrix} \cos\psi_S & \sin\psi_S & 0 \\ -\sin\psi_S & \cos\psi_S & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

When order of sensor rotation angles = 2,

$$\mathbf{M}_{L_S}^S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi_S & \sin\psi_S \\ 0 & -\sin\psi_S & \cos\psi_S \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_S & 0 & -\sin\theta_S \\ 0 & 1 & 0 \\ \sin\theta_S & 0 & \cos\theta_S \end{bmatrix} \cdot \begin{bmatrix} \cos\phi_S & -\sin\phi_S & 0 \\ \sin\phi_S & \cos\phi_S & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

E. Conceptual Sensor-to-Image Transformations. These last two systems have the same origin and are parallel. The transformation can be achieved by a simple reassignment of coordinates ($X_I = Y_S$, $Y_I = -Z_S$, and $Z_I = -X_S$). These reassignments might also be modeled by various combinations of two 90° rotations, one such combination being given in Equations Z.6-10.

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{M}_S^I \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{bmatrix} \cdot \begin{bmatrix} \cos 90^\circ & \sin 90^\circ & 0 \\ -\sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} \qquad \text{(Z.6-10)}$$

F. Local Level-to-Image Transformations. If the guidance in Z.7 Recommended Uniform Implementation is followed and the preferred methods of defining the sensor location and attitude are used, a transformation can be made directly from the local-level to the image coordinate system—eliminating the use of the intermediary platform and sensor coordinate systems. This direct transformation is given here as Equation Z.6-11, where point *A* is understood to be at the location of the sensor perspective center.

* In the use of Equation Z.6-9, it is assumed that the local-level coordinates have their origin at the sensor perspective center, meaning that the sensor position offsets (indices 06d to 06f) are zero. If this is not the case, the platform attitude can be used with the reported offsets to locate the sensor's perspective center in the platform-local-level system. However, if the platform attitude is not known, the vector magnitude of the offsets must be accounted for as contributing to any estimate of the sensor's location uncertainty.

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{M}_{L_A}^{I} \cdot \begin{bmatrix} n \\ e \\ d \end{bmatrix}_A \qquad \text{(Z.6-11)}$$

where $\mathbf{M}_{L_A}^{I}$, the rotation matrix that rotates from the *A*-local-level to the image coordinate system, is defined directly by the reported elements in module 8, as shown in Equation Z.6-12.

$$\mathbf{M}_{L_A}^{I} = \begin{bmatrix} \text{X\_NORTH} & \text{X\_EAST} & \text{X\_DOWN} \\ \text{Y\_NORTH} & \text{Y\_EAST} & \text{Y\_DOWN} \\ \text{Z\_NORTH} & \text{Z\_EAST} & \text{Z\_DOWN} \end{bmatrix} \qquad \text{(Z.6-12)}$$

In summary, the transformation from the geocentric to image coordinates through the platform coordinate system can be given by Equation Z.6-13.

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{M}_S^I \cdot \mathbf{M}_P^S \cdot \left(\mathbf{M}_{L_A}^{P} \cdot \mathbf{M}_E^{L_A} \cdot \left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - A_E \right) - \begin{bmatrix} \Delta_X \\ \Delta_Y \\ \Delta_Z \end{bmatrix} \right) \qquad \text{(Z.6-13)}$$

The inverse of Equation Z.6-10 is given here as Equation Z.6-14.

$$\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \mathbf{M}_E^{L_A\,\mathsf{T}} \cdot \mathbf{M}_{L_A}^{P\,\mathsf{T}} \left(\mathbf{M}_P^{S\,\mathsf{T}} \cdot \mathbf{M}_S^{I\,\mathsf{T}} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + \begin{bmatrix} \Delta_X \\ \Delta_Y \\ \Delta_Z \end{bmatrix} \right) + A_E = \mathbf{M}_{L_A}^{E} \cdot \mathbf{M}_P^{L_A} \left(\mathbf{M}_S^{P} \cdot \mathbf{M}_I^{S} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + \begin{bmatrix} \Delta_X \\ \Delta_Y \\ \Delta_Z \end{bmatrix} \right) + A_E \qquad \text{(Z.6-14)}$$

where the inverses of each of the rotation matrices can be obtained by simply transposing the matrix ($\left(\mathbf{M}_U^V\right)^{-1} = \mathbf{M}_U^{V\,\mathsf{T}} = \mathbf{M}_V^U$).

If the platform Euler angles are unspecified, Equations Z.6-13 and Z.6-14 could be replaced with Equations Z.6-15 and Z.6-16, respectively, and the footnote associated with Equation Z.6-9 would apply.

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{M}_S^I \cdot \mathbf{M}_{L_A}^{S} \cdot \mathbf{M}_E^{L_A} \cdot \left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - A_E \right) \qquad \text{(Z.6-15)}$$

$$\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \mathbf{M}_E^{L_A\,\mathsf{T}} \cdot \mathbf{M}_{L_A}^{S\,\mathsf{T}} \cdot \mathbf{M}_S^{I\,\mathsf{T}} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + A_E = \mathbf{M}_{L_A}^{E} \cdot \mathbf{M}_S^{L_A} \cdot \mathbf{M}_I^{S} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + A_E \qquad \text{(Z.6-16)}$$

If the guidance of Z.7 is followed such that point *A* is coincident with the sensor perspective center and the unit vector coordinates of module 8 are reported, Equations Z.6-13 and Z.6-14 could be replaced with Equations Z.6-17 and Z.6-18, respectively.

$$\begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{M}_{L_A}^{I} \cdot \mathbf{M}_E^{L_A} \cdot \left(\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} - A_E \right) \qquad \text{(Z.6-17)}$$

$$\begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \mathbf{M}_E^{L_A\,\mathsf{T}} \cdot \mathbf{M}_{L_A}^{I\,\mathsf{T}} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + A_E = \mathbf{M}_{L_A}^{E} \cdot \mathbf{M}_I^{L_A} \cdot \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} + A_E \qquad \text{(Z.6-18)}$$

where $\mathbf{M}_{L_A}^{I}$ is as defined by Equation Z.6-12.

Guidance on how to propagate uncertainties into the coordinate systems of Equations Z.6-13 through 6.5-18 is provided in Appendix A of the [Frame Formulation Paper].

Z.6.3. ATTITUDE REPRESENTATION CONVERSIONS

Z.6.3.1. System Overview

SENSRB provides the user with the ability to define the sensor (camera) orientation using one of three separate means: Euler Angles, Quaternion's, or Direction Cosine Matrices. The user may also specify if he/she is defining orientation information with respect to the Geodetic (North, East, Down) coordinate frame or the Earth Centered-Earth Fixed ECEF (X,Y,Z) coordinate frame. Note that the Euler Angle definition can only be employed when the user is defining orientation in the Geodetic (N,E,D) coordinate system.

When the Geodetic coordinate frame is employed, the Euler angles and quaternions equivalently define the rotation from a set of N,E,D coordinates which would then align with the camera frame. In operation, the SENSRB Tool (Z.6.4) then internally generates a Direction Cosine Matrix that defines the orientation of the photographic image frame with respect to NED coordinates.

If the user chooses to directly input a Direction Cosine Matrix (DCM) he must recognize that this input must NOT define the orientation of the camera but rather the orientation of the Photographic Image frame.

When using ECEF (or "Geocentric coordinates") note that the quaternion vector represents the required angular rotation required to bring the XYZ ECEF coordinates into alignment with the camera frame coordinates. Again the SENSRB program modifies this rotation in producing a DCM as this DCM additionally accounts for the alignment of the photographic image frame with respect to the camera frame.

Lastly note that whatever orientation approach and coordinate frame is initially selected, the SENSRB Tool will compute the equivalent representations in whatever form and coordinate frame the user desires. For example, if the initial orientation definition is specified with respect to NED coordinates as Euler angles, the SENSRB Tool can use that information to determine the equivalent Quaternion and direction cosine matrix representations in both the NED ECEF coordinates as specified by the user.

Z.6.3.2. Six Coordinate Systems

SENSRB implicitly utilizes six distinct coordinate systems in the specification of the attitude of the image plane. These are:

- ECEF or Global Cartesian or “C”
- NED or Local Horizontal or Geodetic or “G”
- Platform Relative
- The Camera or Pre-Sensor Coordinate System
- Sensor Coordinate System
- Image Coordinate System

We can represent these coordinate systems as a set of six “levels” as in the following diagram, Z.6-3.

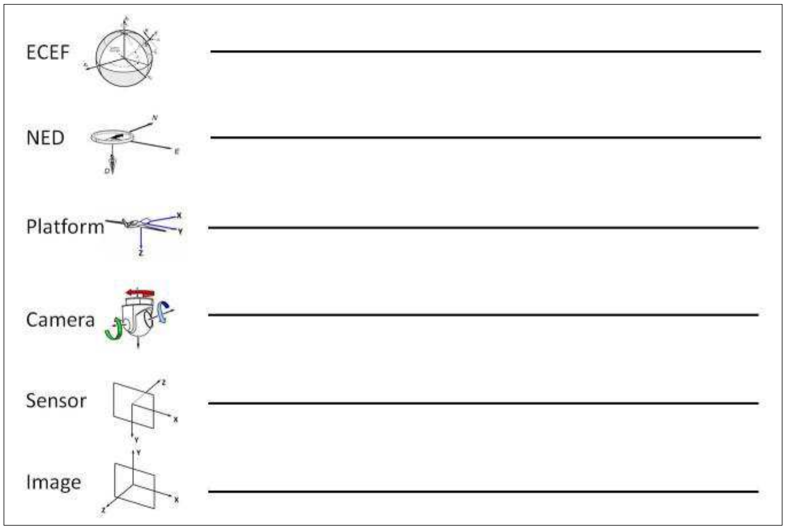


Figure Z.6-3: The Various Coordinate Systems

The ECEF is a coordinate system that is Earth Centered and Earth Fixed. As the name suggests, it is located at the center of the earth and is fixed within it, that is, it rotates along with the earth about its Z axis. It is also referred to as the Global Cartesian system or geodetic type "C".

The NED or North-East-Down system is related to the ECEF system, that is relative to the earth, but in this case it is generally relative to the surface of the earth and has its three axes aligned with the local North, East, and Down to the local gravity direction (which is close but not precisely toward the center of the earth).

The Platform system is the attitude of the vehicle with respect to the NED Coordinate system. This is defined as a Heading, Pitch, and Roll of the platform relative to NED and can *only* be defined using the optional input in the Attitude Euler Angle module. When there is no rotation of the platform relative to the NED system, the platform X axis is in the direction of flight (North), the right wing, the Y axis) points to the East, and the Z axis points Down. The platform system is not used for a satellite, which is typically measured with respect to the ECEF system.

The Camera system is the coordinate frame that results from the rotation of the physical camera system through a set of three gimbals applied in a defined order. Initially the Camera system is aligned with the platform; it is where the camera is "looking" relative to the (0,0,0) position of the camera system with respect to the platform.

The Sensor system is a coordinate frame aligned with the Camera system except that there is an inherent coordinate system transformation so that the sensor inside the camera has its X axis aligned with the rows of the captured image, the Y axis is aligned with the increasing columns (downward) on the image, and the Z axis points forward, that is in the direction that camera is "looking" (the previous X axis of the Camera).

Finally the Image system is very similar to the sensor system except that there is a permutation of the coordinate directions so that the X axis remains the same, the Y axis points upward relative to the rows of the image and the Z axis points out of the image toward the observer of the image.

The point of all of this is to unambiguously define the orientation of the photograph, the image, with respect to some baseline system, either the ECEF, the NED, or the Platform system.

Z.6.3.3. The Euler Angle Transformation

The Euler angles are the most intuitive of the Attitude orientation specifications but even there it gets complicated.

If the Platform Relative flag in Module 07 is set, then the platform orientation is defined with respect to the NED system through an associated Platform Heading, Pitch and Roll. Everything is physical. In this situation, the platform has a physical orientation (in NED) and the Sensor Attitude Angles define the physical rotation of the camera gimbals with respect to the orientation of the platform.

If the Platform Relative flag is *not* set then everything is with respect to the NED system that is not physically a part of the platform and thus the Sensor Rotation Angles are with respect to NED.

Thus the transformation using Euler Angles (typically Yaw, Pitch and Roll, or other order) is either from the NED system to the Camera system or from Platform orientation to the Camera system. This is shown in the diagram below, Z.6.3-2, as well as all of the other transformations.

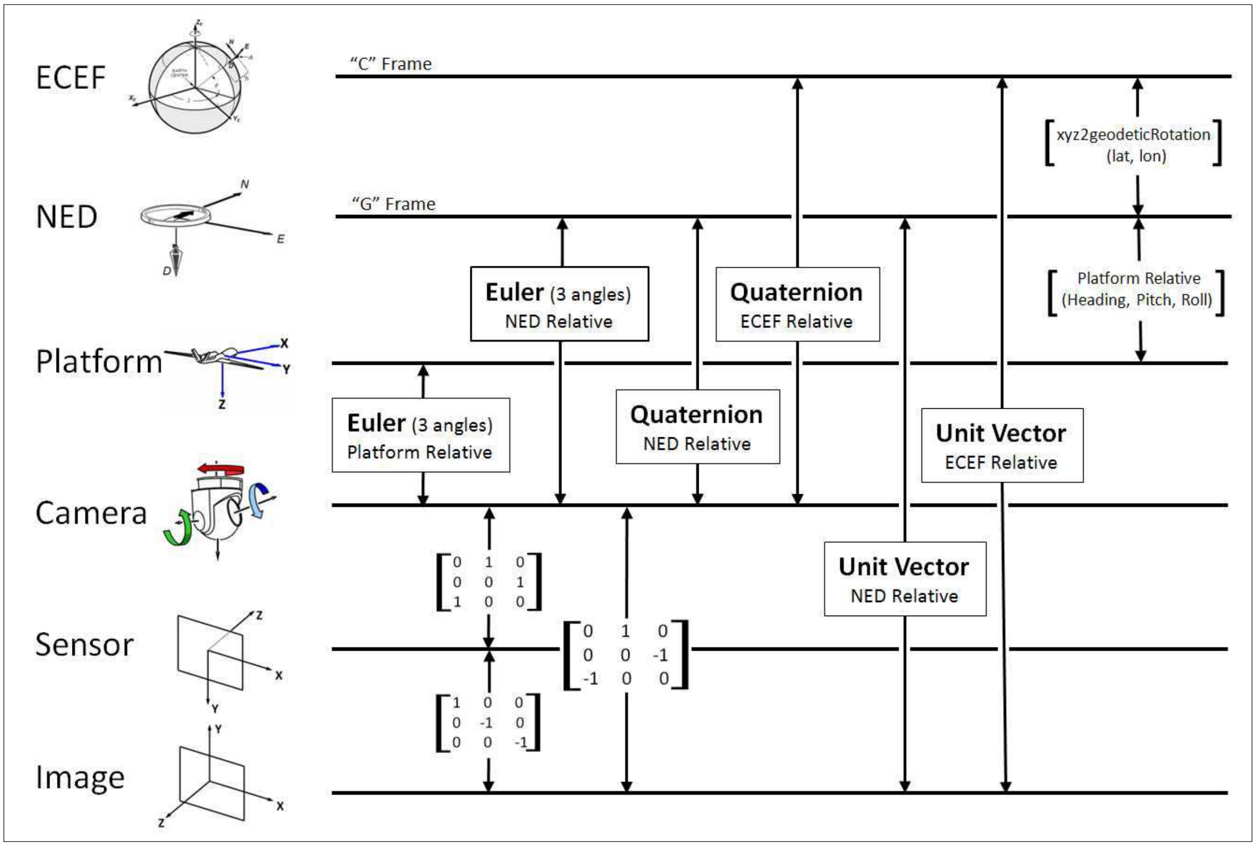


Figure Z.6-4: Transformations between the various Coordinate Systems

In the diagram, Figure Z.6-4, the two possible definitions of the Euler angles are shown as Euler (3 angles) relative to the Platform; and the Euler (3 angles) relative to the NED Coordinate system. The rotations of the physicals gimbals that hold the camera convert the information into what is being called the Camera system. The documentation and the heading of the Module 07 form indicate that this is the transformation to the Sensor system. In reality, the Euler input moves the information to the Camera system and then an implicit transformation moves that information from the Camera system into the Sensor system using the axes permutation

$$C_{Sensor}^{Camera} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Therefore, to wrap up Euler, the domain of the information in the Euler definitions is from either the NED or the Platform system to the Camera system and then implicitly down to the Sensor system.

Z.6.3.4 .The Quaternion Transformation

The Quaternion transformation is defined in Module 09 as

$$[q_1 \quad q_2 \quad q_3 \quad q_4] = \left[\sin\left(\frac{\theta_j}{2}\right)\cdot j_x \quad \sin\left(\frac{\theta_j}{2}\right)\cdot j_y \quad \sin\left(\frac{\theta_j}{2}\right)\cdot j_z \quad \cos\left(\frac{\theta_j}{2}\right)\right]$$

The first three terms represents a scaled axis of rotation and the final term contains the angle of rotation about that axis.

Just like the Euler Attitude Angles without a Platform relative transformation, the Quaternion defines the rotation from the NED frame to the Camera frame. For example, if a user were to input an Euler definition from NED to Camera using the angles (0,0,0) the corresponding Quaternion would be (0,0,0,1). Once again the conversion to the Sensor coordinate system is *implicit* and not a part of either the Euler or Quaternion input.

If the Geodetic System is defined as Cartesian or “C” then the base orientation of the transformation is from the ECEF system to the Camera system.

Z.6.3.5. The Unit Vector Transformation

The most mathematically intuitive Attitude input is the Unit Vector format, which could also be called the Direction Cosine Matrix. Unlike the Euler and the Quaternion, the Unit Vector spans a larger transformation, from either the NED or the ECEF system (depending on the Geodetic frame) down to the Image Coordinate system. The difference is two additional transformations, one from the Camera system to the Sensor system, and the second from the Sensor system to the Image system. The second transformation is

$$C_{Image}^{Sensor} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

and therefore the total is

$$C_{Image}^{Camera} = C_{Image}^{Sensor} * C_{Sensor}^{Camera} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$

One might assume that the Euler and Quaternion definitions would define the same transformation as the Unit Vector transformation. However, the Unit Vector adds an additional transformation, one from the Camera to the Image. This can be seen graphically in Figure Z.6-4.

Note that the SENSRB Data Conversion Tool provides automatic conversion between any of the coordinate frames defined in Modules 07, 08, and 09 and insures consistency when more than one transformation is specified.

Z.6.4. SENSRB DATA CONVERSION TOOL

A specially designed interactive graphic application program *SENSRB Data Conversion Tool* has been created concurrent with the specification to aid in the implementation of the extension. This program can be downloaded at http://www.gwg.nga.mil/ntb/baseline/software/demo.html.

The SENSRB tool supports all of the features of the SENSRB standard in an interactive and intuitive manner. The characteristics of each field are automatically displayed and the results controlled to help reduce errors. For example, each field has its own value range limits displayed in the appropriate units, all fields can be input in any units and will be automatically converted to the standard system of units, the full documentation on any field can be displayed by a “mouse-over”, and the relationship between fields is preserved through changes (such as automatic conversion between geocentric and geodetic frames).

This companion product not only creates and/or reads a SENSRB tagged record, it can also create or read other representations of the data, such as an Excel-like table of the information, an XML representation, and the tool can even create an interactive Google Earth display of the camera-to-earth geometry, image footprints, Point Sets, uncertainties, and more.

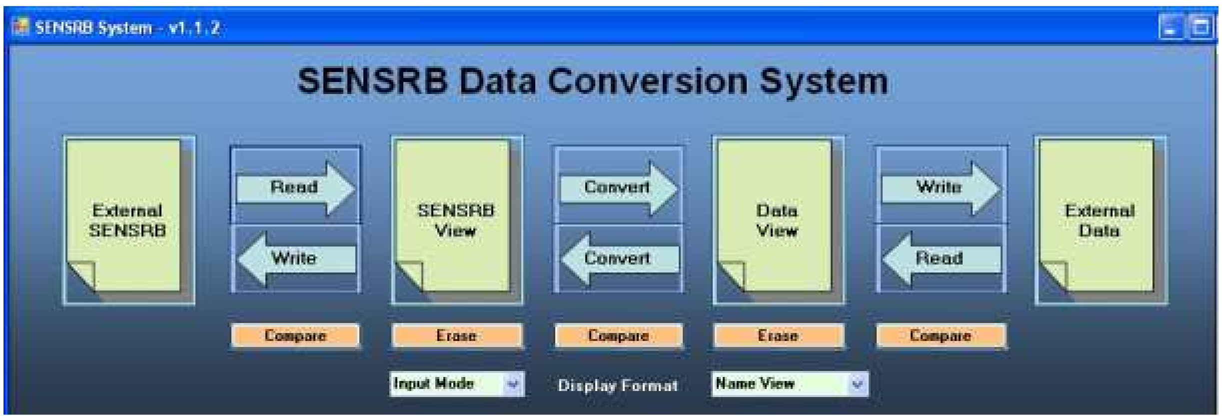


Figure Z.6-5: The iconic user interface at the top level of the SENSRB Data Conversion Tool

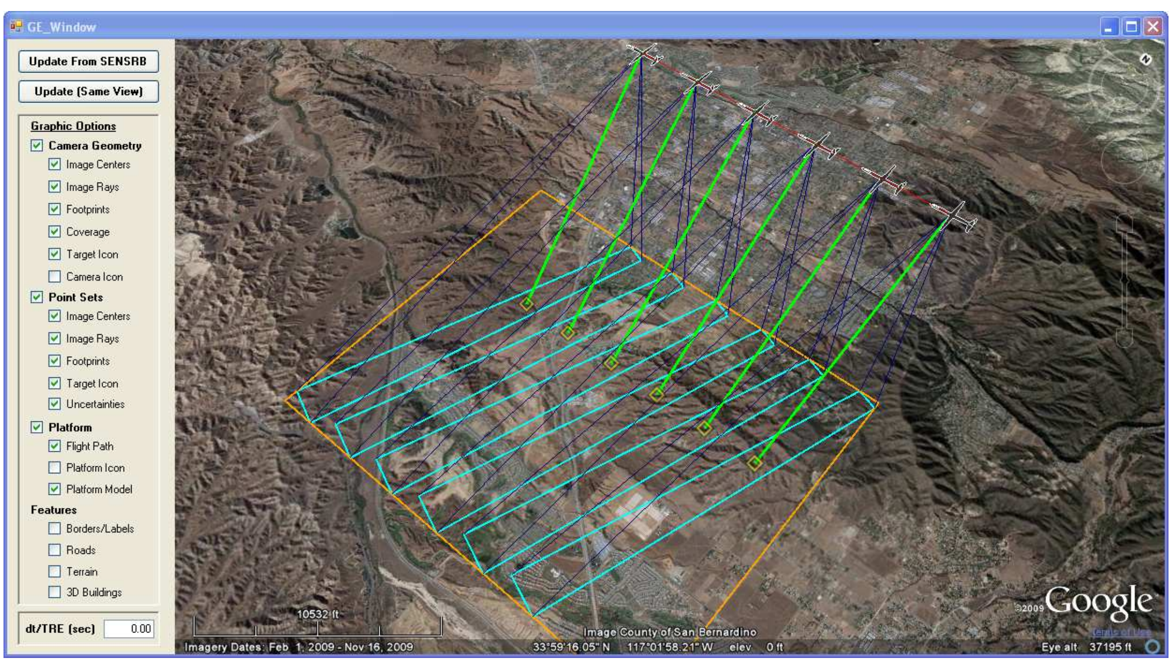


Figure Z.6-6: SENSRB Data Conversion Tool with Google Earth display of image footprints

Z.7. RECOMMENDED UNIFORM IMPLEMENTATION

As previously noted, SENSRB was developed to address a number of immediate needs. It was designed for utilization by a variety of existing and potential users. To do this, several methods for providing data are accommodated by this TRE to support legacy systems, but future implementations of this specification should evolve toward a narrower standard with less optional implementation methods. New sensor systems and those that are being upgraded should move to this recommended uniform implementation guidance. To this end, the following points are provided as the desired standard methods for all systems that are implementing SENSRB:

1. All units will be provided using the international system, (SI).
2. All angular measurements will be provided in degrees, (DEG).
3. All coordinates will be provided as geodetic coordinates, (G). (TBR01)
4. Elevations (and altitudes) will be provided in heights above ellipsoid, (HAE).
5. The sensor location will be provided directly in absolute geodetic terms with zero-valued sensor-position offsets—Module 6.
6. The sensor attitude will be provided relative to the sensor's local-level (NED) coordinate system by means of the unit-vector coordinates (direction cosines)—Module 8.
7. The platform and sensor Euler angles will also be provided for additional reference—Module 7.
8. A sensor array will be defined by focal length and array size, rather than fields of view—Module 2.
9. Sensor calibration parameters (if any) will appropriately use millimeters—Module 3.
10. Time-stamped and pixel-referenced values of Modules 12 and 13 will only occur in the first TRE.

As developers of future implementations work toward these desired methods, the objective of a more standardized implementation with greater interoperable utility will be achieved.

Z.8. ACRONYMS, SYMBOLS, AND GLOSSARY

ACRONYMS

AGL	above ground level
BCS	Basic Character Set
BCS-A	Alphanumeric
BCS-N	BCS-Numeric
BCS-NI	BCS-Numeric Integer
BCS-NPI	BCS-Numeric Positive Integer
BRD	broadband system
BRD+SWIR	Broad Spectral Bandwidth sensor in the Shortwave Infrared regime
BRD+VIS	Broad Spectral Bandwidth sensor in the Visible wavelength regime
cm	centimeters
CE	Controlled Extension
DEG	degrees
Department of Defense	DOD
D	down
E	east
EE	English Engineering Unit System
EO	electro-optical
ft	feet (ft)
ft/s	feet per second
GWG	Geospatial Intelligence Standards Working Group
HAE	height above ellipsoid
HSI	Hyperspectral Imagery
HSI+MWIR+LWIR	Hyperspectral sensor in the Midwave Infrared and Longwave Infrared
HSI+UV2SWIR	Hyperspectral sensor in the Ultraviolet to Shortwave Infrared regime.
I	identity
IC	Image Coordinate or Intelligence Community
IPON	Implementation Practices of the NITFS
In	inches
IR	infrared
LWIR	Longwave Infrared
m	meters
m/s	meters per second
MSI	Multispectral Imagery
MSI+UV2NIR	Multispectral sensor in the Ultraviolet to Near Infrared regime
MSI+VIS+SWIR	Multispectral sensor in the Visible and Shortwave Infrared regimes

MSI+VIS2LWIR	Multispectral sensor in the Visible to Longwave Infrared regime
MSI+VNIR	Multispectral sensor in the Visible-to-Near Infrared regime
MSL	mean sea level
MWIR	Midwave Infrared
NCGIS	National Center for Geospatial Intelligence Standards
NGA	National Geospatial- Intelligence Agency
NITFS	National Imagery Transmission Format Standard
NSG	National System for Geospatial-Intelligence (NSG)
NCGIS	National Center for Geospatial Intelligence Standards
NIR	Near Infrared
NTB	NITFS Technical Board
N	north
OBC	Optical Bar Camera
PAN	Panchromatic
RAD	radians
SDE	Support Data Extension
SENSR	Sensor Parameters
SHARP	Shared Airborne Reconnaissance Pod
SI	International System of Units
SMC	semi-circles
SWIR	Shortwave Infrared
SYERS-2	Senior Year Electro-optical Reconnaissance System
TRE	tagged record extension
UV	ultraviolet
URI	Uniform Resource Identifiers
UTC	Universal Coordinated Time
USI	Ultraspectral Imagery
UV	Ultraviolet
VIS	Visible
VNIR	Visible-to-Near-Infrared
VNIR+SWIR	Visible, near-infrared and short-wave infrared