STANDARDIZATION DOCUMENT

Geospatial Intelligence Standard
(Proposed)

Title:	SENSOR INDEPENDENT DERIVED DATA (SIDD) Volume 1, Design and Implementation Description Document Version 1.0 Final
Project:	Documentation to describe the implementation of various SIDD products for data products generated by Synthetic Aperture Radar (SAR) systems and their data processing elements. The scope of Volume 1 is the description needed by producers of SAR data to design a SIDD product that contains the image data and the set of metadata that describe it. The scope of Volume 2 is to define the placement of SIDD data products in the NITF V2.1 image file format. The scope of Volume 3 is to define the placement of SIDD data products in the GeoTIFF 1.0 format.
Source:	USA National Geospatial-Intelligence Agency (NGA) - Acquisition Engineering – NSG Source Integration Division (AEX) - National Center for Geospatial Intelligence Standards (NCGIS)
Target date:	DD MMM YYYY
Status:	Version 1.0 (publication ready)
Required action:	NTB and community review of proposed standard.
Reference:	GWG/NTB Coordination Page (GWG member credentials required) (https://www.gwg.nga.mil/protected/ntb/index.html)
File name:	SIDD_Design_Impl_DD_v1.0.0.pdf
Date last edited:	15 July 2011

(This page is intentionally left blank.)

NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

NGA.STND.xxxx-1_1.0
2011-07-15

NGA STANDARDIZATION DOCUMENT

SENSOR INDEPENDENT DERIVED DATA (SIDD)

Volume 1,

Design and Implementation Description Document

Specification for the design and implementation of SIDD data products.

(2011-07-15)

Version 1.0

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

CONTACTS

The following points of contact are provided for assistance in understanding the contents of this implementation profile.

NGA/AEX
P-001
12310 Sunrise Valley Drive
Reston, VA 20191-3449
(703) 755-5951

NGA/NCGIS
P-106
12310 Sunrise Valley Drive
Reston, VA 20191-3449
(703) 814-4564
NCGIS-mail@nga.mil

Change Log

Date	Version	Description	DR/CA	Developer
15 JUL 2011	1.0.0	Initial publication.	N/A	

TBR/TBD LOG

Page Number	TBD/TBR	Description	Date Addressed

FOREWORD

The suite of Sensor Independent Derived Data (SIDD) standardization documents describe the implementation of various data products generated by Synthetic Aperture Radar (SAR) systems and their data processing elements.

SAR-derived image products, and their associated metadata, are grouped around common tasks for downstream users. The SIDD documentation provides specifications for these common tasks which are designed to support basic exploitation, geographic measurements, and proper visual display. Additionally, the documentation specifies the SIDD supported coordinate systems and product image pixel arrays. The real utility of SAR image collection is in the products and measurements that may be derived from it. The quality of the pixel array data along with the set of metadata provided are critical in generating the derived products. The "sensor independence" of the SIDD product refers to the ability of the allowed pixel array and metadata options to accurately describe the image products from sensors and data processing systems. Sensor independence does NOT mean that all products have the same format for the pixel array or the same set of metadata parameters.

The SIDD documentation has been organized into four volumes:

Volume 1 is the description needed by producers of SAR data to design a SIDD product that contains the image data and the set of metadata that describe it.

Volume 2 defines the placement of SIDD data products in the NITF V2.1 image file format. Also provided is the description needed by users of SIDD products to read and properly extract the SIDD data components from a SIDD NITF product file.

Volume 3 defines the placement of SIDD data products in the GeoTIFF 1.0 image file format. Also provided is the description needed by users of SIDD products to read and properly extract the SIDD data components from a SIDD GeoTIFF product file.

Volume 4 consists of the XML artifacts used to describe the content of the various SIDD products.

A companion suite of standardization documents, collectively known as Sensor Independent Complex Data (SICD), describe standardized complex image products and measurements from which SIDD products may be derived.

The SICD and SIDD documentation and associated XML artifacts are available on the National System For Geospatial-Intelligence (NSG) Standards Registry (https://nsgreg.nga.mil).

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Scope

The Sensor Independent Derived Data (SIDD) format is designed to store Synthetic Aperture Radar (SAR) derived-image products and their associated metadata, which is grouped around common tasks for downstream users. This document, the SIDD Design and Implementation Documentation (D&I), provides specifications for these common tasks which are designed to support basic exploitation, geographic measurements, and proper visual display. Additionally, this document specifies the SIDD supported coordinate systems and product image pixel arrays.

This document covers the following:

- Image pixel array definition
- Coordinate systems
- Primitive data types
- Extensible Markup Language (XML) schema definition
- Product metadata definitions
- Coordinate mapping models
- Display guidance

1.2 Capabilities & Limitations

The SIDD format is intended for image products only. Future revisions of the SIDD format may be expanded to other products. SIDD products are not required to be built from SICD inputs, but if they are, they will contain the unaltered SICD XML metadata for reference purposes only. The SIDD metadata is intended to, but not limited to, support the following:

- Exploitation tool display
- Geographic measurements
- Library ingest & search
- Annotations

1.3 Sensor Specific Product Profiles

Each product/system has its own specifications and requirements for metadata. In order to meet these specifications, a set of metadata parameters is selected from the available SIDD metadata parameters. These parameters are referred to as the sensor-specific product profile and are contained in SIDD profile implementation documents.

1.4 Applicable Documents

The SIDD product design and implementation descriptions are contained in this and several other documents. The set of additional SIDD documents are listed in Table 1-1.

Table 1-1 SIDD Design Documentation		
Number	**Title**	**Date**
N/A	Sensor Independent Derived Data XML Schema https://nsgreg.nga.mil/	3 June 2011 version 1.0.0

The SIDD product relies, when available, on the SICD product. The set of documents that describes the SICD product design is included in Table 1-2.

Table 1-2 SICD Design Documentation		
Number	**Title**	**Date**
NGA.STND.xxxx-1_1.0	Sensor Independent Complex Data Design & Implementation Description Document https://nsgreg.nga.mil/	12 February 2010 version 1.0
NGA.STND.xxxx-1_2.0	Sensor Independent Complex Data File Format Description Document https://nsgreg.nga.mil/	12 February 2010 version 1.0
NGA.STND.xxxx-1_3.0	Sensor Independent Complex Data Image Projections https://nsgreg.nga.mil/	24 July 2009 version 1.0.0
N/A	Sensor Independent Complex Data XML Schema https://nsgreg.nga.mil/	31 March 2010 version 0.4.1

A listing of other documents referenced by the D&I document is included in Table 1-3.

Table 1-3 Other Applicable Documentation		
Number	**Title**	**Date**
NGA.STND.0014_2.2	Softcopy Image Processing Standard	11 June 2008 2.2
N/A	http://www.w3.org/TR/xmlschema-2/	28 October 2004 1.0
ISO 19125-1:2004	Geographic information -- Simple feature access -- Part 1: Common architecture	2004 19125-1
N/A	Sensor Independent Common Data Types XML Schema	3 June 2010 1.0.0
NGA.STND.0012_2.0	National System for Geospatial Intelligence metadata Foundation (NMF) - Part 1: Conceptual Schema Profile, Version 2.0 https://nsgreg.nga.mil/doc/view?i=2142	16 December 2010
N/A	XML Data Encoding Specification for Information Security Marking Metadata, Note: Use of updated versions of IC ISM will be aligned with the currently published, and GEOINT community-adopted, version of NGA.STND.0012	As specified in NGA.STND.0012

2 SIDD Image Pixel Array

The purpose of this section is to define the SIDD image pixel array, which is specified by the following items:

- Supported Pixel Types
- Visual Pixel Grid Layout
- Coordinate System

2.1 Supported Pixel Types

An image stored in the SIDD format may be represented as monochrome (8- or 16-bit), indexed color (“pseudo-color”), or 24-bit color. If the image is 8-bit monochrome or indexed color, the array will consist of byte values, each in the range of 0-255. For 24-bit data, each pixel will have three byte values in the range of 0-255, where the first byte indicates the red value, the second byte indicates the green value, and the third byte indicates the blue value.

In an indexed color pixel array, each byte represents an index into the color palette lookup table, rather than the actual pixel value. The lookup table contains a set of 256 triplet entries representing the red, green and blue values for a pixel, respectively.

The 8-bit monochrome data may also have a 256-entry lookup table mapping the byte into an output space that is between 8-bits and 16-bits.

2.2 Visual Pixel Grid Layout

The SIDD grid is defined in terms of rows and columns. The origin is zero-based and starts in the upper left corner. Movement toward the bottom of the image is defined to be in the increasing row direction and movement toward the right of the image is defined to be in the increasing column direction as shown in Figure 2-1 and Figure 2-2.

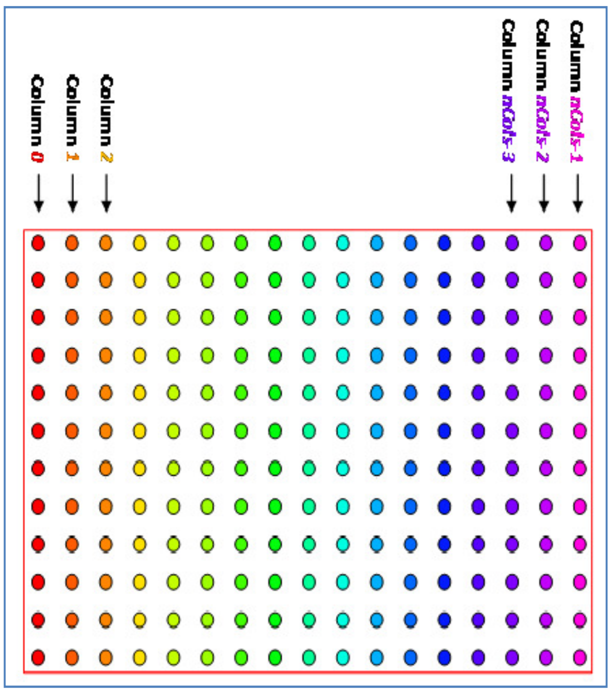


Figure 2-1 SIDD Column Grid Definition

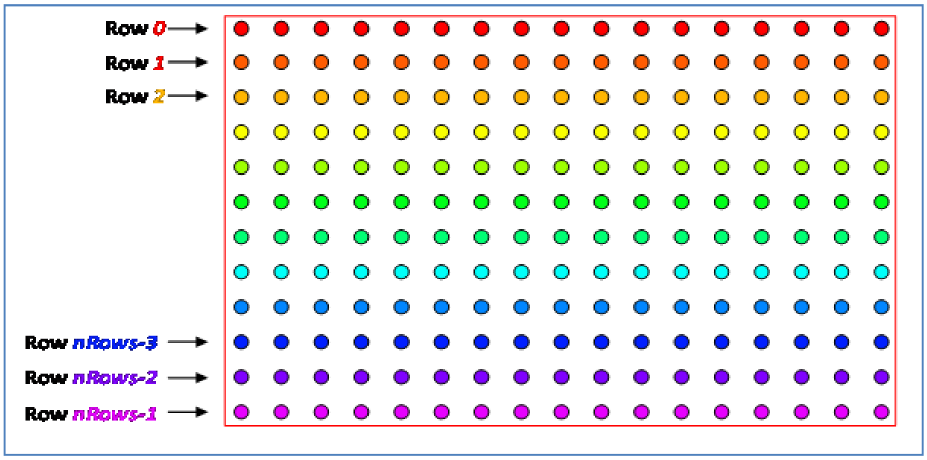


Figure 2-2 SIDD Row Grid Definition

All pixels in SIDD product pixels represent an area of data meaning that the integer pixel location is in the upper-left of the pixel and (0.5, 0.5) of the pixel is the middle of the pixel. This was selected to conform to the NITF CCS and other standards.

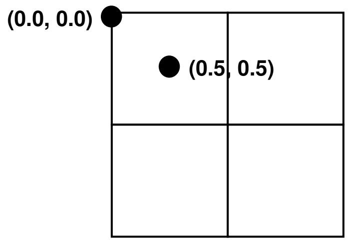


Figure 2-3 Pixel-is-area Diagram

2.3 Coordinate System

SIDD provides projection options for display of pixel amplitude data to four distinct representations: a Planar Gridded Display (PGD), Geodetic Gridded Display (GGD), Cylindrical Gridded Display (CGD), and Polynomial Fit Gridded Display (PFGD). These coordinate system layouts are intended for products which preserve the imaging geometry's layover. Furthermore, the grid layouts utilize constant sample spacing with respect to the underlying coordinate system; however, the row and column sample spacing can be unequal. The tables below provide variable definitions related to describing the PGD, GGD, CGD, and PFGD coordinate systems.

Table 2-1 Coordinate System Pixel Grid Variables		
Variable Definition	**Definition**	**Units**
nRows	Number of rows in SIDD product image	Pixels
nCols	Number of columns in SIDD product image	Pixels
$\mathbf{P}_{PGD}$	Vector in Earth-Centered, Earth-Fixed (ECEF) coordinate system. Also sometimes shown as (X_0,Y_0,Z_0). Reference point for the PGD grid type.	Meters
$\mathbf{R}_{PGD}$	Unit vector defining the increasing visual row direction in the ECEF coordinate system	Unitless
$\mathbf{C}_{PGD}$	Unit vector defining the increasing visual column direction in the ECEF coordinate system	Unitless
$\mathbf{Z}_{PGD}$	Unit vector in the ECEF coordinate system orthogonal to $\mathbf{R}_{PGD}$ and $\mathbf{C}_{PGD}$, pointing out of the earth.	Unitless
Δ_r	Row pixel sample spacing	Meters/Pixel – PGD Meters/Pixel – CGD Arcsec/Pixel – GGD
Δ_c	Column pixel sample spacing	Meters/Pixel – PGD Meters/Pixel – CGD Arcsec/Pixel – GGD
r_o	Image grid row position corresponding to the scene center point	Pixels
c_o	Image grid column position corresponding to the scene center point	Pixels
$\mathbf{P}_{ECEF}$	Vector defining an arbitrary position in the ECEF coordinate system	Meters
r	Arbitrary row position in an image	Pixels
c	Arbitrary column position in an image	Pixels
λ	Longitude	Decimal Degrees
φ	Latitude	Decimal Degrees
h	Height above ellipsoid	Meters

Table 2-1 Coordinate System Pixel Grid Variables		
Variable Definition	**Definition**	**Units**
$\mathbf{P}_{GGD}$	Reference point in a GGD pixel grid. Also referred to as $\{\varphi_0, \lambda_0, h_0\}$.	See λ, φ, and h
a	World Geodetic System (WGS)-84 semi-major axis = 6378137 meters	Meters
b	WGS-84 semi-minor axis = 6356752.31424518 meters	Meters
f	WGS-84 flattening factor = 1/298.257223563	Unitless
e_1	First eccentricity = $\sqrt{\frac{a^2-b^2}{a^2}}$	Unitless
e_2	Second eccentricity = $\sqrt{\frac{a^2-b^2}{b^2}}$	Unitless
R_c	Radius of curvature in the prime vertical = $\frac{a}{\sqrt{1-e_1^2\sin^2\varphi}}$	Meters
$\mathbf{P}_{CGD}$	Vector in ECEF coordinate system. Also sometimes shown as $(X_{CGD}, Y_{CGD}, Z_{CGD})$. Reference point for the CGD grid type.	Meters
$\mathbf{R}_{CGD}$	Unit vector defining the increasing visual row direction for the CGD grid type	Unitless
$\mathbf{C}_{CGD}$	Unit vector defining the increasing visual column direction for the CGD grid type	Unitless
$\mathbf{S}_{CGD}$	Unit vector defining the along stripmap direction in the ECEF coordinate system	Unitless
R_S	Radius used for the CGD projections	Meters

2.4 ECEF Coordinate System Definition

The origin of the ECEF coordinate system is at the center of mass of the Earth. The Z axis intersects the International Earth Rotation Service (IERS) Reference Pole. The X-axis intersects the IERS Reference Meridian and the plane passing through the origin and is normal to the Z-axis. The Y-axis completes a right-handed Cartesian coordinate system. The representation of the ECEF coordinate system is shown in Figure 2-4.

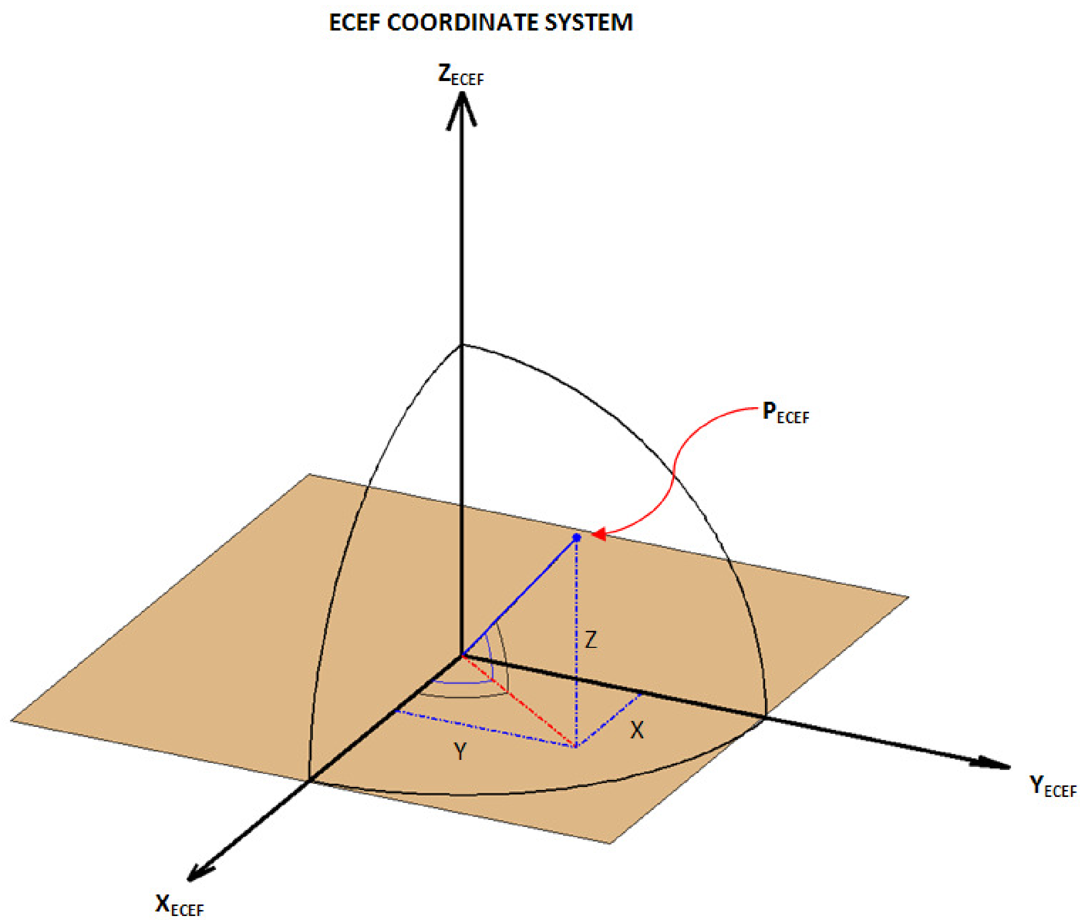


Figure 2-4 ECEF Coordinate System

2.5 Planar Gridded Display (PGD)

The Planar Gridded Display, PGD, shown in Figure 2-6, represents a row/column image sampled in a plane with constant sample spacing in the ECEF coordinate system. It is fully defined by an ECEF reference point, $\mathbf{P}_{PGD}$, and a vector normal to the plane, $\mathbf{Z}_{PGD}$. The two in-plane vectors define movement in the increasing row direction, $\mathbf{R}_{PGD}$, and the increasing column direction, $\mathbf{C}_{PGD}$, and are orthogonal. The in-plane vector relationship to the grid layout is shown in Figure 2-5. It is recommended that projection to this grid be performed such that the resulting image is "shadows-down", if possible.

The row and column sample spacing, Δ_r and Δ_c respectively, define the relationship between PGD pixel space and physical linear measurements in meters. $\mathbf{P}_{\mathbf{PGD}}$ is the PGD reference point associated with the row and column, r_0 and c_0.

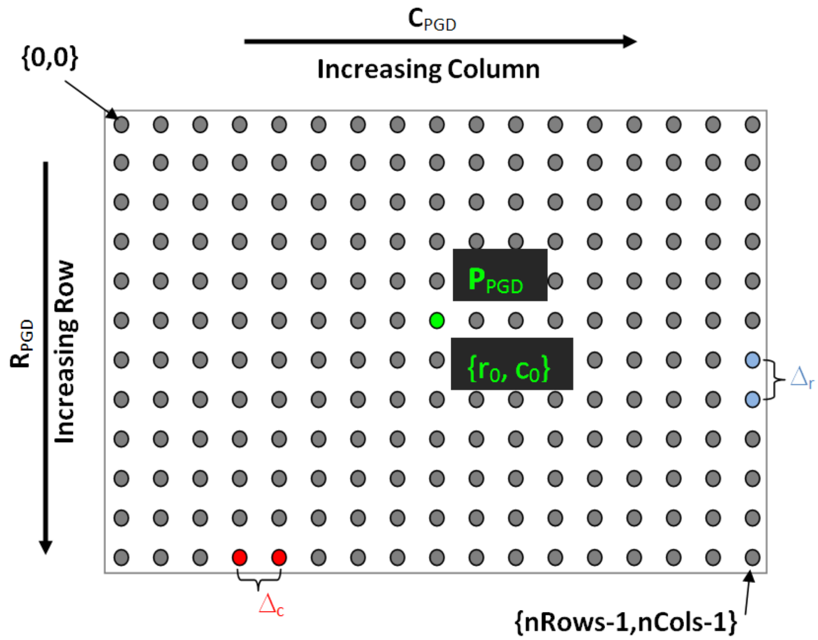


Figure 2-5 PGD Gridded Display

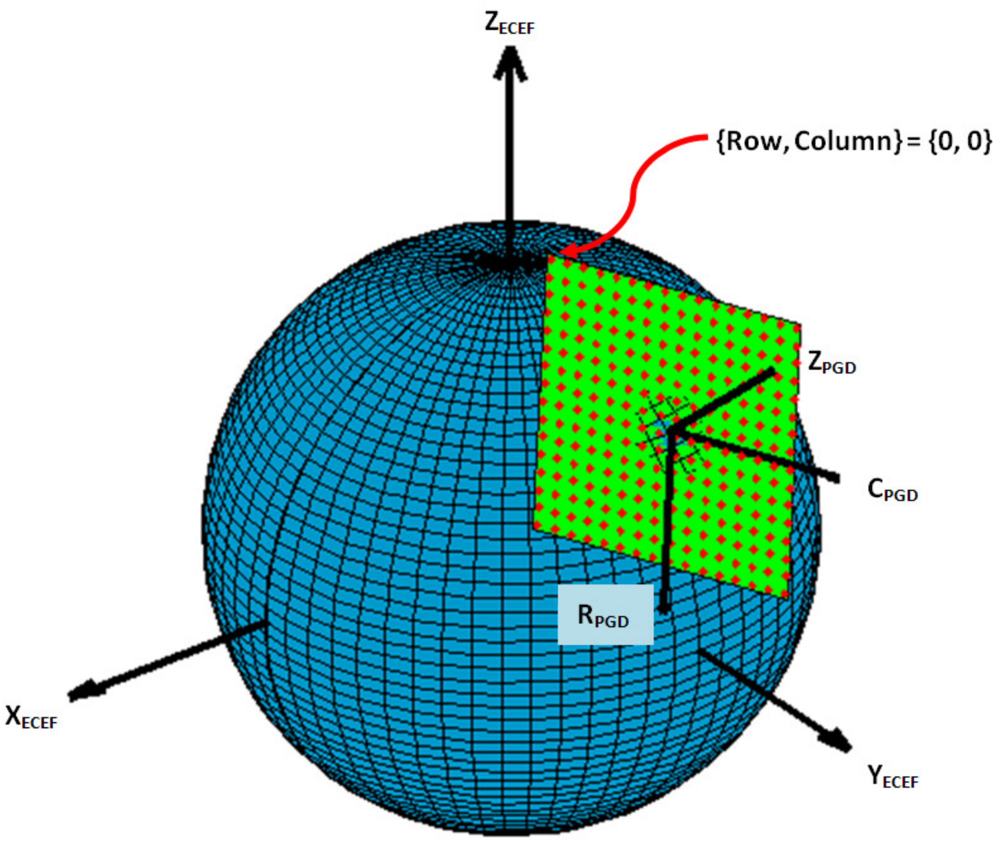


Figure 2-6 PGD Example

2.6 Geodetic Coordinate System Definition

The geodetic coordinate system shown in Figure 2-7 is based on latitude, longitude, and height above an ellipsoid and an ellipsoid model. The ellipsoid model used for SIDD is the WGS-84 ellipsoid model. The equator and prime meridian define the zero values for latitude and longitude, respectively. Geodetic latitude is the angle between the equatorial plane and a line that is normal to the reference ellipsoid.

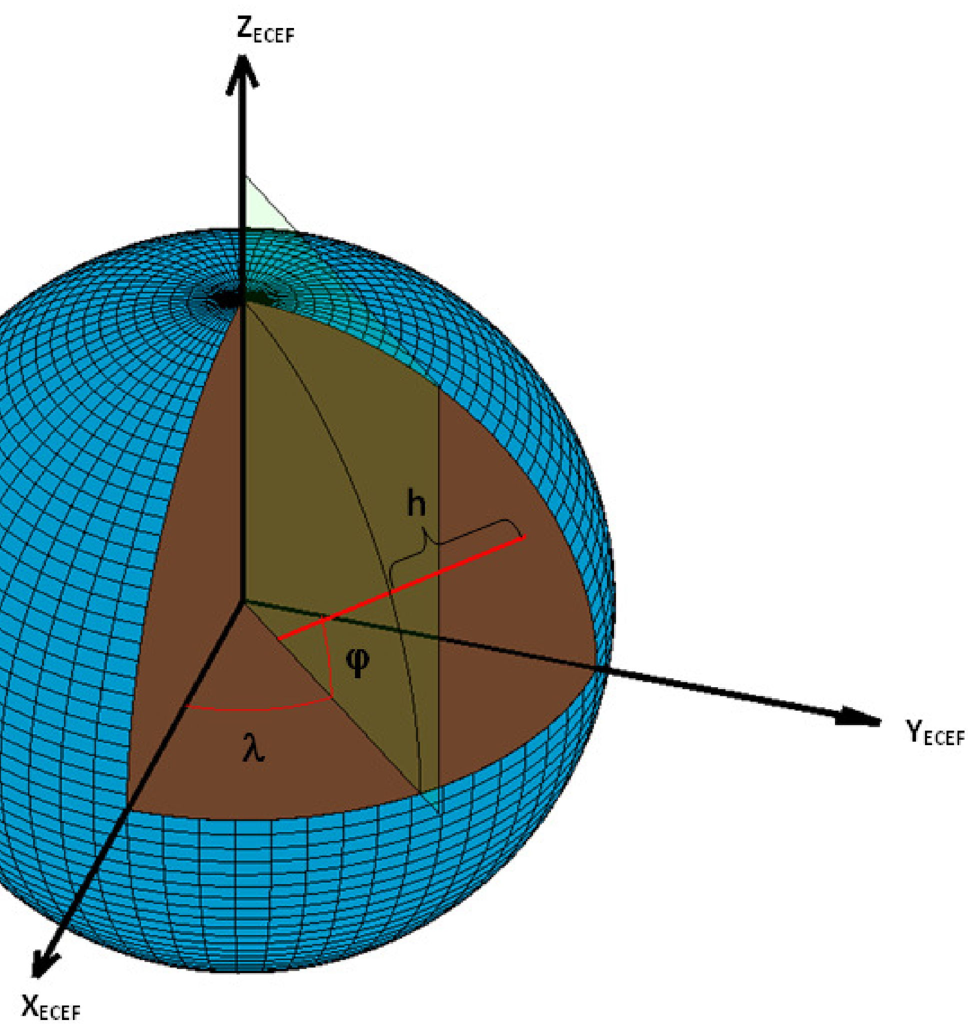


Figure 2-7 Geodetic Coordinate System

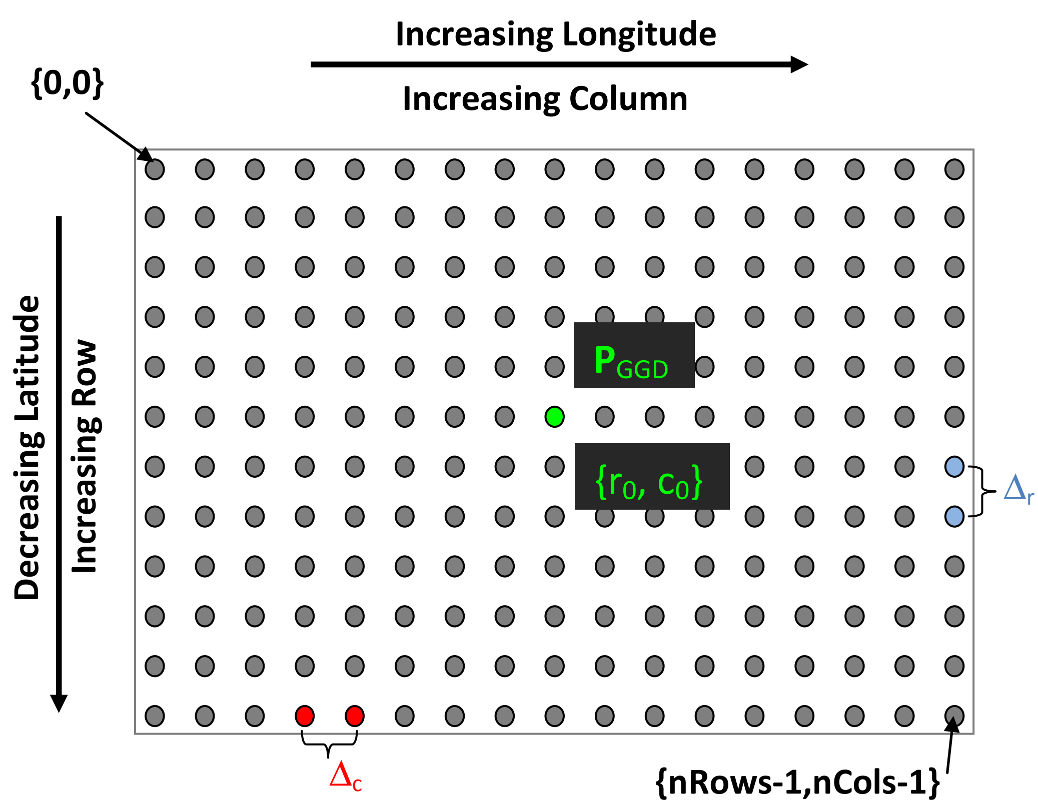
2.7 Geodetic Gridded Display (GGD)

The Geodetic Gridded Display, GGD is a row/column image coordinate system (see Figure 2-8) and has an associated set of geodetic coordinates, which are expressed by latitude, longitude, and height (see Figure 2-9). The GGD is fully defined by a geodetic reference point, $\mathbf{P}_{GGD}$, and the WGS-84 ellipsoid model.

The row and column sample spacing, Δ_r and Δ_c respectively, define the relationship between GGD pixel space and angular measurements, e.g. arc seconds. The association with the geodetic reference point, $\mathbf{P}_{GGD}$, is a particular GGD row and column, r_0 and c_0. An adjustment in either row or column pixel location is represented by a corresponding adjustment in latitude and longitude. Each image coordinate is thus directly associated with a geodetic latitude and longitude.

For image footprints which are not aligned in latitude and longitude, for which the elimination of the surrounding black-fill is desired, the *GeometricChip* metadata can be used to represent the rotation of the grid space (Section 4.2.7.2).

Figure 2-8 Geodetic Gridded Display



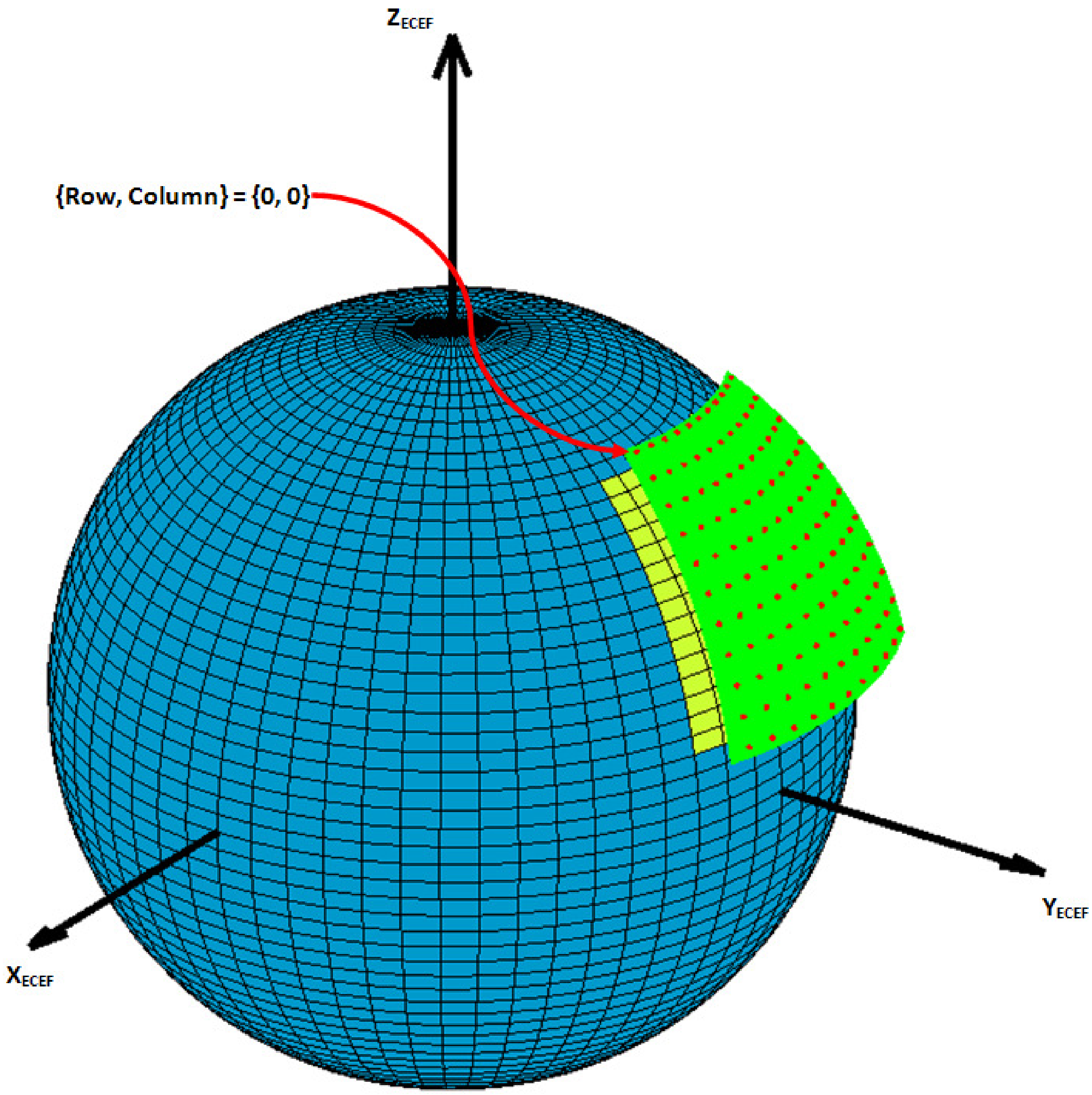


Figure 2-9 GGD Example

2.8 Cylindrical Gridded Display (CGD)

The Cylindrical Gridded Display represents an image sampled on a cylindrical surface. This grid type is useful when the imaging mode is (dynamic) stripmap and mapping on the PGD would result in large image distortions. The cylinder's axis is parallel to the cross-stripmap direction and the cylinder's radius is designed to match an inflated WGS-84 ellipsoid. Note that this inflated ellipsoid is only used to determine the cylinder's radius; the pixels themselves do not lie along an inflated ellipsoid.

The CGD is fully defined by a CGD reference point, $\mathbf{P}_{CGD}$, a cylinder radius, R_s, and the along stripmap direction, $\mathbf{S}_{CGD}$. If a cylinder radius is not supplied, then a radius is computed by an inflated ellipsoid. The row and column sample spacing, Δ_r and Δ_c respectively, define the relationship between CGD pixel space and linear distance. Unlike the PGD, where $\mathbf{R}_{PGD}$ and $\mathbf{C}_{PGD}$ are allowed to be in any orientation, the CGD basis vectors are constrained to one orientation; $\mathbf{R}_{CGD}$ is aligned in the cross-stripmap direction and $\mathbf{C}_{CGD}$ is in the along-stripmap direction.

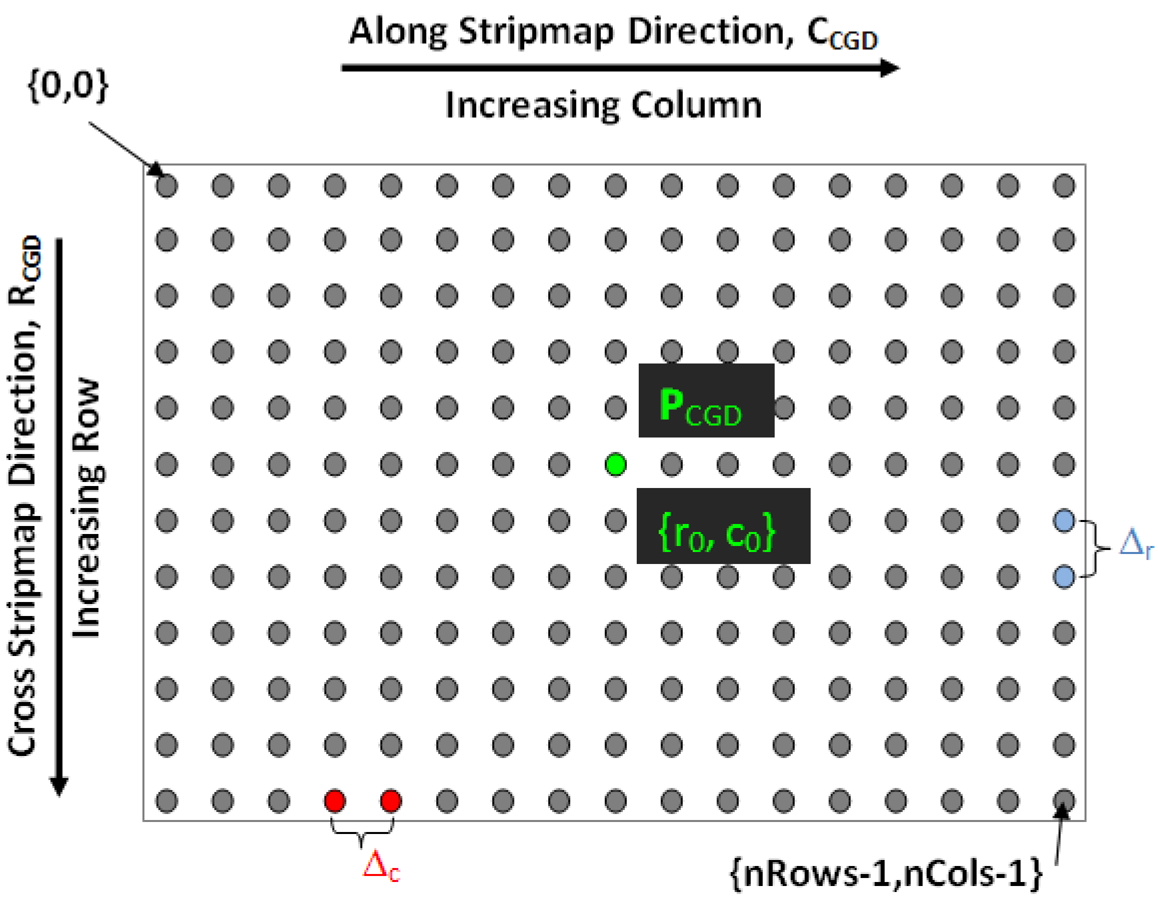


Figure 2-10 CGD Image Example

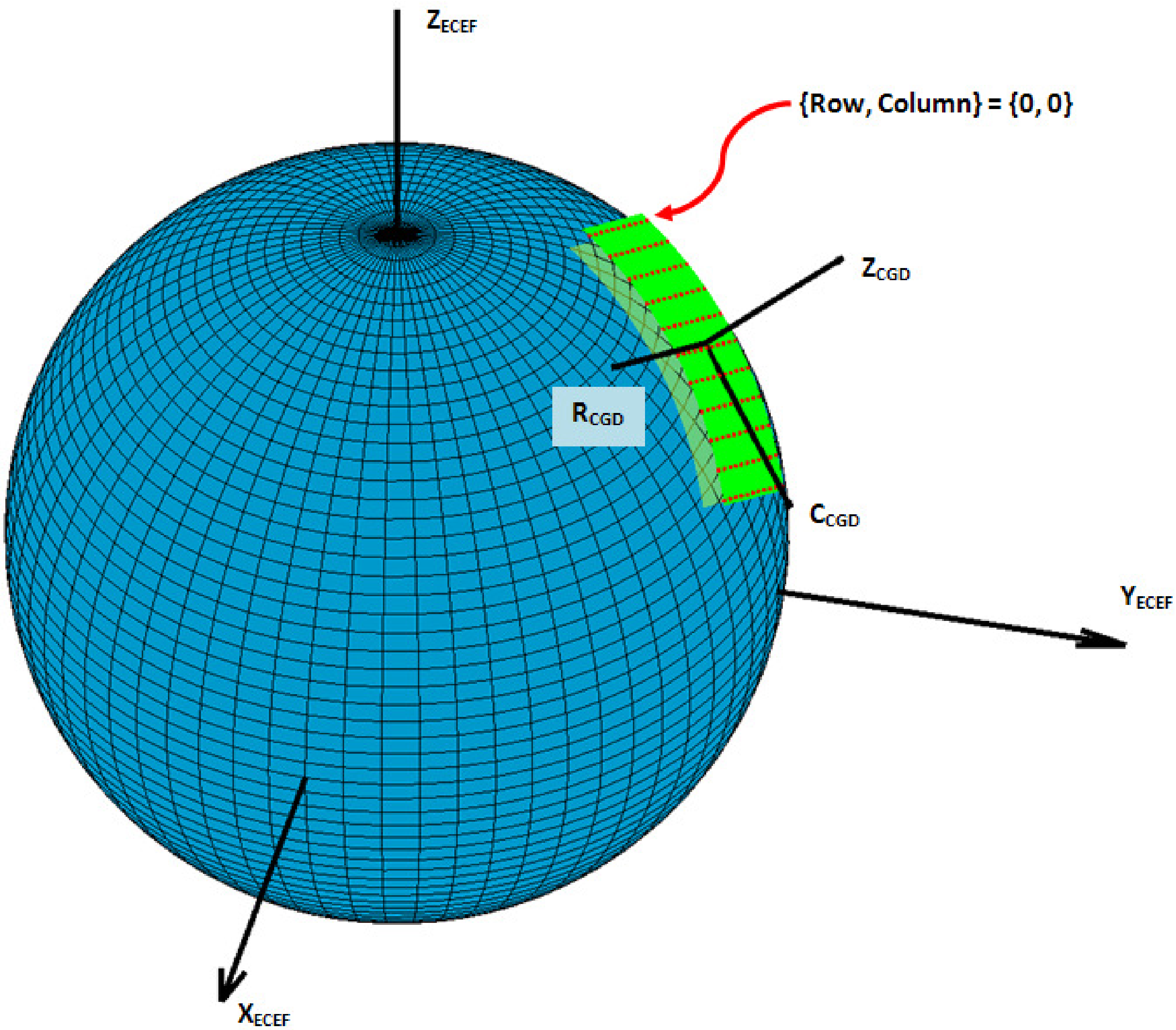


Figure 2-11 CGD Example

2.8.1 CGD inflated ellipsoid

The inflated ellipsoid has the same shape as the WGS-84 ellipsoid model but with a modified equatorial radius. The new equatorial radius is computed so that $\mathbf{P}_{CGD}$ is on the inflated ellipsoid. The computation for the inflated ellipsoid equatorial radius is as follows:

$$a' = \sqrt{\left(X_{CGD}^2 + Y_{CGD}^2 + \frac{Z_{CGD}^2}{(1-f)^2}\right)}$$

The modified geodetic coordinates are computed below:

$$\lambda' = \lambda$$

$$\varphi' = \tan^{-1}\frac{Z_{CGD}}{(1-f)^2\sqrt{X_{CGD}^2 + Y_{CGD}^2}}$$

Note that this equation is correct if and only if the ECEF point (X_{CGD}, Y_{CGD}, Z_{CGD}) lies on the ellipsoid. The ECEF point can be assumed to lie on the ellipsoid for the CGD case.

2.8.2 Modified Local East, North, Up Coordinate system

The inflated ellipsoid necessarily changes the direction of the east, north, and up directions. These bases must be recomputed at the modified geodetic coordinates.

$$\boldsymbol{E}' = [-\sin\lambda' \quad \cos\lambda' \quad 0]$$
$$\boldsymbol{N}' = [-\sin\varphi'\cos\lambda' \quad -\sin\varphi'\sin\lambda' \quad \cos\varphi']$$
$$\boldsymbol{U}' = [\cos\varphi'\cos\lambda' \quad \cos\varphi'\sin\lambda' \quad \sin\varphi']$$

2.8.3 CGD Bases computations

The CGD bases need to be in a plane that is tangent to the inflated ellipsoid. Given a (dynamic) stripmap direction, $\mathbf{S}_{CGD}$, the CGD bases are computed below:

$$\alpha = \tan^{-1}\frac{\boldsymbol{E}'\cdot\boldsymbol{S}_{CGD}}{\boldsymbol{N}'\cdot\boldsymbol{S}_{CGD}}$$

$$\boldsymbol{C}_{CGD} = \cos\alpha\,\boldsymbol{N}' + \sin\alpha\,\boldsymbol{E}'$$
$$\boldsymbol{R}_{CGD} = \boldsymbol{C}_{CGD}\times\boldsymbol{U}'$$

2.8.4 Modified Radius of Curvature in the Meridian

The radius of curvature in the North-South direction at a specified latitude is shown below:

$$R_N = \frac{a'(1-e_1^2)}{(1-e_1^2\sin^2\varphi')^{1.5}}$$

2.8.5 Modified Radius of Curvature in the Prime Vertical

The radius of curvature in the East-West direction at a specified latitude is shown below:

$$R_E = \frac{a'}{\sqrt{1-e_1^2\sin^2\varphi'}}$$

2.8.6 Radius of Curvature in the Along Stripmap Direction

The radius of curvature in the along-stripmap direction is computed below; this is equivalent to the cylinder's radius:

$$\frac{1}{R_S} = \frac{\cos^2 \alpha}{R_N} + \frac{\sin^2 \alpha}{R_E}$$

Note that the radius of curvature R_S, derived above, should be utilized with the CGD unless a different one is provided in the XML metadata.

2.9 Polynomial Fit Gridded Display (PFGD)

The Polynomial Fit Gridded Display (PFGD) is an arbitrary surface which can be represented by a polynomial. The PFGD metadata provides for an approximation to the rigorous projection models. *If a sensor's metadata supports creation of a rigorous geometry model, then one of the previous grid types should be used instead of the Polynomial Fit Gridded Display. This model is reserved for backwards compatibility with systems that do not provide the metadata necessary for a rigorous projection.* The approximation uses a set of polynomials which expresses row and column pixel locations as a function of latitude and longitude. In addition, a set of polynomial approximations are included in PFGD to express latitude, longitude and an approximate height as a function of row and column pixel locations.

3 Coordinate Transformations

The image formation process generates a two-dimensional projection of the three-dimensional imaged scene. Each resolution cell is a combination of a range/range-rate projection (see *Sensor Independent Complex Data Image Projections*) and image formation algorithm dependent geometrical distortions. The goal of the SIDD grids is to remove all image formation algorithm dependent geometrical distortions; however, it is ultimately up to the end user to determine the ideal grid for exploitation. It is important to emphasize that the PGD, GGD, CGD, and PFGD coordinates defined in Section 2 are not necessarily terrain surface locations. In order to determine accurate terrain surface locations, knowledge of the underlying terrain must be provided (via a priori terrain height information).

Section 3.1 describes the difference between the pixel grid coordinates and the sensor model grid coordinates. The ECEF coordinates lying within the pixel grid can be derived using the equations outlined in Sections 3.2 to 3.11. Mapping of the SIDD grids to the projection models is outlined in Section 3.12.

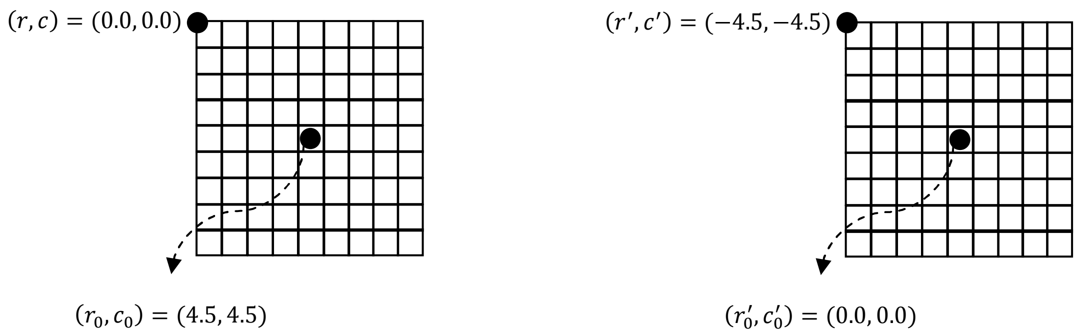
3.1 SIDD Sensor Model Grid

SIDD heavily leverages the *Sensor Independent Complex Image Data Image Projections* specification (Table 1-2) for its projections equations. SIDD projection utilizes a reference point-centered representation of the sensor model grid for projection. This means that there is a translation between pixel grid coordinates {r, c} and sensor model coordinates {r', c'} and distances {d_r, d_c} that must be accounted for prior to application of the equations.

$$r' = r - r_0$$

$$c' = c - c_0$$

The above equations define the translations to sensor model coordinates {r', c'}.



(a) Pixel Grid (b) Sensor Model Grid

Figure 3-1 Differences between pixel and sensor model grids

Distance within the sensor model grid can now be defined as

$$d_r = \Delta_r * r'$$

$$d_c = \Delta_c * c'$$

Note that because each of the different grid types utilize sample spacing in different units, the above distance computations also result in units based upon the grid type.

SIDD reuses the data types from the SICD specification to promote consistency between the two specifications. Thus particular metadata items which are functions of sensor model coordinate distances, for example, the *TimeCOAPoly* (see Table 4-23) are built from distances {d_r, d_c} in the sensor model space. Please review the *Sensor Independent Complex Data Image Projections* and *Sensor Independent Complex Data Design & Implementation* specifications for more details.

3.2 PGD Pixel to ECEF Coordinate Conversion

The conversion of a PGD pixel grid coordinate {r, c} (and thus sensor grid distance {d_r, d_c}) to an ECEF coordinate $\mathbf{P}_{ECEF}$, {x, y, z}, is shown below.

$$\boldsymbol{P}_{ECEF} = \boldsymbol{P}_{PGD} + d_r * \boldsymbol{R}_{PGD} + d_c * \boldsymbol{C}_{PGD}$$

3.3 ECEF Coordinate to PGD Pixel Conversion

The conversion of an ECEF coordinate $\mathbf{P}_{ECEF}$, {x, y, z}, to a PGD pixel coordinate is shown below.

$$r = r_0 + \frac{(\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{PGD}) \cdot \boldsymbol{R}_{PGD}}{\Delta_r}$$

$$c = c_0 + \frac{(\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{PGD}) \cdot \boldsymbol{C}_{PGD}}{\Delta_c}$$

3.4 GGD Pixel to Geodetic Coordinate Conversion

The conversion of a GGD pixel grid coordinate {r, c} (and thus sensor grid distance {d_r, d_c}) to a geodetic coordinate is shown below. A constant height above the ellipsoid is used and is set from the reference point $\mathbf{P}_{GGD}$. In the equations below, the sample spacing is assumed to be in arc seconds, φ_0 and λ_0, are in decimal degrees and h_0 is in meters.

$$\varphi = \varphi_0 - \frac{d_r}{3600}$$

$$\lambda = \lambda_0 + \frac{d_c}{3600}$$

$$h = h_0$$

3.5 Geodetic Coordinate to GGD Pixel Conversion

The conversion of a geodetic coordinate to a GGD pixel is shown below. Again, a constant height above the ellipsoid is assumed, as well as sample spacing in arc seconds.

$$\mathrm{r} = \mathrm{r}_0 + \frac{3600(\varphi_0 - \varphi)}{\Delta_\mathrm{r}}$$

$$c = c_0 + \frac{3600(\lambda - \lambda_0)}{\Delta_c}$$

3.6 Geodetic to ECEF Coordinate Conversion

The relationship between the WGS-84 ellipsoid model and the ECEF coordinate system is shown below.

$$\mathrm{X} = (R_c + h)\cos(\varphi)\cos(\lambda)$$
$$\mathrm{Y} = (R_c + h)\cos(\varphi)\sin(\lambda)$$
$$\mathrm{Z} = \left(\frac{b^2}{a^2}R_c + h\right)\sin(\varphi)$$

3.7 ECEF Coordinate to Geodetic Coordinate Conversion

The relationship between the ECEF coordinate system, {X, Y, Z}, and the geodetic coordinate system {φ, λ, h} is described below. The four-quadrant inverse tangent (arctangent) function, atan2(Y,X), with range on the interval $[-\pi, \pi]$, is used.

$$\lambda = \text{atan2}(Y, X)$$

$$D_{XY} = \sqrt{X^2 + Y^2}$$

$$\theta = \text{atan2}(a * Z, b * D_{XY})$$

$$\tan \varphi_{i+1} = \frac{Z + e_2^2 b \sin^3 \theta}{D_{XY} - e_1^2 a \cos^3 \theta}$$

$$\tan \theta_{i+1} = (1 - f) \tan \varphi_{i+1}$$

This iterative procedure is terminated when $\left|\tan \varphi_{i+1} - \tan \varphi_i\right| \le \varepsilon$, where ε is small. The height above the WGS-84 ellipsoid is then found by the following equation:

$$h = \frac{D_{XY}}{cos(\varphi)} - R_C$$

3.8 CGD Pixel to ECEF Coordinate Conversion

The conversion of a CGD grid pixel coordinate {r, c} (and thus sensor grid distance {d_r, d_c}) to an ECEF coordinate {x, y, z}, is shown below.

$$\theta = \frac{d_c}{R_S}$$

$$\boldsymbol{P}_{ECEF} = \boldsymbol{P}_{CGD} + d_r * \boldsymbol{R}_{CGD} + R_S * \sin\theta * \boldsymbol{C}_{CGD} + R_S * (\cos\theta - 1) * \boldsymbol{U}'$$

3.9 ECEF Coordinate to CGD Pixel Conversion

The conversion of an ECEF coordinate, $\boldsymbol{P}_{ECEF}$, to a CGD pixel coordinate {r,c} is shown below.

$$r = r_0 + \frac{(\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{CGD}) \cdot \boldsymbol{R}_{CGD}}{\Delta_r}$$
$$c_c = (\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{CGD}) \cdot \boldsymbol{C}_{CGD}$$
$$c_u = (\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{CGD}) \cdot \boldsymbol{U}'$$
$$\theta = \cot^{-1} \frac{c_u + R_S}{c_c}$$
$$c = c_0 + \frac{R_S \theta}{\Delta_c}$$

3.10 PFGD Latitude and Longitude to Row and Column Conversion

The following polynomials convert latitude and longitude into row and column pixel locations.

$$r = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} \lambda^m \varphi^n$$

$$c = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} \varphi^m \lambda^n$$

3.11 PFGD Row and Column to Latitude, Longitude and Height Conversion

The following polynomials convert row and column pixel locations to latitude, longitude and altitude.

$$\lambda = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} r^m c^n$$

$$\varphi = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} r^m c^n$$

$$h = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} r^m c^n$$

3.12 Projection of SIDD data

Projection of SIDD data (image-to-scene and scene-to-image) can be accomplished via the equations outlined in the *Sensor Independent Complex Data Image Projection* document. The key connection is the computation of the R/Rdot contour for a particular pixel grid coordinate {r,c}.

For SIDD pixel grids PGD, GGD, and CGD (PFGD does not support a formal model), the R/Rdot contours are specified as follows:

$$R_{COA}^{TGT} = \left| ARP_{COA}^{TGT} - P_{ECEF} \right|$$

$$Rdot_{COA}^{TGT} = \left(\frac{VARP_{COA}^{TGT}}{R_{COA}^{TGT}} \right) \cdot \left(ARP_{COA}^{TGT} - P_{ECEF} \right)$$

Where P_{ECEF} is defined in the above sections for each grid type. ARP_{COA}^{TGT} and $VARP_{COA}^{TGT}$ are computed as specified in the *Sensor Independent Complex Data Image Projection* document.

4 SIDD Metadata

The purpose of this section is to define the SIDD metadata. The definition of SIDD metadata includes the items below:

- XML primitives
- Groupings

- Complex structures
- Metadata parameter definitions
- Primitive types

The metadata hierarchical structure for SIDD and SICD is XML. The foundation of the SIDD format is constructed from primitive data types. Primitive data types are reusable structures that define the type of data contained within the metadata parameter and are defined in Section 4.1.2 and in the following documentation: http://www.w3.org/TR/xmlschema-2/. A metadata parameter refers to a piece of information, such as sample spacing, that is required for downstream usage of a SIDD product. The parameters are further organized into complex structures. Each complex structure can contain other parameters, complex structures, or both that are linked together by common metadata. The top level complex structures within the SIDD are referred to as groupings to distinguish them from lower level complex structures. The SIDD metadata is organized into groupings centered on exploitation tasks, such as the *Display* grouping shown in Figure 4-12. An example of the naming convention and the general layout of the different types is shown in Figure 4-1.

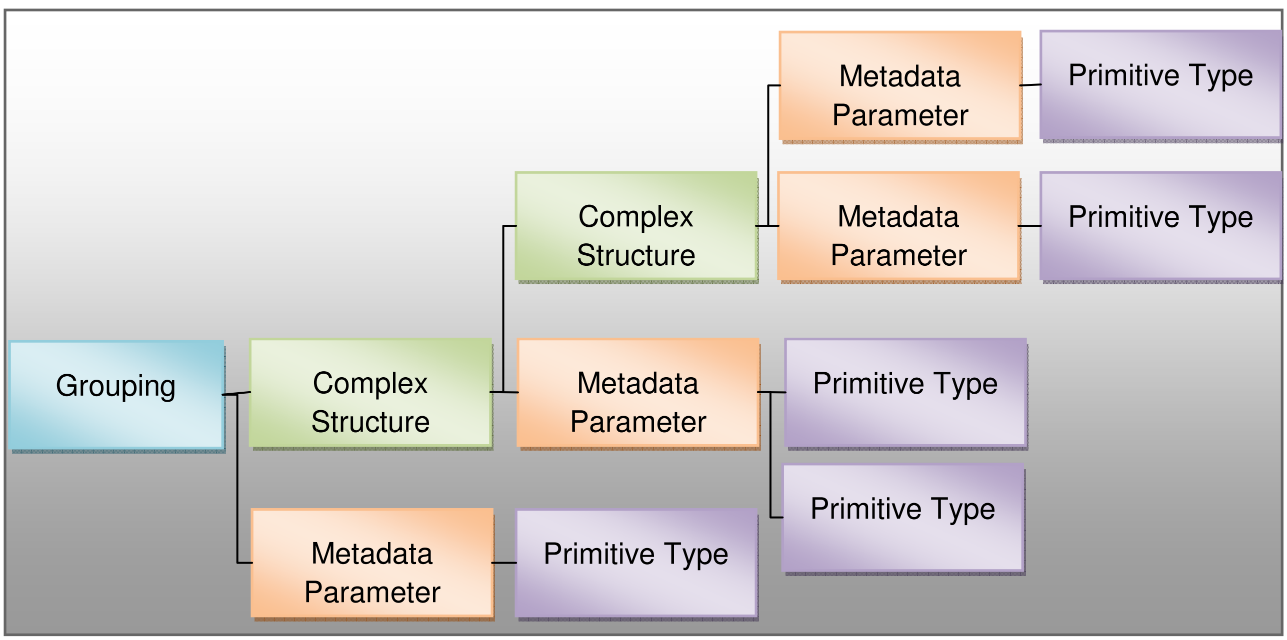


Figure 4-1 Example of Naming Convention and General Schema Layout

4.1 XML Metadata Types

This section provides the following information

- XML schema interpretation
- XML primitive type overview
- XML primitive type definitions

4.1.1 XML Schema Interpretation

The purpose of the next three subsections is to provide instructions for reading the XML schema diagrams provided throughout this document.

4.1.1.1 Required vs. Optional

The schema diagrams use dashed lines for optional parameters or complex structures and solid lines for required parameters or complex structures. The required and optional parameters / complex structures are enforced through both the schema and the documentation. Figure 4-8 shows an example of both optional and required complex structures. An example of an optional grouping is the *ErrorStatistics* (Section 4.2.8) grouping, and an example of a required grouping is the *ProductCreation* (Section 4.2.1) grouping.

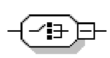
The schema does not enforce conditional parameters or conditional complex structures; therefore, the schema diagrams don't indicate conditional parameters or conditional complex structures. Conditional parameters are only enforced through system-specific documentation (profiles).

If a tag is marked as required and its parent tag is marked as optional, then the tag is only required if its parent tag is included. If a profile has a stricter requirement than the SIDD schema or D&I documentation, products created using the profile should follow the profile's requirements.

4.1.1.2 Repeatable

A repeatable type is shown with an n (n = 0, 1 … N-1) to infinity symbol (∞) under the field name. An example of a repeatable type is shown Figure 4-3 with the *Poly1DType* primitive type. In this example, n starts at one and spans to infinity. The type specifies that at least one *Coef* parameter must exist and up to infinity parameters can also exist.

4.1.1.3 Choice

A choice option is shown with a  node. The schema choice option means that one of the set of parameters that connect to the node must exist.

4.1.1.4 Units

Units are defined under the type or parameter in the schema layout. Table 4-1 list the units and abbreviations used in this document.

Table 4-1 Unit Abbreviations

Units	Abbreviations
Seconds	s
Meters	m
Meters per second	m/s
Hertz	Hz

Hertz per second	Hz/s
Decimal degrees	dd
Cycles	cyc
Cycles per meter	cyc/m
Cycles per meter squared	cyc/m^2
Radians	rad
Radians per second	rad/s
Samples per second	samples/s
Decibel	dB
Degrees	deg

4.1.2 XML Types Overview

The SICommon Types XML schema defines a set of reusable primitive types that declare XML metadata parameter types. The SICommon Types are the reusable types that are shared between the SICD and SIDD XML schema. There are two basic forms of primitive types: complex and simple. A simple primitive has only one child and can define a specific data type. The simple primitive types are defined in the following documentation: http://www.w3.org/TR/xmlschema-2/. In addition, some of the simple primitive types can be restricted to a specific range of values. For example *Neg180To180* is a parameter that usually refers to an angle that ranges in values from -180 degrees to +180 degrees. The restricted simple primitive types are defined in Table 4-2.

A complex primitive type is used for storing complex information, such as polynomials. An example of a complex primitive type is *Poly1DType* which stores an arbitrary number of coefficients and the order of the polynomial. The complex primitive types are defined in Section 4.1.4.

The SIDD and SICD common schema types are listed in Table 4-3 and in Sections 4.1.3 and 4.1.4.

The following table lists the restricted primitive types.

Table 4-2 Restricted Primitive Types

Type	Definition
Neg180To180Type	*Neg180To180Type* restricts inclusively, double values from -180 to 180.
Neg90To90Type	*Neg90To90Type* restricts inclusively, double values from -90 to 90.

4.1.3 Complex SICommon Types

The following table describes the SICommon Types that are used by SIDD. The SICommonTypes are the XML parameters that are shared between the SIDD and SICD schemas.

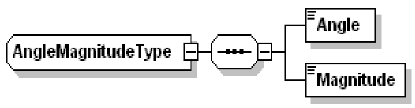
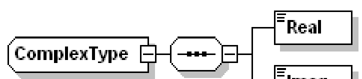
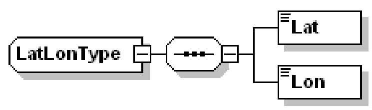
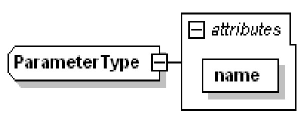
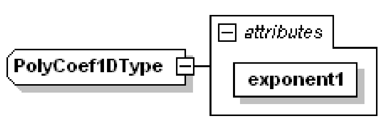
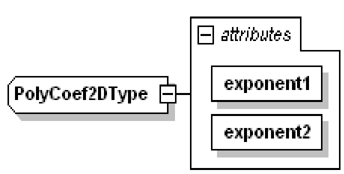
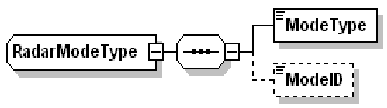
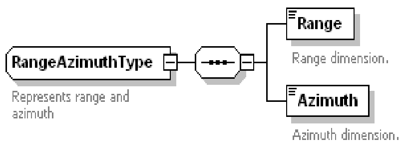
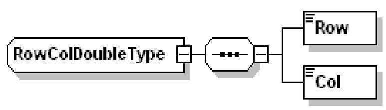
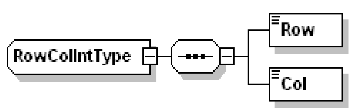
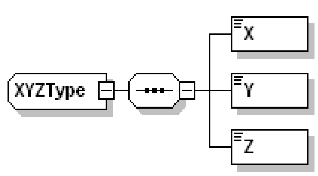
Table 4-3 of SICommon Types		
Type	**Schema Layout**	**Definition**
AngleMagitude Type		The *AngleMagnitudeType* is composed of angle and magnitude values. The *Layover* parameter is an implementation example for populating the *AngleMagnitudeType*. <Layover> <Angle>-79.5</Angle> <Magnitude>1.348E0</Magnitude> </Layover>
ComplexType		The *ComplexType* is composed of Real and Imag values used to represent the real and imaginary parts of a complex number, respectively. They are of type double.
LatLonType		The *LatLonType* is composed of Lat and Lon values representing latitude and longitude values, respectively. These parameters are of type double.
ParameterType		The *ParameterType* is used throughout the standard to include information which is often sensor/system specific that is not already included in the metadata. The *ModuleParameter* in the *ProductProcessingType's ProcessingModuleType* is an example for populating a *ParameterType* primitive type. <ModuleParameter name="C1_Weighting"> 1.0 </ModuleParameter>
PolyCoef1DType		The *PolyCoef1DType* is a representation of a coefficient for a 1D polynomial.
PolyCoef2DType		The *PolyCoef2DType* is a representation of a coefficient for a 2D polynomial.

Table 4-3 of SICommon Types		
Type	**Schema Layout**	**Definition**
RadarModeType		The *RadarModeType* specifies the optional ModeID, which is a string used to specify system-specific mode identifiers, and the ModeType, which is a required parameter referring to the collection type via one of the following enumerations: SPOTLIGHT, STRIPMAP, DYNAMIC STRIPMAP, SCANSAR.
RangeAzimuthType		The *RangeAzimuthType* specifies range and azimuth values as type double. The *Resolution* parameter within the *ExploitationFeaturesType's ExploitationFeaturesCollectionType Information* grouping is an example the *RangeAzimuthType*. <Resolution> <Range>3.0E0</Range> <Azimuth>6.0E0</Azimuth> </Resolution>
RowColDoubleType		The *RowColIntType* is which is composed of row and col values of type integer. The *RowColDoubleType* has the same structure as the RowColIntType; however the *Row* and *Col* are doubles. The *ChipSize* parameter is an implementation example for populating the *RowColDoubleType*. <ChipSize> <Row>1.5000E4</Row> <Col>3.0000E4</Col> </ChipSize>
RowColIntType		
XYZType		It is a complex primitive type with the parameters X, Y, and Z of type double. For example, the *GeographicProjectionType's ReferencePoint* contains an *ECEF* parameter which is a *XYZType*. <ECEF> <X>-7.425071E5</X> <Y>-5.4627385E6</Y> <Z>3.1967065E6</Z> </ECEF>

4.1.4 Expanded Complex Primitive Data Types

This section defines complex primitive types which have expanded structures. For example, *FootprintType* has a child called *Vertex* which also has the following children: *Lat* and *Lon*.

4.1.4.1 FootprintType

The *FootprintType* primitive definition and schema layout is shown in Figure 4-2. It is constructed of four vertices describing the footprint. The vertices are built using *Lat* and *Lon* parameters, which describe the latitude and longitude of each of the vertices and are measured in decimal degrees. The first vertex lists the upper left corner of the footprint and increasing vertices are populated from the footprint corners in a clockwise direction.

For example, *GeographicCoverageType's Footprint* uses the *FootprintType.*

```
<Footprint size="4">
        <Vertex index="1">
                <Lat>3.809545314288736e+01</Lat>
                <Lon>-1.226389570643719e+02</Lon>
        </Vertex>
        <Vertex index="2">
                <Lat>3.809545314288736e+01</Lat>
                <Lon>-1.126389570643719e+02</Lon>
        </Vertex>
        <Vertex index="3">
                <Lat>3.909545314288736e+01</Lat>
                <Lon>-1.126389570643719e+02</Lon>
        </Vertex>
        <Vertex index="4">
                <Lat>3.909545314288736e+01</Lat>
                <Lon>-1.226389570643719e+02</Lon>
        </Vertex>
</Footprint>
```

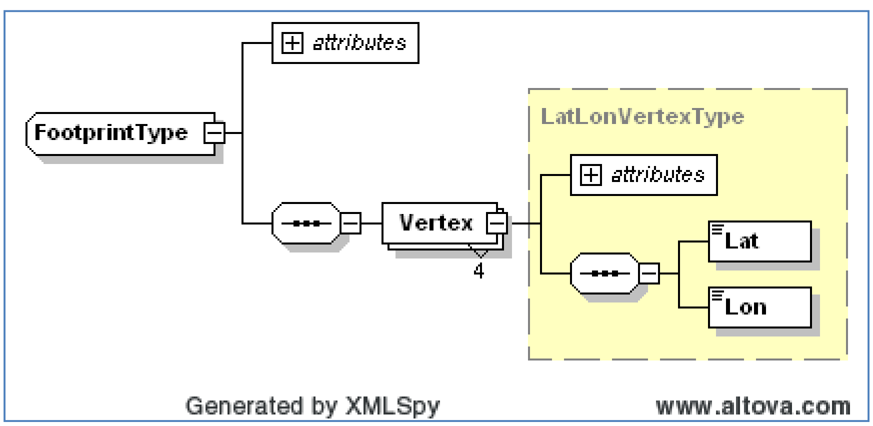


Figure 4-2 Primitive Definition & Schema Layout: *FootprintType*

4.1.4.2 *Poly1DType*

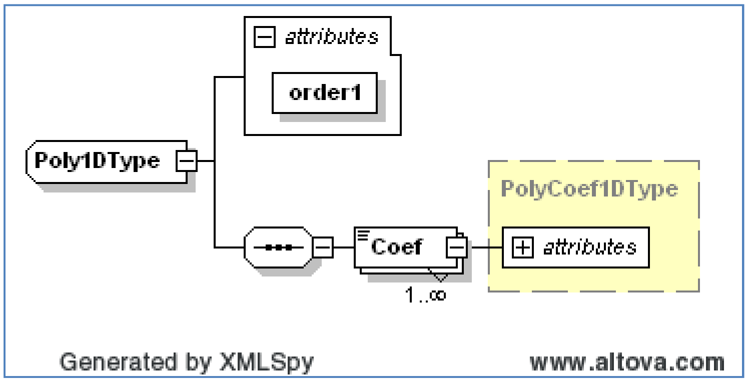


Figure 4-3 Primitive Definition & Schema Layout: *Poly1DType*

The *Poly1DType* is shown in Figure 4-3. It contains an expandable structure for a 1-D polynomial where the order of the polynomial, *M*, is defined by the attribute *order1* along with a number of coefficients based on the order (up to *M+1*). Only non-zero entries of the polynomial need to be specified as parameters *exponent1*.

$$F(x) = \sum_{m=0}^{M} c_m x^m$$

For example, a polynomial named *testpoly* representing the equation

$$x^2 + 3$$

would be represented as:

```
<testpoly order1="2">
        <Coef exponent1="2">1</Coef>
        <Coef exponent1="0">3</Coef>
</testpoly>
```

4.1.4.3 *Poly2DType*

The *Poly2DType* is shown in

Figure 4-4. The attributes *order1* and *order2* specify the order of the two-dimensional polynomial and are defined as M and N, respectively. The parameters *exponent1* and *exponent2* define the exponents for a given *Coef*. The total number of possible coefficients is (M+1)*(N+1), where only non-zero entries of the polynomial need to be specified. This complex primitive type represents an equation of the following form.

$$F(x, y) = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{m,n} x^m y^n$$

For example, a polynomial named *testpoly*, representing the equation

$$x^2 \cdot y^6 + 3y^2 + 4x + 5$$

would be represented as

```
<testpoly order1="2" order2="6">
        <Coef exponent1="2" exponent2="6">1</Coef>
        <Coef exponent1="0" exponent2="2">3</Coef>
        <Coef exponent1="1" exponent2="0">4</Coef>
        <Coef exponent1="0" exponent2="0">5</Coef>
</testpoly>
```

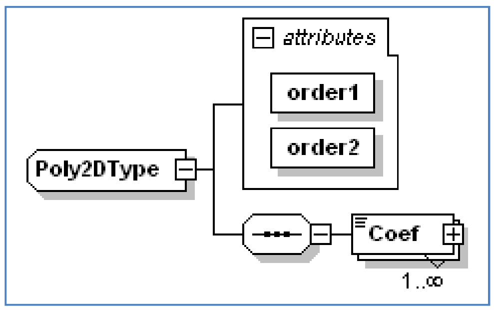


Figure 4-4 Primitive Definition & Schema Layout: *Poly2DType*

4.1.4.4 *XYZPolyType*

The *XYZPolyType* is shown in Figure 4-5. It contains X, Y, and Z parameter types where each parameter in the complex structure contains a single *Poly1DType* complex primitive. For implementation details please see *Poly1DType*.

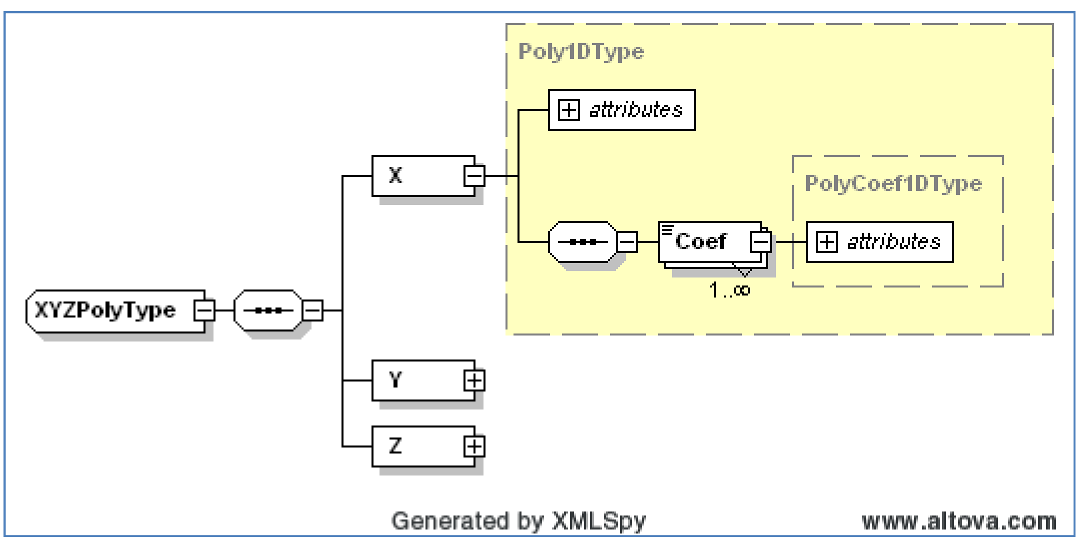


Figure 4-5 Primitive Definition & Schema Layout: *XYZPolyType*

4.1.4.5 LookupTableType & Lookup3TableType

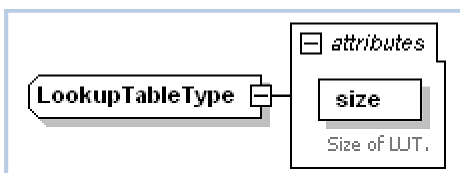
The *LookupTableType* is a complex primitive, shown in Figure 4-6, and it is used to provide a mapping of values from one input space to another. In the SIDD standard, it is only used to map between integer representations (see Section 4.2.2). The *LUT* element is a list object which represents the look-up table. The list should be indexed using the input value and the corresponding value at that location in the output. The *size* attribute is used to represent the length of the *LUT* list object. This is required to be consistent with the size of the *LUT*. This is an xs:list object which delimits the values in the list using whitespace.

An example of this is the *RemapLUT* within the *MonochromeDisplayRemapType*.

```
<MonochromeDisplayRemapType>
        <RemapLUT size="256">0 2 5 7 10…</RemapLUT>
</MonochromeDisplayRemapType>
```

The Lookup3TableType is analogous to the LookupTableType described above with the exception that each entry in the xs:list is a comma-separated triplet. An example of this is the RemapLUT within the ColorDisplayRemapType used for 8-bit indexed color products.

```
<ColorDisplayRemapType>
        <RemapLUT size="256">0,0,0 0,0,1 0,0,2…</RemapLUT>
</ColorDisplayRemapType>
```



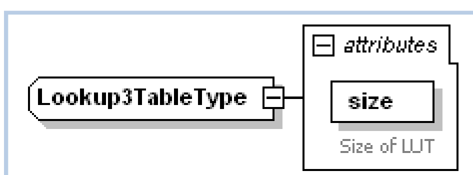


Figure 4-6 Primitive Definition & Schema Layout: *LookupTableType* & *Lookup3TableType*

4.1.4.6 ReferencePointType

The *ReferencePointType* is shown in Figure 4-7. The *XYZType* and *RowColDoubleType* are complex primitive types within the *ReferencePointType*, and are referenced in Table 4-2. The *ECEF* parameter is contained in the *ReferencePointType* and is an *XYZType*. The ECEF parameter metadata is a *XYZType* and is populated by the *XYZType* instructions (Table 4-2). In addition, the *Point* parameter is a *RowColDoubleType*; please refer to *RowColDoubleType* for population instructions (Table 4-2).

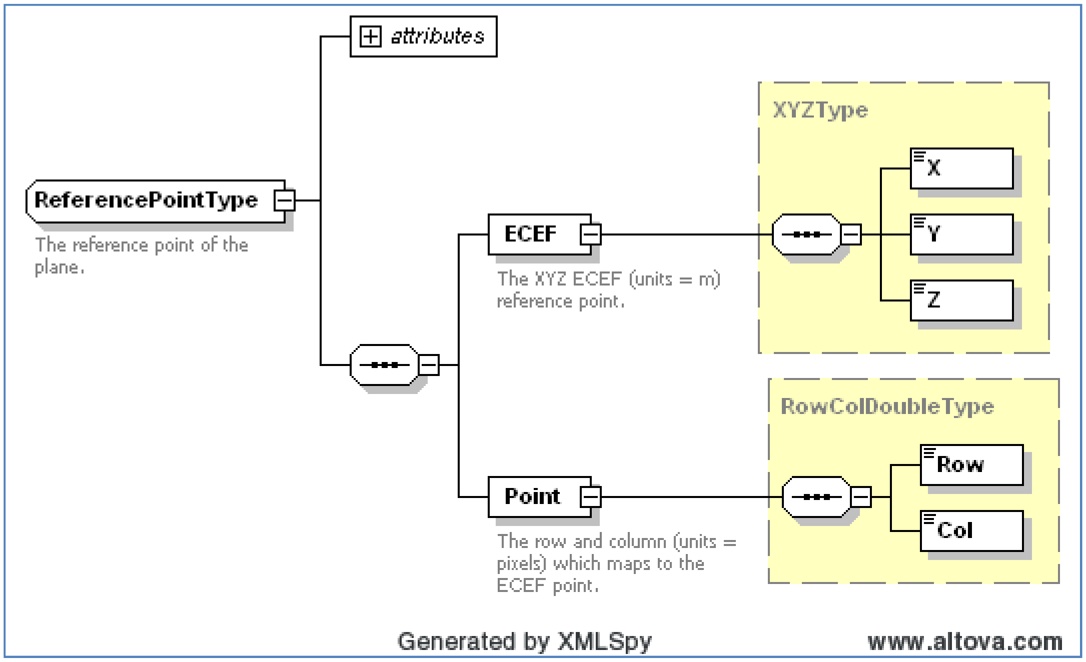


Figure 4-7 Primitive Definition & Schema Layout: *ReferencePointType*

4.2 XML Metadata Parameter List

The SIDD XML data is arranged to have related parameters grouped together in complex structures. Each complex structure has branches to either other complex structures or parameters. The complex structure stops branching once a parameter has been reached. The parameter is made up of primitives, which declares the data type of the parameter. In addition, the parameter contains the actual metadata associated with the image product.

The section is broken into the top level groupings displayed in Figure 4-8. The top level groupings are complex structures but will be referred to as groupings to distinguish from lower level complex structures. In each grouping section, the branching layout for each complex structure is displayed. The complex structural layouts start with the highest point in the branch and end with the lowest. The parameters' primitives and definitions are then listed in tables. In addition, a reference is provided to the reusable primitives defined in Sections 4.1.3 and 4.1.4.

Figure 4-8 displays the top level groupings for the SIDD XML. The definitions for each grouping are listed in the Figure 4-8 and Table 4-4.

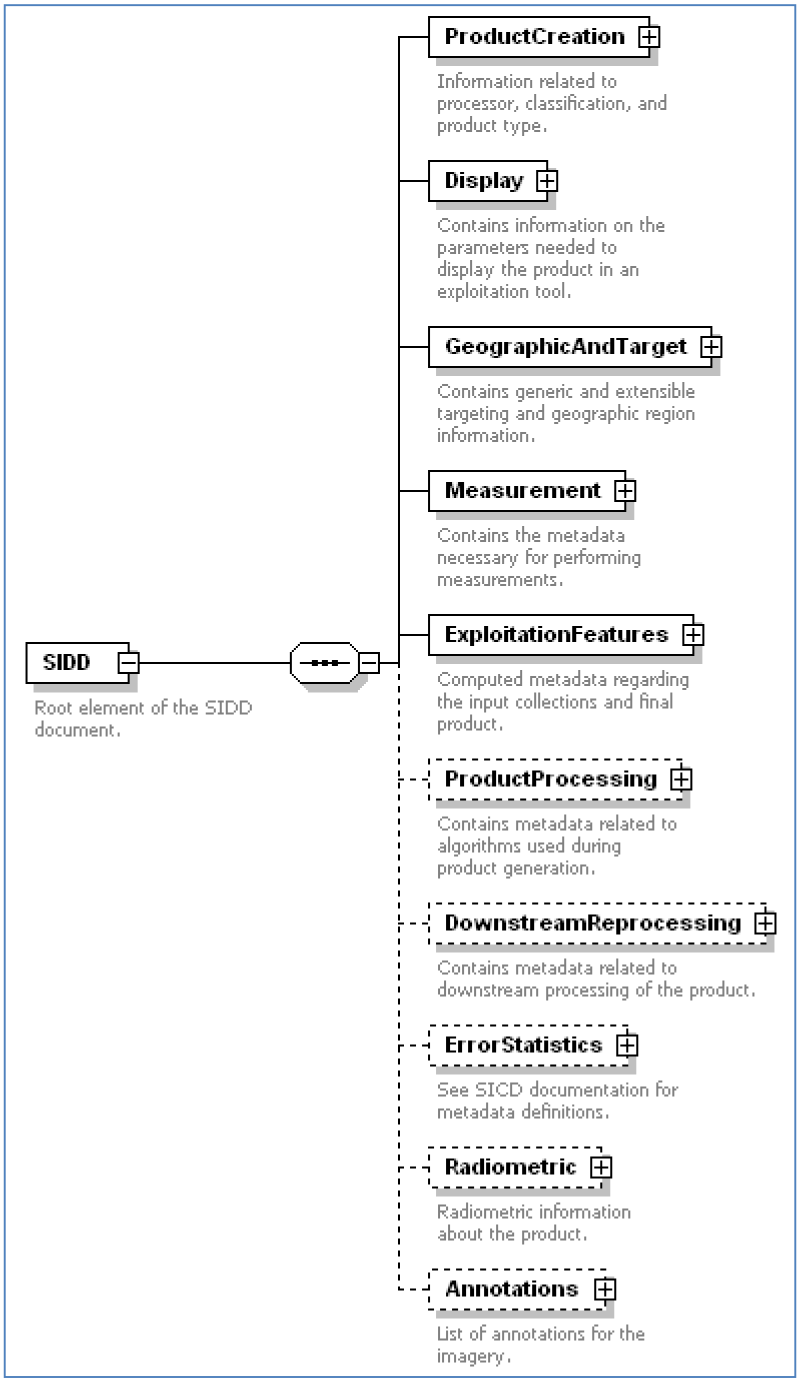


Figure 4-8 *SIDD* Schema Layout

Table 4-4 SIDD Schema Layout Paragraph Reference		
Grouping	**Section**	**Definition**
ProductCreation	4.2.1	Provides information related to the initial processing of the product including classification, product type, and processor that produced it. This should be populated upon product creation.
Display	4.2.2	Contains information needed to help properly display the product in an exploitation tool. This should be populated upon product creation.
GeographicAndTarget	4.2.3	Contains generic and extensible targeting and geographic region information. This should be populated upon product creation.
Measurement	4.2.4	Contains the metadata detailing the projection applied as well as collection metadata necessary for performing measurements. This should be populated upon product creation.
ExploitationFeatures	4.2.5	Contains information that provides aid to an end user in interpreting product phenomenology with regard to the collections. The metadata in this grouping can also be used to generate legends and icons. This should be populated upon product creation.
ProductProcessing	4.2.6	Contains an extensible structure for recording processor-specific algorithm information applied during product generation. This should be populated upon product creation.
DownstreamReprocessing	4.2.7	The metadata describes the downstream exploitation modifications made to the file such as geometric chipping, resampling, etc. This metadata should be populated by the downstream tool making the modifications, such as an Electronic Light Table (ELT). This should be populated upon product creation.
ErrorStatistics	4.2.8	Contains metadata that describes the errors in radar collection parameters, and that is required for propagation of error ellipses to the product. See SICD documentation (Table 1-2) for metadata definitions. Only one set of error data is included in the SIDD product that reflects the overall product. This should be populated upon product creation.
Radiometric	4.2.9	Contains radiometric information about the product (parameters that enable the conversion of pixel power level to radar

		reflectivity parameters). See SICD documentation (Table 1-2) for metadata definitions. Only one set of radiometric data is included in the SIDD product that reflects the overall product. This should be populated upon product creation.
Annotations	4.2.10	List of annotations for the imagery.

4.2.1 Product Creation

The *ProductCreation* contains information related to initial processing, classification, and product type. The *ProductCreation* complex structures are laid out in Figure 4-9 through Figure 4-10. The definitions are defined in Table 4-5 through Table 4-6.

4.2.1.1 Overview & Parameter Definition: *ProductCreation*

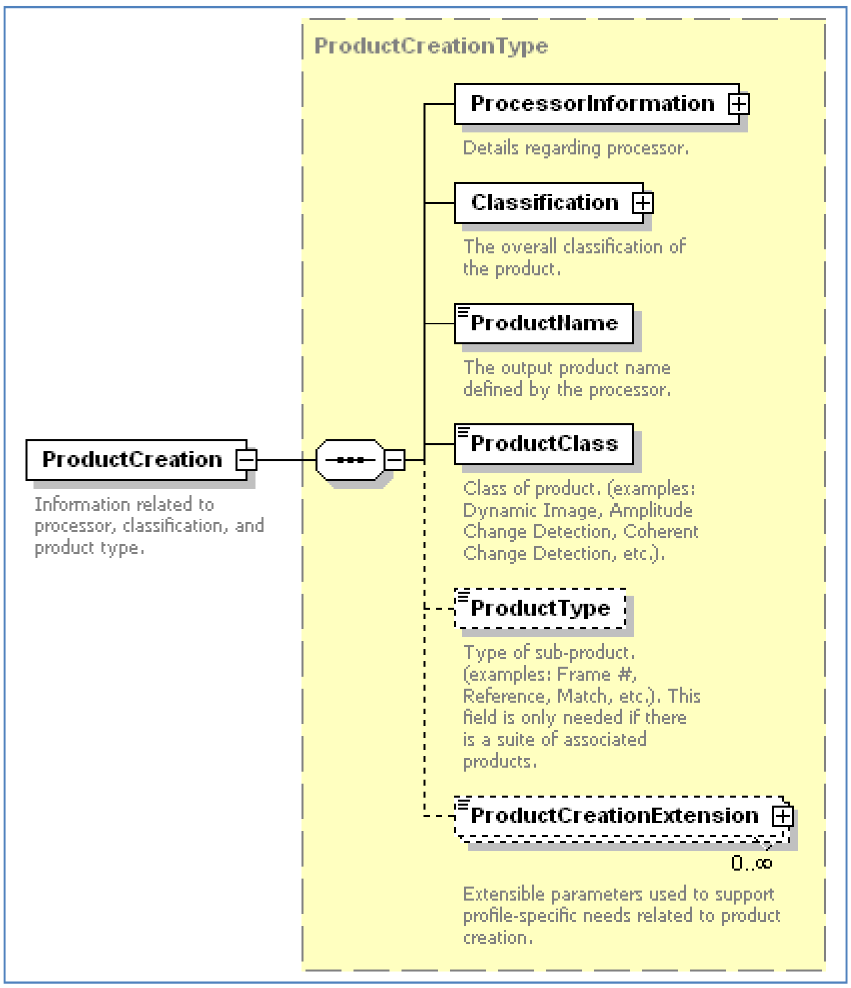


Figure 4-9 *ProductCreation* Parameter and Complex Structures

Table 4-5 *ProductCreation* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ProcessorInformation	ProcessorInformationType	Table 4-6 Figure 4-10	Required parameter containing basic information about the processor used to create the product.
Classification	ProductClassificationType	Table 4-6 Figure 4-11	Required parameter containing the overall classification of the product.
ProductName	String	Table 1-3 Reference 2	Required parameter containing the output product name defined by the processor.
ProductClass	String	Table 1-3 Reference 2	Required parameter containing the class of product. Examples: Dynamic Image, Amplitude Change Detection, Coherent Change Detection, etc.)
ProductType	String	Table 1-3 Reference 2	Optional parameter containing information on the type of sub-product. This field is only needed if there is a suite of associated products. Examples: Frame #, Reference, Match, etc.
ProductCreationExtension	ParameterType	Table 4-3	Optional extensible parameters used to support profile-specific needs related to product creation

4.2.1.2 Overview and Parameter Definitions: *ProcessorInformation*

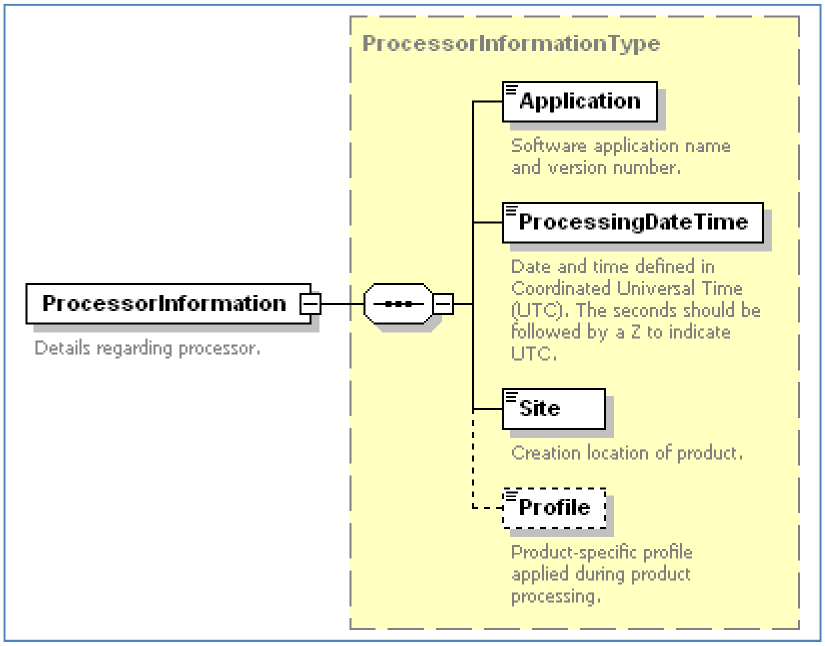


Figure 4-10 *ProcessorInformation* Structure

Table 4-6 *ProfessorInformation* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Application	String	Table 1-3 Reference 2	Required parameter that gives the name and version of the application used to create the product.
ProcessingDateTime	DateTime	Table 1-3 Reference 2	Required parameter that contains the date and time defined in Coordinated Universal Time (UTC). The seconds should be followed by a Z to indicate UTC.
Site	String	Table 1-3 Reference 2	Required parameter that specifies the creation location of the product.
Profile	String	Table 1-3 Reference 2	Optional parameter containing information about a product-specific profile applied during product processing.

4.2.1.3 Overview and Parameter Definition: *ProductClassification*

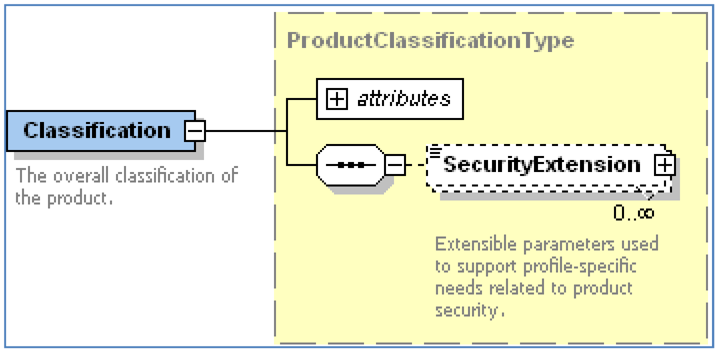


Figure 4-11 *ProductClassification* Structure

Table 4-7 *ProductClassification* Parameter Definitions			
Attribute	**Type**	**Reference**	**Definition**
attributes	ISMRootNodeAttributeGroup ResourceNodeAttributeGroup	Table 1-3	Security attributes. Please see associated reference and specification for further details.
Parameter	**Type**	**Reference**	**Definition**
SecurityExtension	ParameterType	Table 4-3	Optional extensible parameters for profile-specific security markings.

4.2.2 Display

The *Display* grouping contains information required for proper display of the imagery. The parameters in this block are expected to be utilized in conjunction with a NGA Softcopy Image Processing Standard (SIPS) v2.1 compliant viewer. In addition, the grouping also describes any remaps or monitor compensations applied to the data, as well as, differentiating whether a color remap or monochrome remap was applied to the data.

4.2.2.1 Parameters and Complex Structures: *Display*

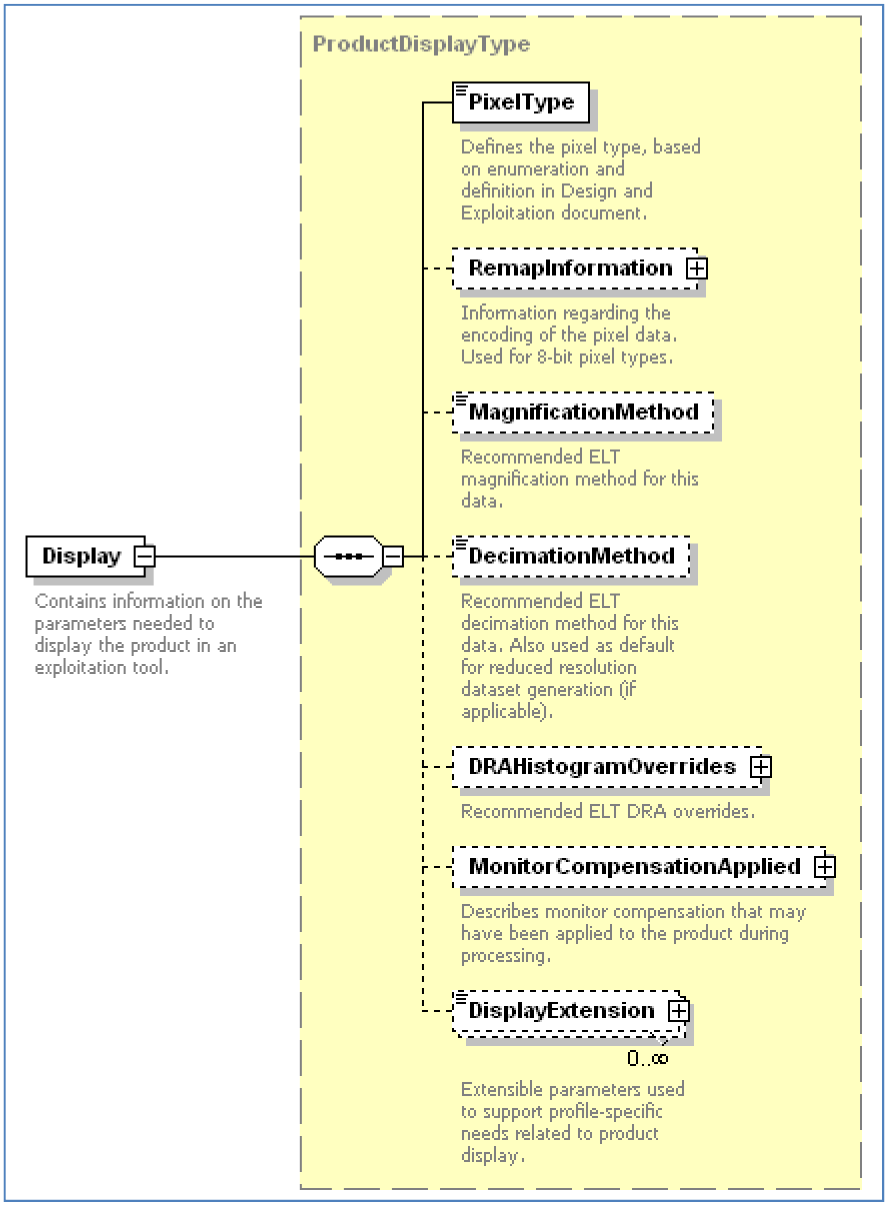


Figure 4-12 ***ProductDisplay*** **Parameter and Complex Structures**

Table 4-8 *ProductDisplay* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
PixelType	PixelType	Table 4-9	Required parameter defining the pixel type, based on enumeration defined in Table 4-9.
RemapInformation	RemapChoiceType	Figure 4-13 Table 4-12	Optional parameter containing information regarding the encoding of the pixel data. Used for 8-bit pixel types.
MagnificationMethod	MagnificationMethodType	Table 4-10	Optional parameter specifying the recommended ELT magnification method for this data. The parameter is an enumeration defined by Table 4-10.
DecimationMethod	DecimationMethodType	Table 4-11	Optional parameter specifying the recommended ELT decimation method for this data. It should also be used as default for reduced resolution dataset generation (if applicable). The parameter is an enumeration defined by Table 4-11.
DRAHistogramOverrides	DRAHistogramOverridesType	Figure 4-16 Table 4-15	Optional parameter specifying the recommended ELT DRA overrides.
MonitorCompensationApplied	MonitorCompensationAppliedType	Table 4-16 Figure 4-17	Optional parameter describing monitor compensation that may have been applied to the product during processing.
DisplayExtension	ParameterType	Table 4-3	Optional extensible parameter used to support profile-specific needs related to product display.

Table 4-9 Enumeration Types for *PixelType*	
Enumeration Type	**Definition**
MONO8I	Monochrome 8-bit image
MONO8LU	Monochrome 8-bit image with a LUT
MONO16I	Monochrome 16-bit image
RGB8LU	RGB 8-bit image with a color LUT

RGB24I	RGB 24-bit color image

Table 4-10 Enumeration Types for *MagnificationMethodType*

Enumeration Type	**Definition**
NEAREST_NEIGHBOR	Magnification through nearest neighbor has historically been referred to as replication. Under this technique any coordinate which falls within the pixel boundaries obtains the same values as that pixel. The magnification factor is d in both x and y. d>0 $$g(x,y) = f(round\left(\frac{x}{d}\right), round(\frac{y}{d}))$$
BILINEAR	Bilinear interpolation for a magnification factor of d in both the x and y direction is accomplished with the equations below. $$x_1 = floor\left(\frac{x}{d}\right)$$ $$x_2 = ceiling\left(\frac{x}{d}\right)$$ $$y_1 = floor\left(\frac{y}{d}\right)$$ $$y_2 = ceiling\left(\frac{y}{d}\right)$$ $$g(x,y) = \left[\left(x_2 - \frac{x}{d}\right) \quad \left(\frac{x}{d} - x_1\right)\right]\begin{bmatrix} f(x_1,y_1) & f(x_1,y_2) \\ f(x_2,y_1) & f(x_2,y_2) \end{bmatrix}\begin{bmatrix} \left(y_2 - \frac{y}{d}\right) \\ \left(\frac{y}{d} - y_1\right) \end{bmatrix}$$
LAGRANGE	Lagrange decimation is the standard interpolation method specified in the SIPS documentation. The Lagrange interpolation uses a 4x4 region about the pixel of interest when determining the new value. Review Section 2.3.5 of the SIPS document.

Table 4-11 Enumeration Types for *DecimationMethodType*	
Enumeration Type	**Definition**
NEAREST_NEIGHBOR	Decimation by a factor d in both the x and y directions by the nearest neighbor method requires that the value at the output is computed by rounding the new pixel location (x,y) to the nearest integer pixel that was in the original image. $$g(x,y) = f(round(dx), round(dy))$$
BILINEAR	Bilinear interpolation for a decimation factor of d in both the x and a y direction is accomplished with the equations below. $$x_1 = floor(dx)$$ $$x_2 = ceiling(dx)$$ $$y_1 = floor(dy)$$ $$y_2 = ceiling(dy)$$ $$g(x,y) = [(x_2 - dx)\ \ (dx - x_1)] \begin{bmatrix} f(x_1,y_1) & f(x_1,y_2) \\ f(x_2,y_1) & f(x_2,y_2) \end{bmatrix} \begin{bmatrix} (y_2 - dy) \\ (dy - y_1) \end{bmatrix}$$

BRIGHTEST_PIXEL	For brightest pixel decimation, the brightest pixel in the region near the new pixel is chosen. The decimation factor in x and y is d. The strategy shown is for MONO type data. The brightest pixel strategy used for RGB24I and RGB8LU is addressed in the SIPS document.
	$$g(x,y) = -\infty$$
	$$x_1 = floor\left(d\left(x - \frac{1}{2}\right)\right)$$
	$$x_2 = ceiling\left(d\left(x + \frac{1}{2}\right)\right)$$
	$$y_1 = floor\left(d\left(y - \frac{1}{2}\right)\right)$$
	$$y_2 = ceiling\left(d\left(y + \frac{1}{2}\right)\right)$$
	For(k = x_1:1:x_2)
	    For(m = y_1:1:y_2)
	        $g(x,y) = max\big(g(x,y), f(k,m)\big)$
	    End
	End

LAGRANGE	Lagrange decimation is the standard interpolation method specified in the SIPS documentation. The Lagrange interpolation uses a 4x4 region about the pixel of interest when determining the new value. Review Section 2.3.5 of the SIPS document.

4.2.2.2 Overview and Parameter Definitions*: RemapInformation*

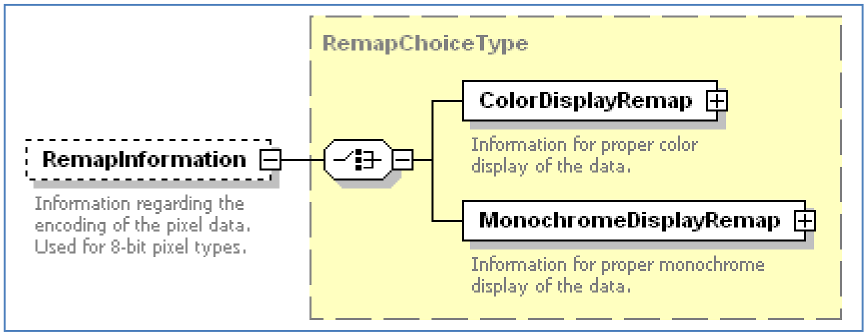


Figure 4-13 *RemapInformation* Structure

Table 4-12 *RemapInformation* Parameters Definitions

Parameter	Type	Reference	Definition
ColorDisplayRemap	ColorDisplayRemapType	Table 4-13 Figure 4-14	Mutually exclusive required parameter containing information for proper color display of the data. *RemapChoiceType* will be either populated with the parameter *ColorDisplayRemapType* or *MonochromeDisplayRemapType*
MonochromeDisplayRemap	MonochromeDisplayRemapType	Figure 4-15 Table 4-14	Mutually exclusive required parameter containing information for proper monochrome display of the data. *RemapChoiceType* will be either populated with the parameter *ColorDisplayRemapType* or *MonochromeDisplayRemapType*

4.2.2.3 Overview and Parameter Definitions*: ColorDisplayRemap*

The color remap description (*ColorDisplayRemapType*) contains a set of optional lookup tables for red, green, and blue. These tables are only used for 8-bit indexed color products. The look-up table maps the 8-bit index to a 24-bit multi-channel pixel space. This should operate as described in the “Product Generation Option” step of the SIPS document. Note that 24-bit true multi-channel color products should not have any transform applied to them. The complex structure’s layouts within the *Display* grouping are shown in Figure 4-14. The complex structure’s parameters and primitives are defined in Table 4-13.

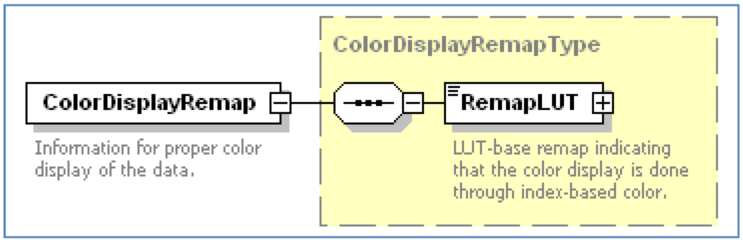


Figure 4-14 *ColorDisplayRemap* Structure

Table 4-13 *ColorDisplayRemap* Parameter Definition			
Parameter	**Type**	**Reference**	**Definition**
RemapLUT	Lookup3TableType	Figure 4-6	Required LUT-base remap indicating the color display is done through indexed-based color.

4.2.2.4 Overview and Parameter Definitions: MonochromeDisplayRemap

The monochrome remap description (*MonochromeDisplayRemapType*) describes two major items. First, a textual and parameter definition of the remap is provided through the *RemapType* and optional *RemapParameter* fields. These fields are for information purposes only, as the actual un-mapping of the data to density space is accomplished via look-up table. Second, the *RemapLUT* block provides a remap from 8-bit space to log amplitude space. This un-mapping of the data from 8-bit space corresponds to the “Product Generation Option” step of the SIPS documentation. By providing this remap as a generalized look-up table, the block is allowing multiple remap definitions to be represented without need for a change in either the SIPS or the SIDD documentation. For 16-bit data, the *RemapLUT* should not be utilized.

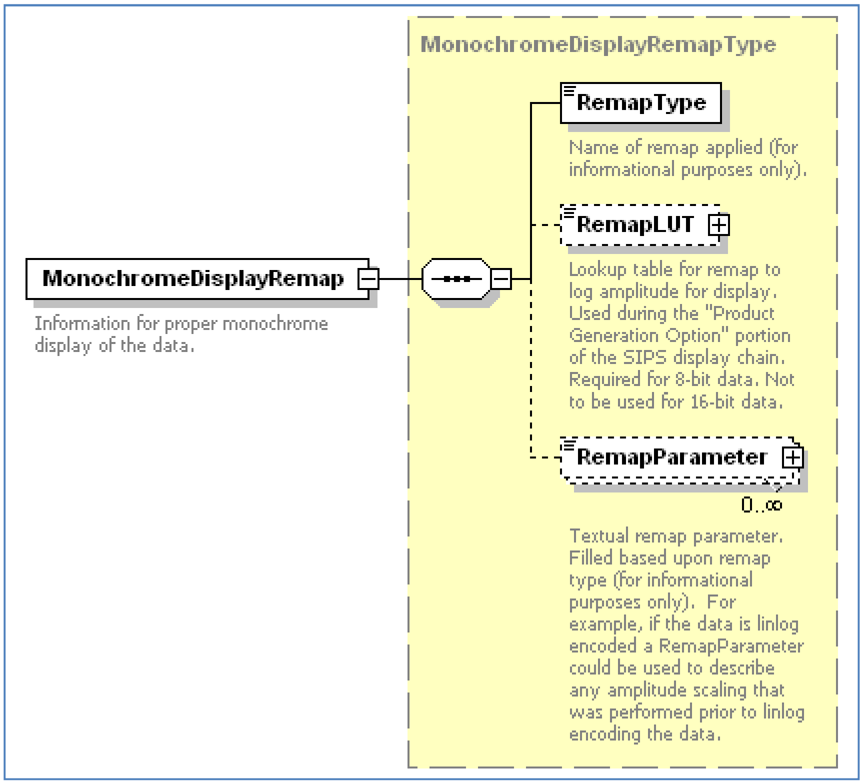


Figure 4-15 *MonochromeDisplayRemap* Structure

Table 4-14 *MonochromeDisplayRemap* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
RemapType	String	Table 1-3 Reference 2	Required parameter that contains the name of remap applied (for informational purposes only).
RemapLUT	LookupTableType	Figure 4-6	Optional parameter that contains the lookup table for remap to log amplitude for display. Used during the “Product Generation Option” portion of the SIPS display chain. Should be utilized for 8-bit data.
RemapParameter	ParameterType	Table 4-3	Optional textual remap parameter. Filled based upon remap type (for informational purposes only). For example, if the data is linlog encoded a *RemapParameter* could be used to describe any amplitude scaling that was performed prior to linlog encoding the data.

4.2.2.5 Overview and Parameter Definitions: *DRAHistogramOverrides*

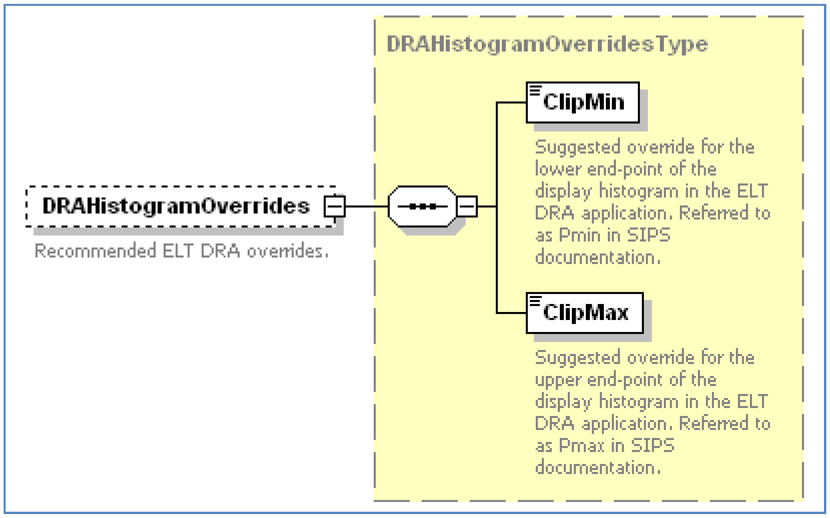


Figure 4-16 *DRAHistogramOverrides* Structure

Table 4-15 *DRAHistogramOverrides* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ClipMin	Int	Table 1-3 Reference 2	Required parameter describing the suggested override for the lower end-point of the display histogram in the ELT DRA application. Referred to as P_{min} in SIPS documentation.
ClipMax	Int	Table 1-3 Reference 2	Required parameter describing the suggested override for the upper end-point of the display histogram in the ELT DRA application. Referred to as P_{max} in SIPS documentation.

4.2.2.6 Overview and Parameter Definitions: *MonitorCompensationApplied*

The *MonitorCompensationApplied* grouping describes monitor compensation applied to the pixel data. The parameters of the monitor compensation (*Gamma* and *XMin*) are provided for reference. If such compensation is applied, it is recommended that any softcopy display chain skip DRA application. These parameters are shown in Figure 4-17.

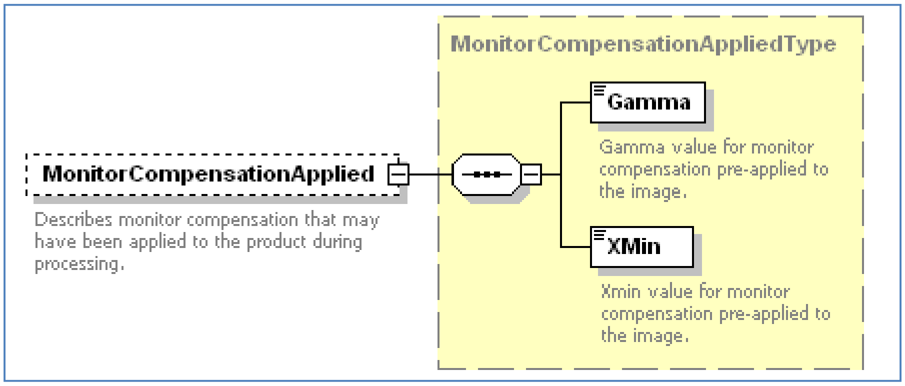


Figure 4-17 *MonitorCompensationApplied* Structure

Table 4-16 *MonitorCompensationApplied* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Gamma	*Double*	Table 1-3 Reference 2	The value of this parameter provides the gamma or xmin value of the nominal monitor compensation applied to the product prior to dissemination. Nominal monitor compensation is applied to some products to ensure an acceptable visual display of the products across display tools. In order to be displayed on a computer monitor, the pixel values are first converted to density (D) space and are then converted to "transmittance". The formula for density to transmittance conversion is a follows: (see equation below) where Dclip = 255, xmax = 255, and Rt = 50.01. In viewing standard detected gray-scale images, y (gamma) and xmin are normally determined by display. Optimal display of products, however, may require default display parameters be used. This field must be used to express the default display parameters. $$x(D) = (x_{max} - x_{min})\left(\frac{R_t^{\left(\frac{min(D,D_{clip})}{D_{clip}}\right)} - 1}{R_t - 1}\right)^{\frac{1}{r}} + x_{min}$$
XMin	*Double*	Table 1-3 Reference 2	The value of this parameter provides the gamma or xmin value of the nominal monitor compensation applied to the product prior to dissemination. Nominal monitor compensation is applied to some products to ensure an acceptable visual display of the products across display tools. In order to be displayed on a computer monitor, the pixel values are first

Table 4-16 *MonitorCompensationApplied* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
			converted to density space and are then converted to “transmittance”. The formula for density to transmittance conversion is a follows: (see equation below) where Dclip = 255, xmax = 255, and Rt = 50.01. In viewing standard detected gray-scale images, y (gamma) and xmin are normally determined by display. Optimal display of products, however, may require default display parameters be used. This field must be used to express the default display parameters. $$x(D) = (x_{max} - x_{min})\left(\frac{R_t^{\left(\frac{min(D,D_{clip})}{D_{clip}}\right)} - 1}{R_t - 1}\right)^{\frac{1}{r}} + x_{min}$$

4.2.3 *GeographicAndTarget*

The *GeographicAndTarget* grouping contains information about the targets residing in the product and the geographic coverage of the product. The structural layouts of the groupings are shown in Figure 4-18 through Figure 4-20. The grouping’s parameter definitions are defined in Table 4-17 through Table 4-20.

4.2.3.1 Parameters and Complex Structures: *GeographicAndTarget*

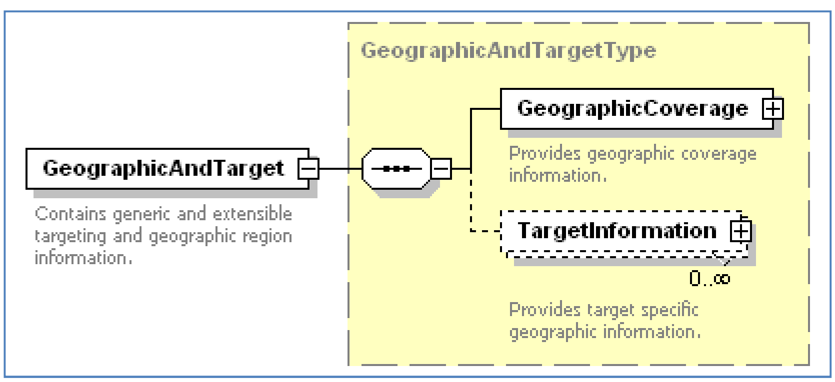


Figure 4-18 *GeographicAndTarget* Parameter and Complex Structures

Table 4-17 *GeographicAndTarget* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
GeographicCoverage	*GeographicCoverageType*	Table 4-18 Figure 4-19	Required parameter containing information for the ground coverage of the image.
TargetInformation	*TargetInformationType*	Table 4-20 Figure 4-21	Optional parameter repeated multiple times for each target in the product footprint. Provides target specific geographic information.

4.2.3.2 Overview and Parameter Definition: *GeographicCoverage*

The *GeographicCoverage* complex structure represents a hierarchical decomposition of the area contained within the product. It contains an optional *GeoregionIdentifier* for the area, a geodetic *Footprint*, and either information pertaining to the region or a decomposition of the region into subregions. Each *SubRegion* contains the same information as the *GeographicCoverageType* element and may be decomposed in a similar manner. Using this sub-region decomposition may be useful to relate specific security or country identifiers to particular portions of the product's ground coverage.

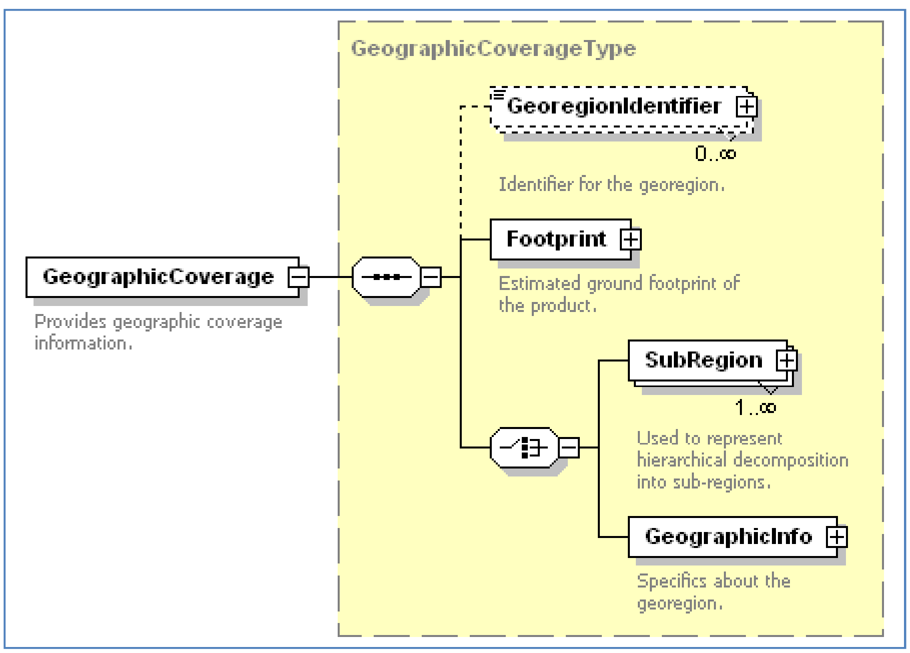


Figure 4-19 *GeographicCoverage* Structure

Table 4-18 *GeographicCoverage* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
GeoregionIdentifier	ParameterType	Table 4-3	Optional parameter used to identify the geo-region.
Footprint	FootprintType	Figure 4-2	Required parameter containing the estimated ground footprint of the product.
Subregion	GeographicCoverageType	Figure 4-19 Table 4-18	Choice required parameter used to represent hierarchical decomposition into sub-regions.
GeographicInfo	GeographicInformationType	Figure 4-20 Table 4-19	Choice required parameter specifying specifics about the geo-regions.

4.2.3.3 Overview & Parameter Definition: *GeographicInfo*

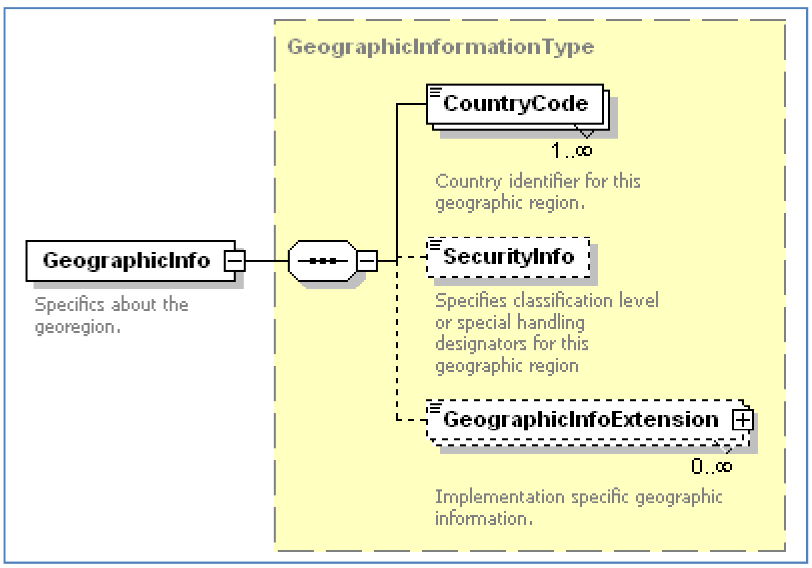


Figure 4-20 *GeographicInformation* Structure

Table 4-19 *GeographicInformation* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
CountryCode	String	Table 1-3 Reference 2	Optional parameter used for the country identifier for this geographic region.
SecurityInfo	String	Table 1-3 Reference 2	Optional parameter that specifies the classification level or special handling designator for this geographic region.
GeographicInfoExtension	ParameterType	Table 4-3	Optional parameter for implementation specific geographic information.

4.2.3.4 *TargetInformation* Parameter and Complex Structures

The *TargetInformation* complex structure contains one or more *Identifier* elements. These are system dependent strings used to describe the target. As a *ParameterType* object, each *Identifier* will contain a *name* attribute used to describe the identifier type (i.e. BE number, order of battle). These names are highly system dependent. For each target, an optional *Footprint* can be specified which represents the ground coverage of the target. The *TargetInformationExtension* parameter is available for ancillary data regarding the target.

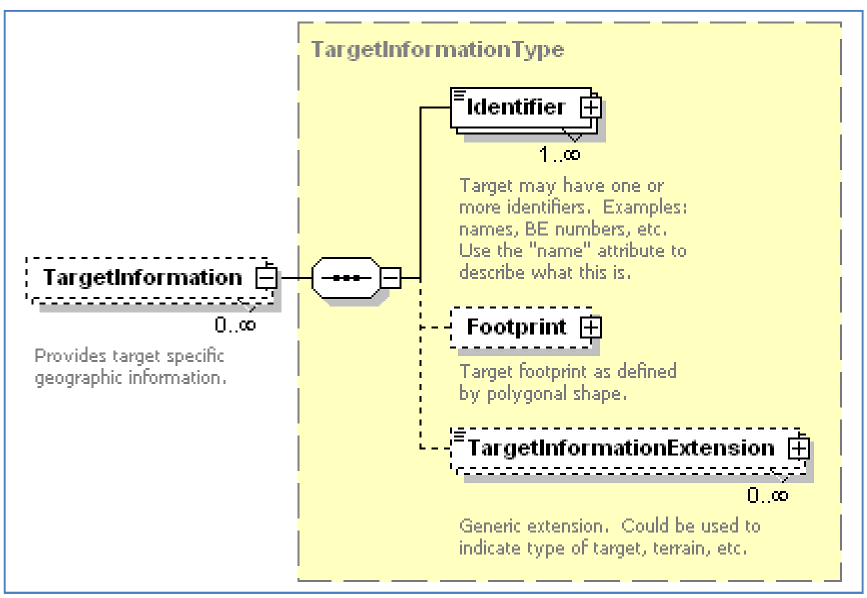


Figure 4-21 *TargetInformation* Structure

Table 4-20 *TargetInformation* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Identifier	ParameterType	Table 4-3	Required target identifier. Examples: names, BE numbers, etc. Use the "name" attribute to populate the description identifier.
Footprint	FootprintType	Figure 4-2	Optional Footprint parameter specifies the ground coverage of the target
TargetInformationExtension	ParameterType	Table 4-3	Optional generic extension. Could be used to indicate type of target, terrain, etc.

4.2.4 Measurement

The *Measurement* grouping encapsulates metadata that is necessary for performing image to geographic measurements. The complex structure layouts are shown in Figure 4-22 through Figure 4-26. The groupings definitions are listed in Table 4-21 through Table 4-25.

4.2.4.1 Overview & Parameter Definition: *Measurement*

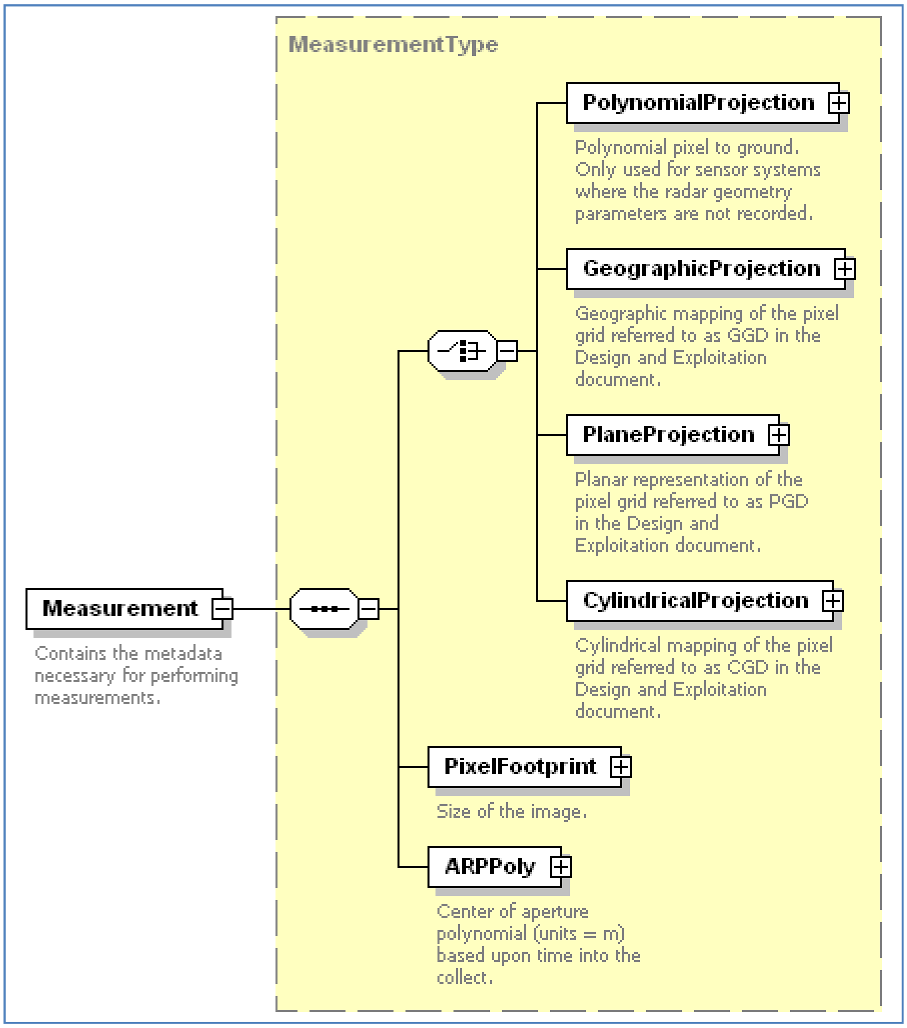


Figure 4-22 *Measurement* Parameter and Complex Structures

Table 4-21 *Measurement* Definitions			
Parameter	**Type**	**Reference**	**Definition**
PolynomialProjection	PolynomialProjectionType	Figure 4-23 Table 4-23	Polynomial representation which converts pixel to ground location. This parameter is only used for sensor systems where the scalar geometry parameters are not recorded.
PlaneProjection	PlaneProjectionType	Table 4-23 Figure 4-24	Planar representation of the pixel grid referred to as PGD in the Section 2.5.
GeographicProjection	GeographicProjectionType	Table 4-25 Figure 4-26	Geographic mapping of the pixel grid referred to as GGD in Section 2.7.
CylindricalProjection	CylindricalProjectionType	Figure 4-27 Table 4-26	Cylindrical mapping of the pixel grid referred to as CGD in Section 2.8.
PixelFootprint	RowColIntType	Table 4-3	Required parameter which gives the size of the image in terms of pixels.
ARPPoly	XYZPolyType	Figure 4-5	Aperture Reference Point (ARP) polynomial in ECF as a function of time t (variable 1). Time t = 0 at collection start. The ARPPoly is a required parameter.

4.2.4.2 Overview & Parameter Definition: *PolynomialProjection*

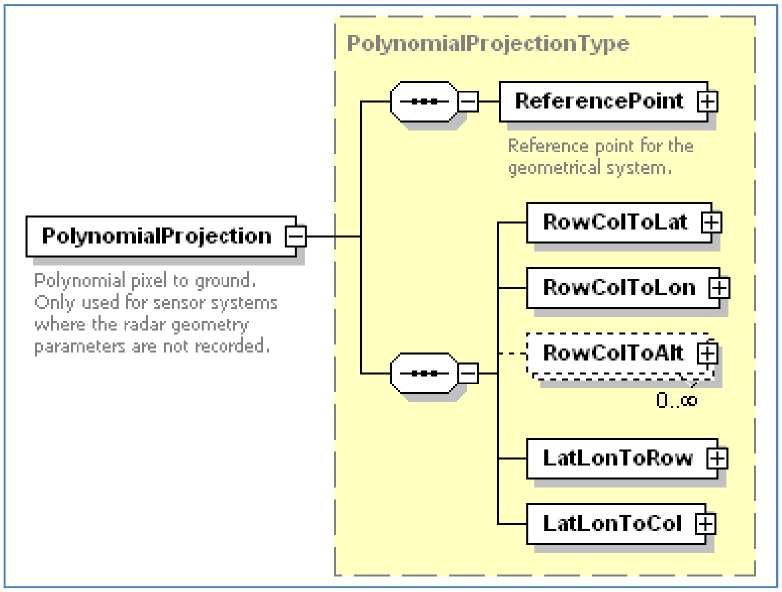


Figure 4-23 *PolynomialProjection* Structure

Table 4-22 PolynomialProjection Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ReferencePoint	ReferencePointType	Figure 4-7	Required parameter describing the reference point of the plane.
RowColToLat	Poly2DType	Figure 4-4	Polynomial used to convert from row and column pixel location to latitude.
RowColToLon	Poly2DType	Figure 4-4	Polynomial used to convert from row and column pixel location to longitude.
RowColToAlt	Poly2DType	Figure 4-4	Polynomial used to convert from row and column pixel location to altitude.
LatLonToRow	Poly2DType	Figure 4-4	Polynomial used to convert latitude and longitude locations into row pixel locations.
LatLonToCol	Poly2DType	Figure 4-4	Polynomial used to convert latitude and longitude locations into column pixel locations.

4.2.4.3 Overview and Parameter Definition: *PlaneProjection*

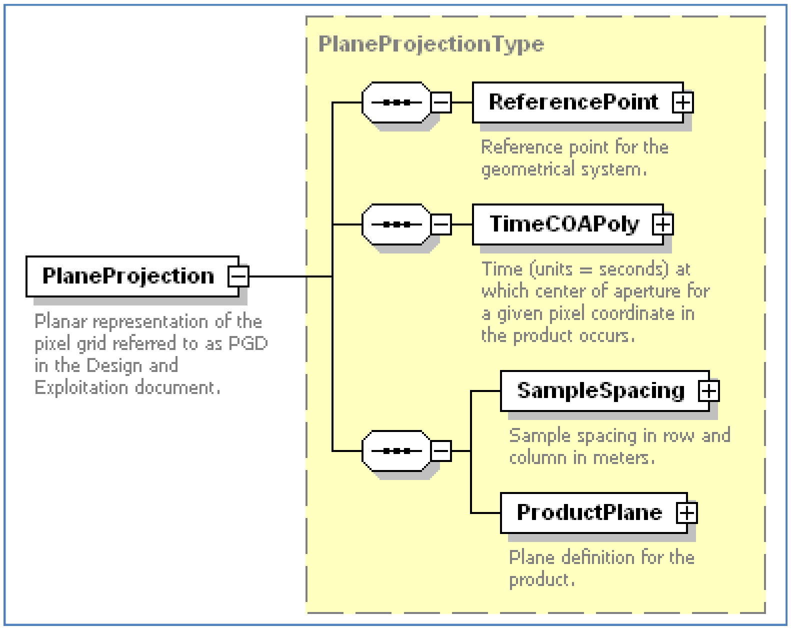


Figure 4-24 *PlaneProjection* Structure

Table 4-23 *PlaneProjection* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ReferencePoint	ReferencePointType	Figure 4-7	Required parameter describing the reference point of the plane. (Defined as $\boldsymbol{P_{PGD}}$ in Section 2.5). The name attribute is used to describe the plane projection. The plane projection options are defined in Section 2.4 and 2.6 as $\mathbf{P}_{PGD}$ and $\mathbf{P}_{ECEF}$.
TimeCOAPoly	Poly2DType	Figure 4-4	Time at which center of the aperture for a given pixel coordinate in the product occurs. The *TimeCOAPoly* is a required parameter.
SampleSpacing	RowColDoubleType	Table 4-3	Required parameter defining the spacing between pixels in the row and column directions.

Table 4-23 *PlaneProjection* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ProductPlane	PlaneProjectionType	Table 4-24 Figure 4-25	Required parameter describing the plane definition for the product.

4.2.4.4 Overview and Parameter Definition: *ProductPlane*

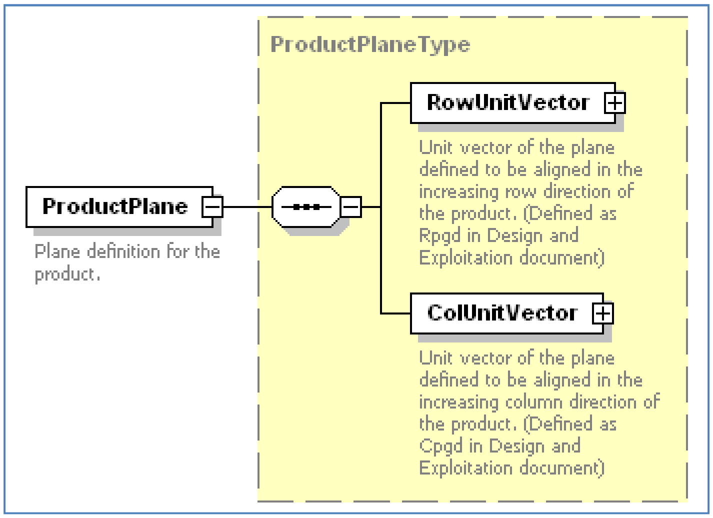


Figure 4-25 *ProductPlane* Structure

Table 4-24 *ProductPlane* Parameter Definition			
Parameter	**Type**	**Reference**	**Definition**
RowUnitVector	XYZType	Table 4-3	Required parameter describing the unit vector of the plane that is defined to be aligned in the increasing row direction of the product. (Defined as $\boldsymbol{R_{PGD}}$ in Section 2.5)
ColUnitVector	XYZType	Table 4-3	Required parameter describing the unit vector of the plane that is defined to be aligned in the increasing column direction of the product. (Defined as $\boldsymbol{C_{PGD}}$ in Section 2.5)

4.2.4.5 Overview and Parameter Definition: *GeographicProjection*

GeographicProjection is a complex structure within the *Measurement* grouping. The complex structure layout can be seen in Figure 4-26. The parameters within the *GeographicProjection* provide a geographic mapping of the pixel grid. The definition and the primitives are listed in Table 4-25.

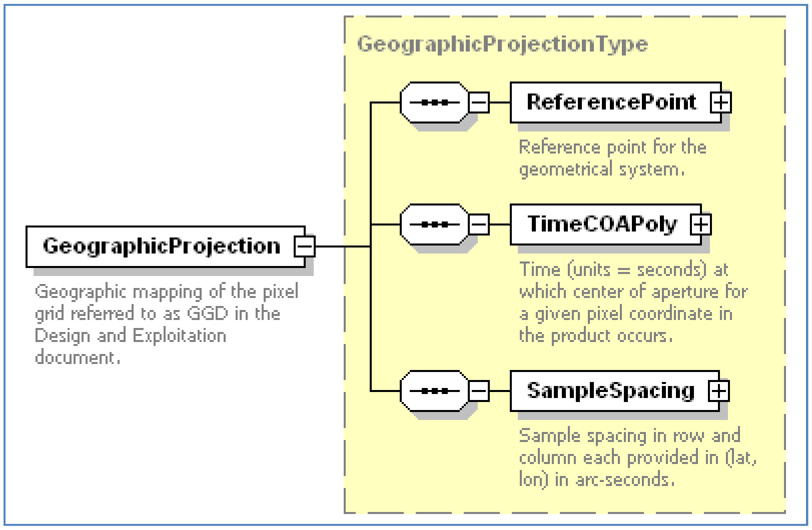


Figure 4-26 *GeographicProjection* Structure

Table 4-25 *GeographicProjection* Parameter Definitions

Parameter	Type	Reference	Definition
ReferencePoint	ReferencePointType	Figure 4-7	Required parameter describing the reference point of the geographic grid, usually the scene center point. The name attribute is used to describe the plane projection. (Defined in Section 2.6)
TimeCOAPoly	Poly2DType	Figure 4-4	Time at which center of the aperture for a given pixel coordinate in the product occurs. The *TimeCOAPoly* is a required parameter.
SampleSpacing	RowColLatLonType	Table 4-3	Required parameter describing the pixel spacing (units = arc seconds).

4.2.4.6 Overview and Parameter Definition: *CylindricalProjection*

CylindricalProjection is a complex structure within the *Measurement* grouping. The complex structure layout can be seen in Figure 4-26. The parameters within the *CylindricalProjection* provide a cylindrical mapping of the pixel grid. The definition and the primitives are listed in Table 4-25.

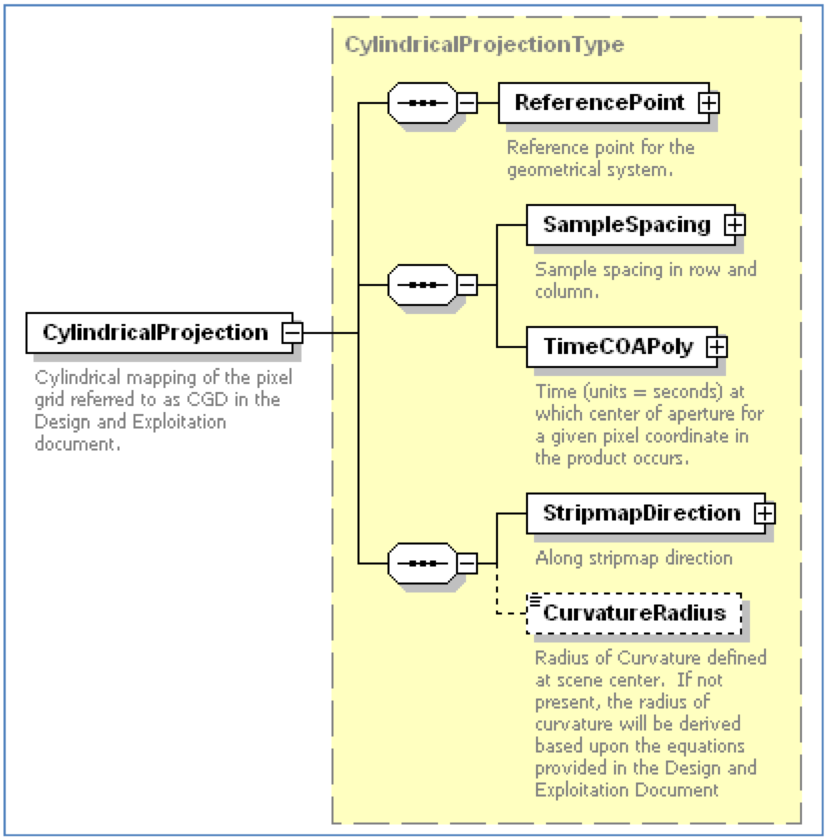


Figure 4-27 *CylindricalProjection* Structure

Table 4-26 CylindricalProjection Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ReferencePoint	ReferencePointType	Figure 4-7	Required parameter describing the reference point of the cylindrical grid usually the scene center point. The name attribute is used to describe the plane projection. (Defined in Section 2.8).
TimeCOAPoly	Poly2DType	Table 1-3 Reference 2 Figure 4-4	Time at which center of the aperture for a given pixel coordinate in the product occurs. The *TimeCOAPoly* is a required parameter.
SampleSpacing	RowColDoubleType	Table 4-3	Required parameter describing the pixel spacing (units = meters).
StripmapDirection	XYZType	Table 4-3	Required parameter describing the along stripmap direction.

Table 4-26 CylindricalProjection Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
CurvatureRadius	Double	Table 1-3 Reference 2	Optional parameter defining the Radius of Curvature at scene center. If not present, the radius of curvature will be derived based upon the equations provided in Section 3.

4.2.5 *ExploitationFeatures*

The *ExploitationFeatures* structure contains additional metadata parameters needed for advanced exploitation tasks such as creating legends and icons.

The complex structures for the *ExploitationFeatures* grouping are shown in Figure 4-28 through Figure 4-35. The grouping definitions are defined in Table 4-27 through Table 4-35.

4.2.5.1 Parameters and Complex Structures: *ExploitationFeatures*

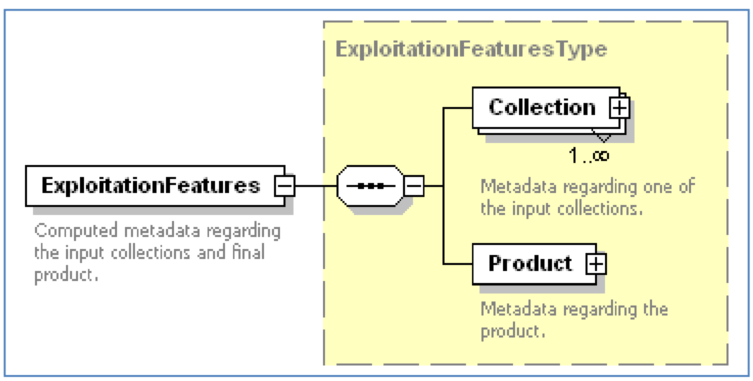


Figure 4-28 *ExploitationFeatures* Parameter and Complex Structures

Table 4-27 *ExploitationFeatures* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Collection	ExploitationFeaturesCollectionType	Table 4-28	Required metadata regarding one or more of the input collections. The attribute is used as an identifier for the collection.
Product	ExploitationFeaturesProductType	Table 4-35 Figure 4-35	Required metadata regarding the product.

4.2.5.2 Overview and Parameter Definition: *Collection*

The ExploitationFeaturesCollection parameter contains metadata from the input collections. This is different from the product parameters in that they contain metadata describing exploitation features which were calculated with regard to the derived product.

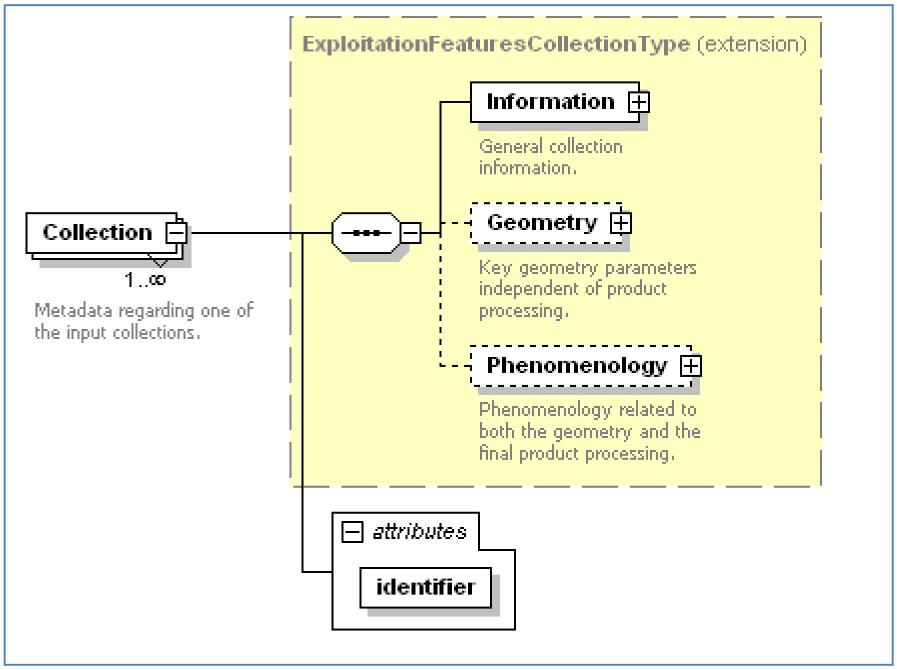


Figure 4-29 *Collection* Structure

Table 4-28 *Collection* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Information	ExploitationFeaturesCollectionInformationType	Table 4-29 Figure 4-30	Required parameter that contains general input collection information.

Table 4-28 *Collection* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Geometry	ExploitationFeaturesCollectionGeometryType	Table 4-33 Figure 4-33	Optional parameter that provides key geometry parameters independent of the product processing.
Phenomenology	ExploitationFeaturesCollectionPhenomenology Type	Table 4-34 Figure 4-34	Optional parameter that contains phenomenology related to both the geometry and the final product's processing.

4.2.5.3 Overview and Parameter Definition: *Information*

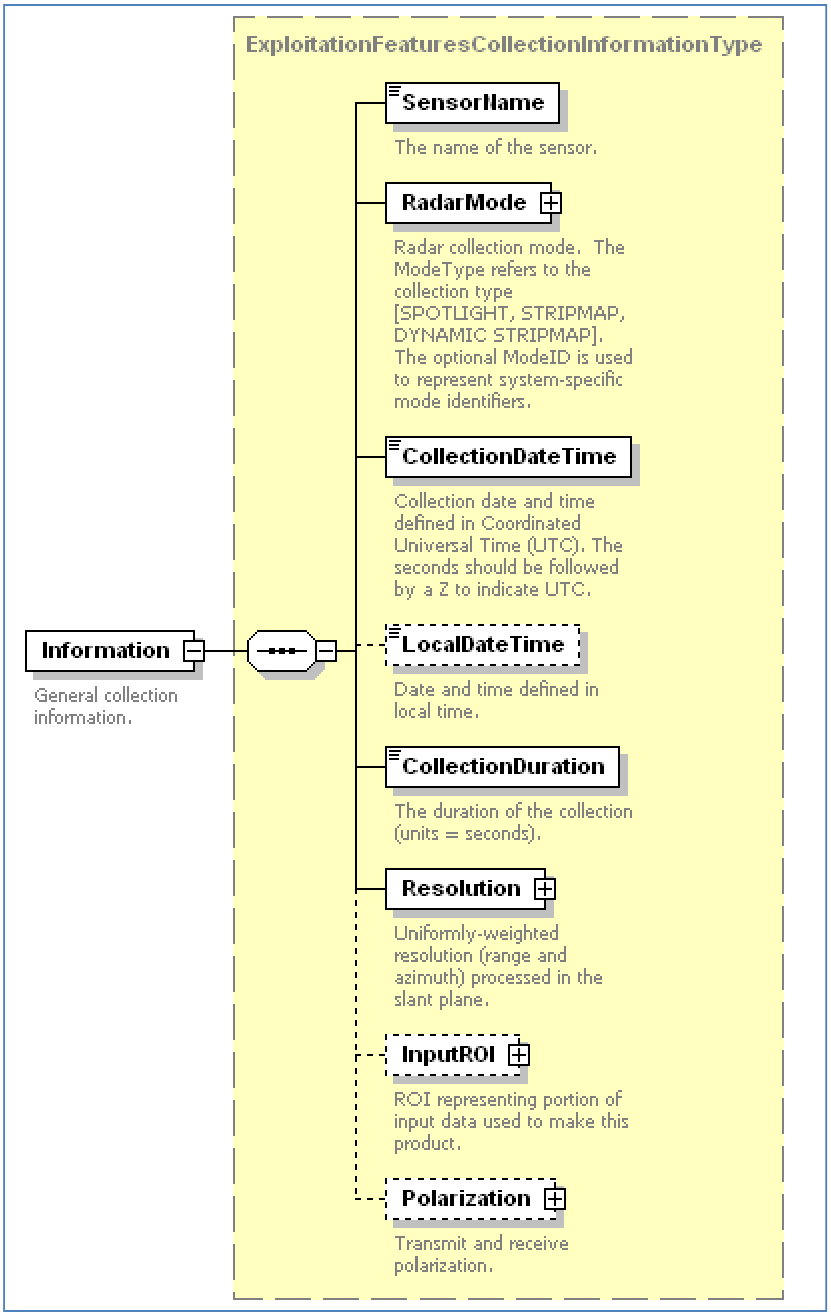


Figure 4-30 *Information* Structure

Table 4-29 *Information* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
SensorName	String	Table 1-3 Reference 2	Required parameter used to identify the name of the sensor.
RadarMode	RadarModeType	Table 4-3	Required parameter describing radar collection mode.
CollectionDateTime	DateTime	Table 1-3 Reference 2	Required parameter describing collection date and time defined in Coordinated Universal Time (UTC). The seconds should be followed by a Z to indicate UTC.
LocalDateTime	DateTime	Table 1-3 Reference 2	Optional parameter describing the date and time defined in local time.
CollectionDuration	Double	Table 1-3 Reference 2	Required parameter describing the duration of the collection (units = s).
Resolution	RangeAzimuthType	Table 4-3	Required parameter containing the uniformly-weighted resolution (range and azimuth) processed in the slant plane.
InputROI	InputROIType	Table 4-30 Figure 4-31	Optional ROI (region of interest) parameter representing portion of input data used to make the derived product.
Polarization	TxRcvPolarizationType	Table 4-31	Optional parameter describing the angle of receive polarization. Please refer to Section 6.3.1.

4.2.5.4 Overview and Parameter Definition: *InputROI*

The InputROI parameter is used only when processing from SICD or the other input data has a parameter that lists the input region of interest.

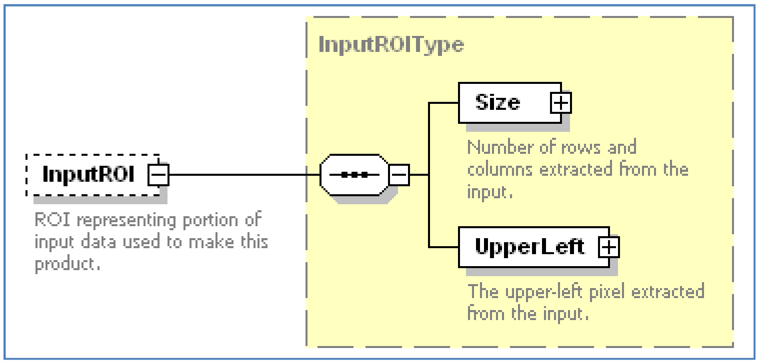


Figure 4-31 *InputROI* Structure

Table 4-30 *InputROI* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Size	RowColIntType	Table 4-3	Required parameter describing the number of rows and columns extracted from the input.
UpperLeft	RowColIntType	Table 4-3	Required parameter describing the upper-left pixel extracted from the input.

4.2.5.5 Overview and Parameter Definition: *Polarization*

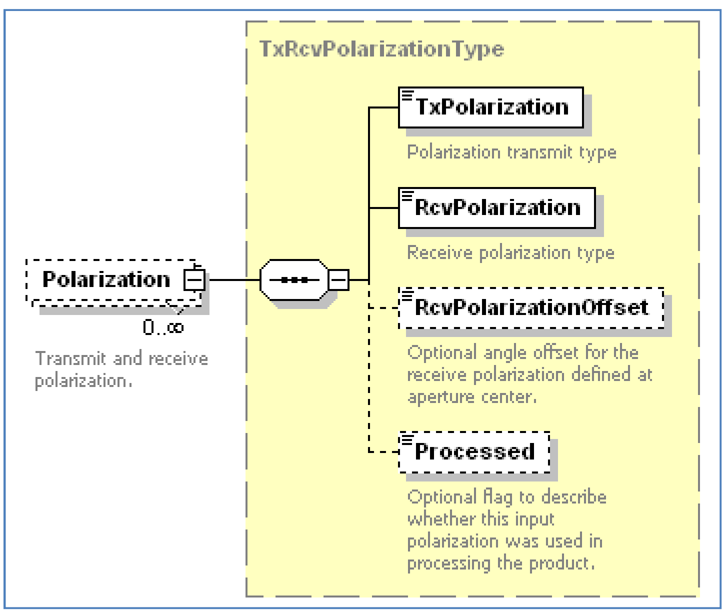


Figure 4-32 *TxRcvPolarization* Structure

Table 4-31 *TxRcvPolarizationType* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
TxPolarization	PolarizationType	Table 4-32	Polarization transmit type
RcvPolarization	PolarizationType	Table 4-32	Polarization receive type
RcvPolarizationOffset	neg180to180type	Table 4-2	Angle from transmit to receive polarization defined at center of aperture.
Processed	Boolean	Table 1-3 Reference 2	Describes whether this polarization was used in processing.

Table 4-32 Enumeration Types for *Polarization*	
Enumeration Type	**Definition**
V	Vertical polarization
H	Horizontal polarization
RHC	Right-handed circular polarization
LHC	Left-handed circular polarization
OTHER	Other type polarization

4.2.5.6 Overview and Parameter Definition: *Geometry*

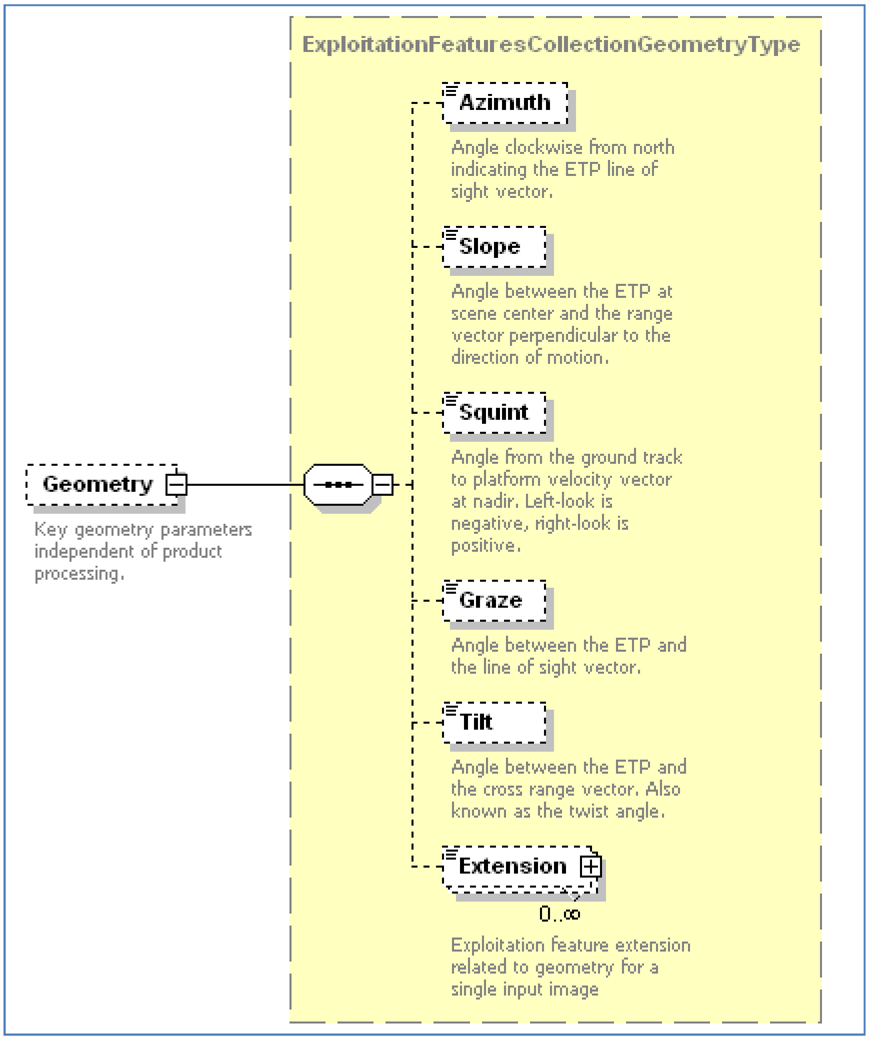


Figure 4-33 *Geometry* Structure

Table 4-33 *Geometry* Parameter Definitions

Parameter	Type	Reference	Definition
Azimuth	Double	Table 1-3 Reference 2	Optional parameter describing the angle clockwise from north indicating the EGTP line of sight vector. Refer to Section 6.4.1 for calculations.

Table 4-33 *Geometry* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Slope	ZeroTo90Type	Table 4-2	Optional parameter describing the angle between the EGTP at scene center and the range vector perpendicular to the direction of motion (velocity). Refer to Section 6.4.2 for calculations.
Squint	Neg180To180Type	Table 4-2	Optional parameter describing the angle from the platform velocity vector. Left-look is negative, right-look is positive. Refer to Section 6.4.3 for calculations.
Graze	ZeroTo90Type	Table 4-2	Optional parameter describing the angle between the EGTP and the line of sight vector. Refer to Section 6.4.4 for calculations.
Tilt	Neg180To180Type	Table 4-2	Optional parameter describing the angle between the EGTP and the cross range vector. Also known as the twist angle. Refer to Section 6.4.5 for calculations.

4.2.5.7 Overview and Parameter Definition: *Phenomenology*

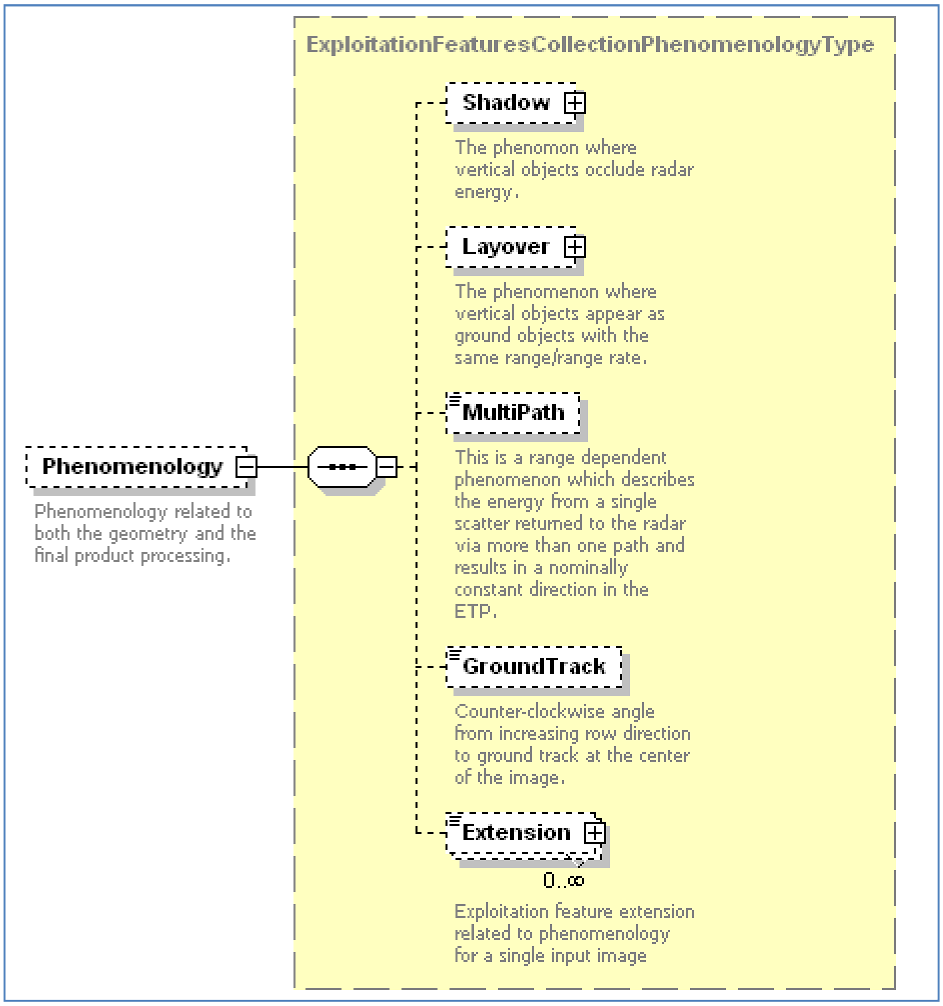


Figure 4-34 *Phenomenology* Structure

Table 4-34 *Phenomenology* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Shadow	AngleMagnitudeType	Table 4-3	Optional parameter describing the phenomenon where vertical objects occlude radar energy. Refer to Section 6.5.1 for calculations.

Table 4-34 *Phenomenology* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Layover	AngleMagnitudeType	Table 4-3	Optional parameter describing the phenomenon where vertical objects appear as ground objects with the same range/range rate. Refer to Section 6.5.2 for calculations.
MultiPath	Neg180To180Type	Table 4-2	Optional parameter that is a range dependent phenomenon which describes the energy from a single scatter returned to the radar via more than one path and results in a nominally constant direction in ground plane imagery. Refer to Section 6.5.5 for calculations.
GroundTrack	Neg180To180Type	Table 4-2	Optional parameter that describes the angle from increasing row direction to the ground track at the center of the image.
Extension	ParameterType	Table 4-3	Exploitation feature extension related to the phenomenology for a single input image.

4.2.5.8 Overview and Parameter Definition: *Product*

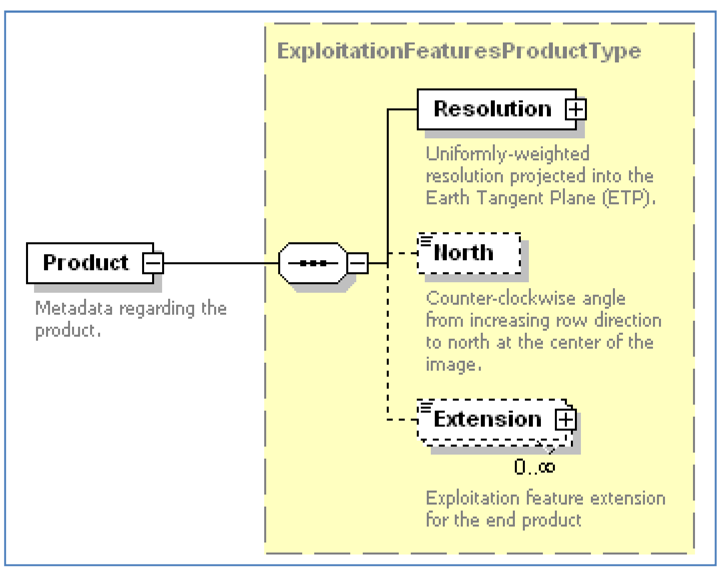


Figure 4-35 *Product* Structure

Table 4-35 *Product* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
Resolution	RowColDoubleType	Table 4-3	Required parameter describing the uniformly-weighted resolution projected into the Earth Geodetic Tangent Plane (EGTP). Refer to Section 7 for calculations.
North	Neg180To180Type	Table 4-2	Optional parameter describing the angle from increasing row direction to north at the center of the image. Refer to Section 6.5.3 for calculations.
Extension	ParameterType	Table 4-3	Optional parameter for the product complex structure which can be used to extend the metadata supported.

4.2.6 *ProductProcessing*

The *ProductProcessing* grouping contains free parameters to allow the product developer to define product and implementation-specific parameters that describe the "algorithm-specific" processing done to create the detected image product. The complex structures for the grouping are shown in

Figure 4-36 through Figure 4-39.

4.2.6.1 Parameters and Complex Structures: *ProductProcessing*

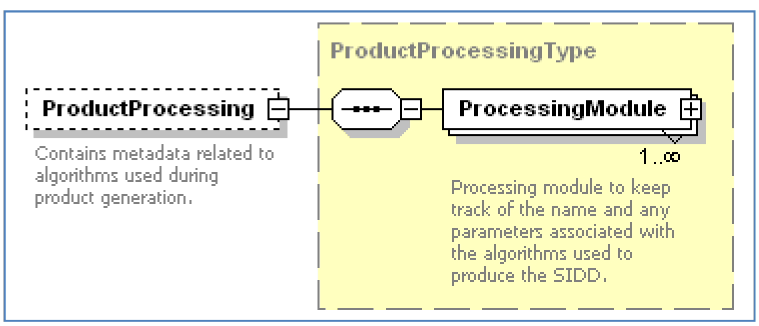


Figure 4-36 *ProductProcessing* Parameter and Complex Structures

Table 4-36 *ProductProcessing* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ProductProcessing	ProcessingModuleType	Figure 4-36 Table 4-36	Required processing module to track the name and any parameters associated with the algorithms used to produce the SIDD.

4.2.6.2 Overview and Parameter Definition: *ProcessingModule*

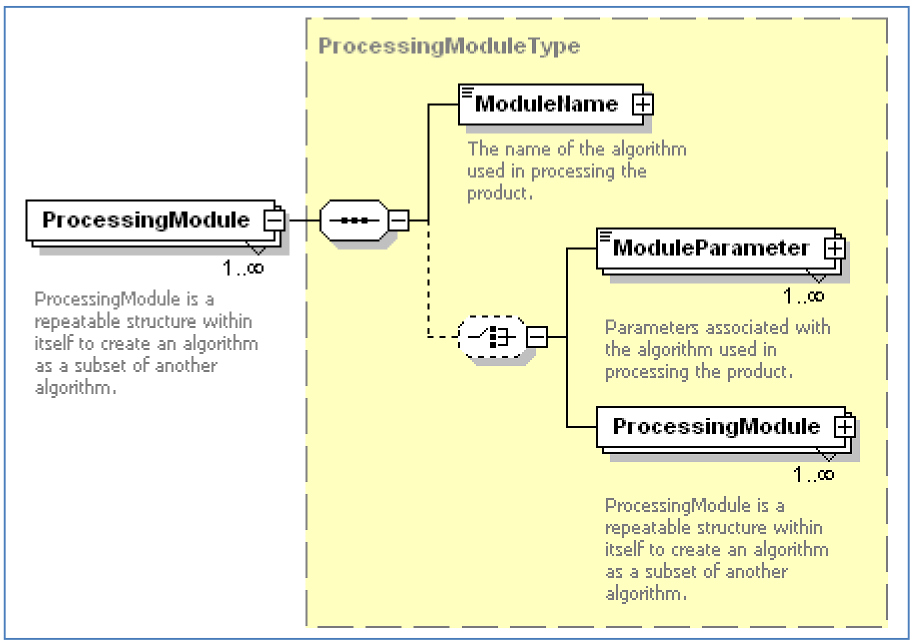


Figure 4-37 *ProcessingModule* Structure

Table 4-37 *ProcessingModule* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ModuleName	ParameterType	Table 4-3	Required parameter containing the name of the algorithm used in processing the product.

ModuleParameter	ParameterType	Table 4-3	Optional switch parameter associated with the algorithm used in the processing the product.
ProcessingModule	ProcessingModuleType	Figure 4-37 Table 4-37	Optional switch parameter that is a repeatable structure within itself to create an algorithm as a subset of another algorithm.

4.2.7 *DownstreamReprocessing*

The optional *DownstreamReprocessing* grouping contains information about processing performed on a SIDD product by an exploitation tool. This grouping should not be included in the SIDD metadata during initial product creation; it will only be included when an ELT or other tool performs post-processing on a SIDD product. When any tool performs post-processing on a SIDD product, this grouping will be used by the tool to document the post-processing that occurred.

4.2.7.1 Parameter and Complex Structures: *DownstreamReprocessing*

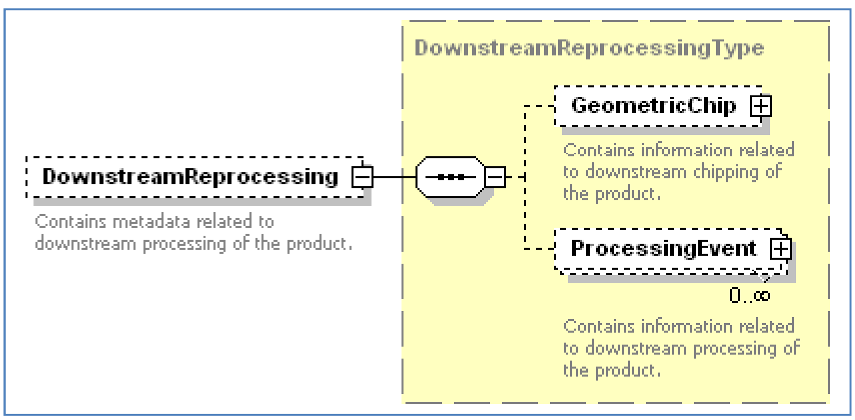


Figure 4-38 *DownstreamReprocessing* Parameter and Complex Structures

Table 4-38 DownstreamReprocessing Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**

GeometricChip	GeometricChipType	Figure 4-39 Table 4-39	Contains information related to downstream chipping of the product. There is only one instance, and the instance is updated with respect to the full image parameters. For example, if an image is chipped out of a smaller chip, the new chip needs to be updated to the original full image corners.
ProcessingEvent	ProcessingEventType	Figure 4-40 Table 4-40	Contains information related to downstream processing of the product.

4.2.7.2 Overview and Parameter Definition: *GeometricChip*

The chipping metadata contains the relationship between chipped product's corner coordinates, and the original, full-image corner coordinates. Since this relationship is linear, bi-linear interpolation is sufficient to determine an arbitrary chip coordinate in terms of the original full image coordinates. Chipping is typically done using an exploitation tool, and should not be done in the initial product creation.

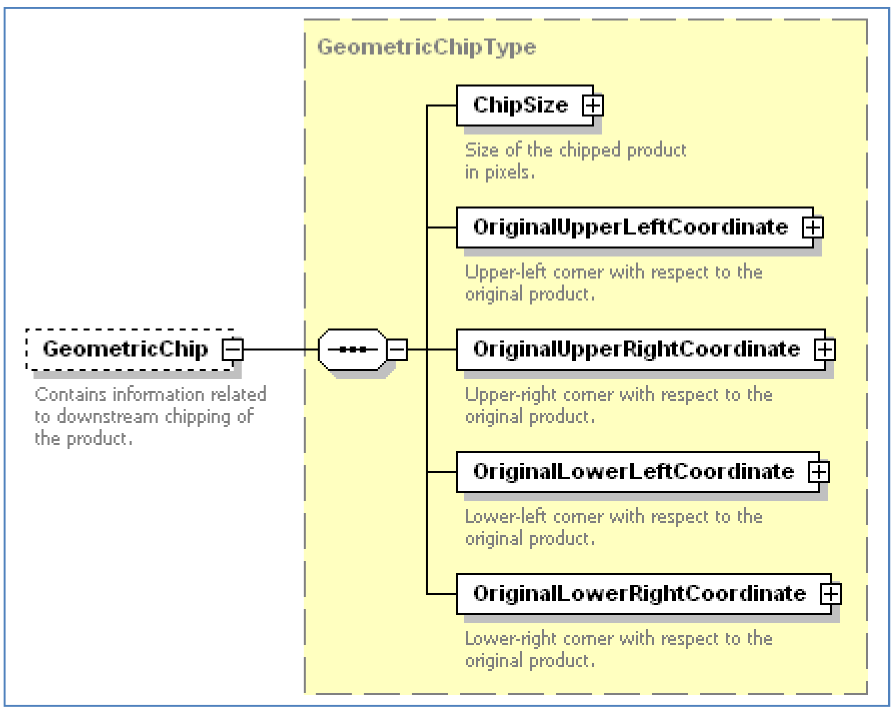


Figure 4-39 *GeometricChip* Structure

Table 4-39 *GeometricChip* Parameter Definitions			
Parameter	**Type**	**Reference**	**Definition**
ChipSize	RowColIntType	Table 4-3	Specifies the size of the chipped product in pixels.
OriginalUpperLeftCoordinate	RowColDoubleType	Table 4-3	The *OriginalUpperLeftCoordinate* element describes the chipped image corner that is associated with the original upper left corner of the full image. The chipped image corner corresponds to **{0,0}** and the original upper left full image corner corresponds to **{R_1,C_1}** in Figure 5-1.
OriginalUpperRightCoordinate	RowColDoubleType	Table 4-3	The *OriginalUpperRightCoordinate* element describes the chipped image corner that is associated with the original upper right corner of the full image. The chipped image corner corresponds to **{0,cCols-1}** and the original upper right full image corner corresponds to **{R_2,C_2}** in Figure 5-1.
OriginalLowerLeftCoordinate	RowColDoubleType	Table 4-3	The *OriginalLowerLeftCoordinate* element describes the chipped image corner that is associated with the original lower left corner of the full image. The chipped image corner corresponds to **{cRows-1, 0}** and the original lower left full image corner corresponds to **{R_4,C_4}** in Figure 5-1.
OriginalLowerRightCoordinate	RowColDoubleType	Table 4-3	The *OriginalLowerRightCoordinate* element describes the chipped image corner that is associated with the original lower right corner of the full image. The chipped image corner corresponds to **{cRows-1, cCols-1}** and the original lower right full image corner corresponds to **{R_3,C_3}** in Figure 5-1.

4.2.7.3 Overview and Parameter Definition: *ProcessingEvent*

Each time an exploitation tool performs post-processing an instance of the *ProcessingEvent* complex structure is added to the *ProductProcessing* grouping.

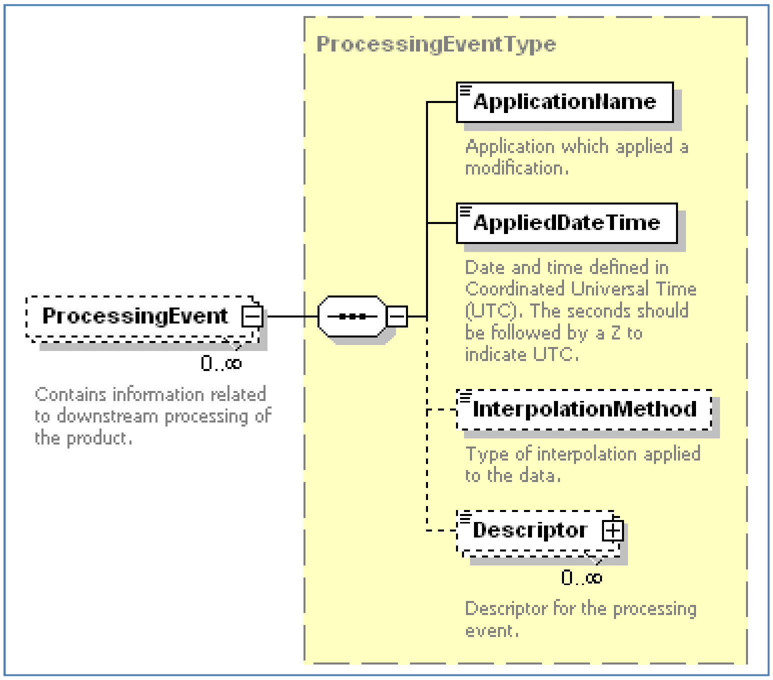


Figure 4-40 *ProcessingEvent* Structure

Table 4-40 *ProcessingEvent* Parameter Definitions

Parameter	Type	Reference	Definition
ApplicationName	String	Table 1-3 Reference 2	Require parameter used to describe the application (usually referring to specific exploitation tool and version) that applied a modification.
AppliedDateTime	DateTime	Table 1-3 Reference 2	Required parameter used to describe the date at which the processing was applied.
InterpolationMethod	String	Table 1-3 Reference 2	Optional parameter used to describe the type of interpolation applied to the data.
Descriptor	ParameterType	Table 3-2	Optional descriptor for the processing event.

4.2.8 ErrorStatistics

The optional *ErrorStatistics* grouping will be carried forward from the SICD metadata when available. In the case where there are multiple SICD inputs, the profile (see Section 1.3) will define which should be propagated from the SICD. The top level layout for this grouping can be seen in Figure 4-41. For more details, see the *SICD Design & Implementation Description Document* referenced in Table 1-2.

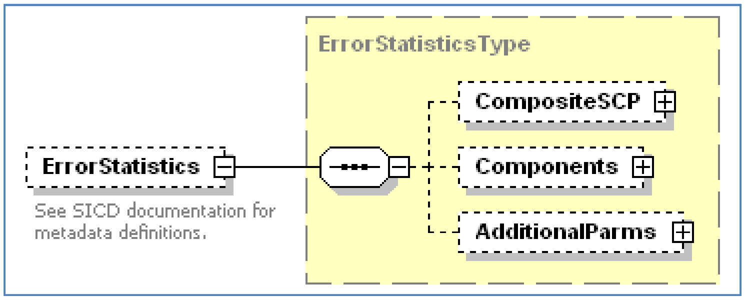


Figure 4-41 *ErrorStatistics* Schema Layout

4.2.9 Radiometric

The optional *Radiometric* is defined in the SICD specifications. All elements map pixel intensities to the radiometric parameter. The top level layout for this grouping can be seen in Figure 4-42. For more details, see the *SICD Design & Implementation Description Document* referenced in Table 1-2.

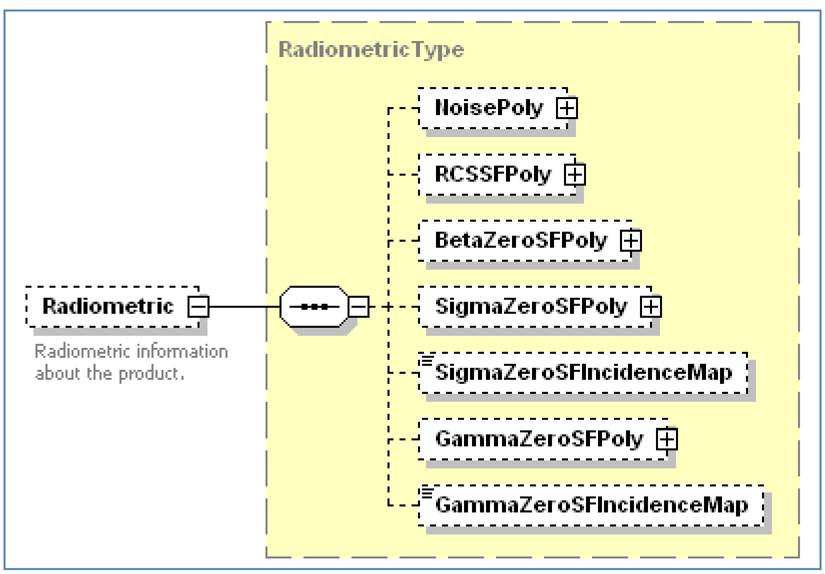


Figure 4-42 *Radiometric* Schema Layout

4.2.10 Annotations

OpenGIS Simple Feature Access (SFA) is an ISO standard (19125) for describing simple features in a spatial reference system. The feature representation is limited to objects like points, curves, polygons, surfaces, etc. This standard is considered a profile for the Geographic information – Spatial schema (ISO 19107:2003). In addition, this representation has a direct mapping for geospatial databases provided in SQL Call-Level Interface (SQL/CLI) (ISO/IEC 9075-3:2003).

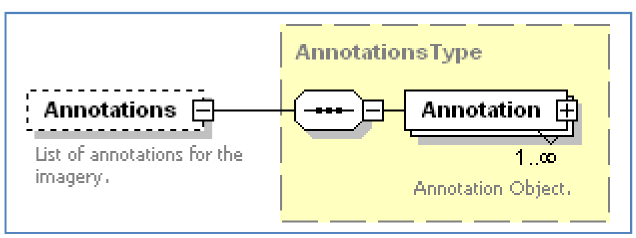
SIDD provides a direct mapping in XML of the SFA. The SFA XML objects, along with their SIDD textual descriptions, define an annotation which maps directly to common geospatial databases. In addition, SFA is simple enough such that SIDD producers and exploiters, whether they are ELTs, libraries, or engineers, can read the definition and understand the annotation without requiring a complex interpretation engine.

The main components of the SFA are geometrical objects, a spatial reference system, and a measure reference system. Geometrical objects (*Object*) provide a mapping between the SIDD textual description of the annotation and the representation in 2-, 3-, or 4-space. For SIDD, the SFA implementation is limited to a 3-dimensional geocentric representation or a 2-dimensional geographic representation. The spatial reference system (*SpatialReferenceSystem*) identifies 2 or 3-dimensional coordinate space that the data lies in (See Section 9, Well-known Text Representation of Spatial Reference System, of OGC 06-103r3). The measure references system (*MeasureReferenceSystem*) defines an optional 4th dimension for the coordinate space. This 4th dimension is not permitted in SIDD, but it is has been modeled in XML schema for completeness of the SFA representation.

The SFA annotation representation in SIDD is shown in Figure 4-43 and maps the SFA objects to the textual description of the annotation (*Identifier*). This provides a direct mapping of an object or collection of objects to an identifier.

Use of SFA annotations in SIDD XML does not prohibit other forms of annotation supported by the file format specifications, like CGM within NITF graphics segments. Where possible, objects identified in those ancillary annotation formats should be duplicated within the SIDD XML as SFA *Objects*.

4.2.10.1 Overview and Parameter Definition: Annotation



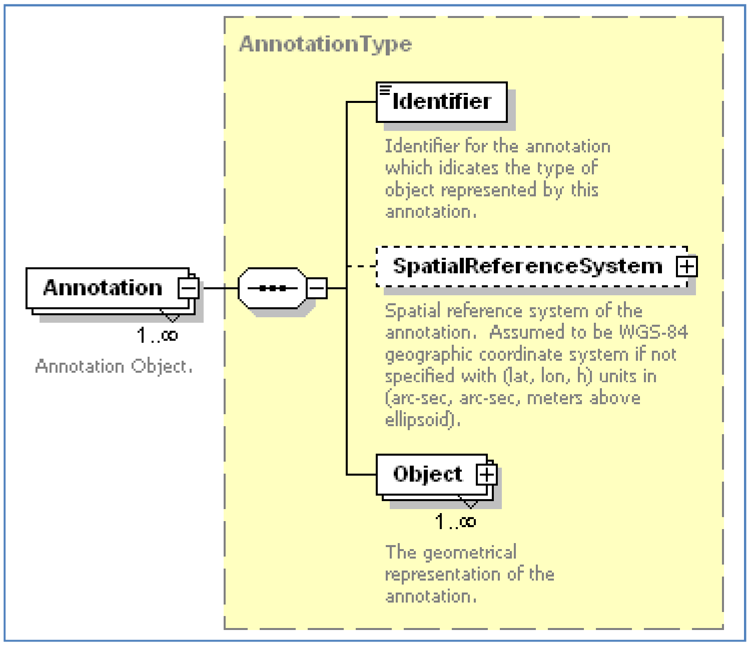


Figure 4-43 *Annotation* Parameter and Complex Structures

Table 4-41 *Annotation* Parameter Definitions

Parameter	Type	Reference	Definition
Identifier	String	Table 1-3 Reference 2	Descriptor for the object(s) that this annotation represents.
SpatialReferenceSystem	ReferenceSystemType	Figure 4-44	(Not Used) This is the geospatial reference system for the annotation. SIDD does not currently support different annotation reference systems. This is assumed to be WGS-84.
Object	AnnotationObjectType	Figure 4-45	List of objects that constitute the annotation.

4.2.10.2 Overview and Parameter Definition: SpatialReferenceSystem

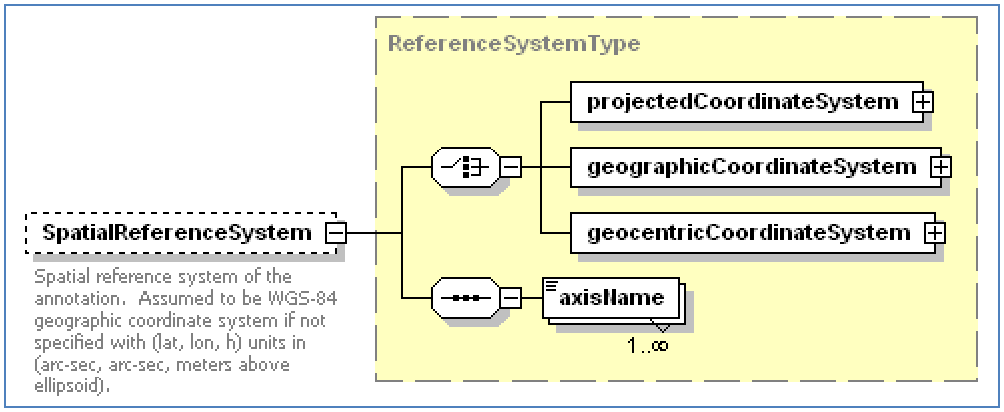


Figure 4-44 *SpatialReferenceSystem* Schema Layout

The optional *SpatialReferenceSystem* is currently not supported by SIDD. SIDD assumes that all annotation objects lay within the SFA-defined WGS-84 *GeographicCoordinateSystem* when only two dimensions are used, and the *GeocentricCoordinateSystem* when three dimensions are used.

4.2.10.3 Overview and Parameter Definition: Object

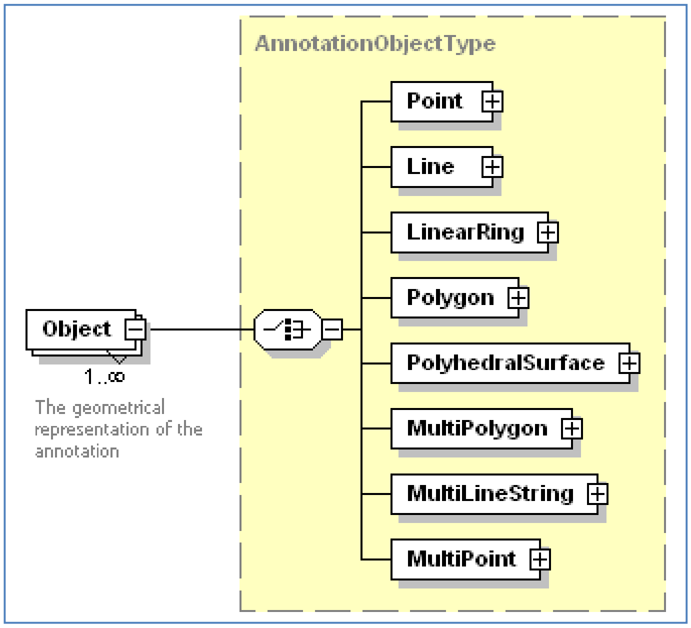


Figure 4-45 *Object* Schema Layout

Table 4-42 Object Parameter Definition		
Parameter	**Schema**	**Reference**
Point	PointType	Table 1-3 Reference 3
Line	LineType	Table 1-3 Reference 3
LinearRing	LinearRingType	Table 1-3 Reference 3
Polygon	PolygonType	Table 1-3 Reference 3
PolyhedralSurface	PolyhedralSurfaceType	Table 1-3 Reference 3
MultiPolygon	MultiPolygonType	Table 1-3 Reference 3
MultiLineString	MultiLineStringType	Table 1-3 Reference 3
MultiPoint	MultiPoint	Table 1-3 Reference 3

Annotations are described using ISO Standard 19125 definitions. The ISO standard provides the full definition of these geometrical objects; therefore, it is not repeated within the SIDD D&I.

4.2.10.4 SFA PointType

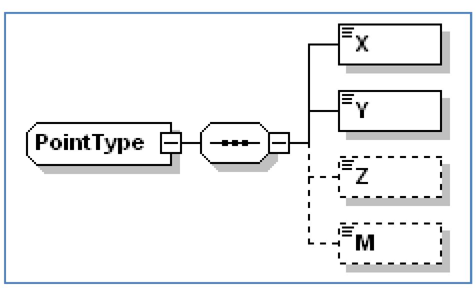


Figure 4-46 Primitive Definition & Schema Layout: *PointType*

All SFA *Object*s utilize the *SFA:PointType* as their coordinate encoding mechanism. For a *GeocentricCoordinateSystem X*, *Y*, and *Z* represent ECEF coordinates in meters. For a WGS-84

GeographicCoordinateSystem X and *Y* represent latitude and longitude respectively in arcseconds.

5 User Exploitation

The purpose of this section is to describe basic operations that can be done to a SIDD product. Included in this section is the procedure for chipping a SIDD product.

5.1.1 Chipping

The purpose of this section is to describe the procedure for relating a chipped image to its original full product SIDD. The chipping supports three linear operations; translation, rotation, and scaling. Once a chipping operation has been done, it is necessary to continuously map the newly formed chip coordinates to the full original image coordinates; this ensures that all capabilities on the original full image product are also supported on the chip image product. To enable the correct mapping, the coordinates of the chipped image product in terms of the original full image product coordinates must be recorded, see Table 4-39. For the purposes of chipping, all pixel coordinates refer to the center of the pixel.

Figure 5-1 shows a chipped product of chip size cRows and cCols relative to the full image product. The chip has been rotated, translated and scaled. The corners are recorded in full image coordinates as $\{R_1,C_1\}$ through $\{R_4,C_4\}$, the corresponding chip coordinates are listed below these values. The chip coordinate {r,c} can be related to the original full image chip coordinate {R,C} through bi-linear interpolation.

The bi-linear interpolation method is developed for the rows below, it is then listed for both the rows and columns following Figure 5-1.

The unknown value, R, is estimated from the four known corner coordinates,R_1 through R_4, the matrix form of bi-linear interpolation is below. The unknown value, C, is estimated in a similar manner.

$$R = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} R_1 & R_2 \\ R_4 & R_3 \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

$$u, v \in [0,1]$$

The above equation can be re-written as:

$$R = R_1 + (R_4 - R_1)u + (R_2 - R_1)v + (R_1 + R_3 - R_2 - R_4)uv$$

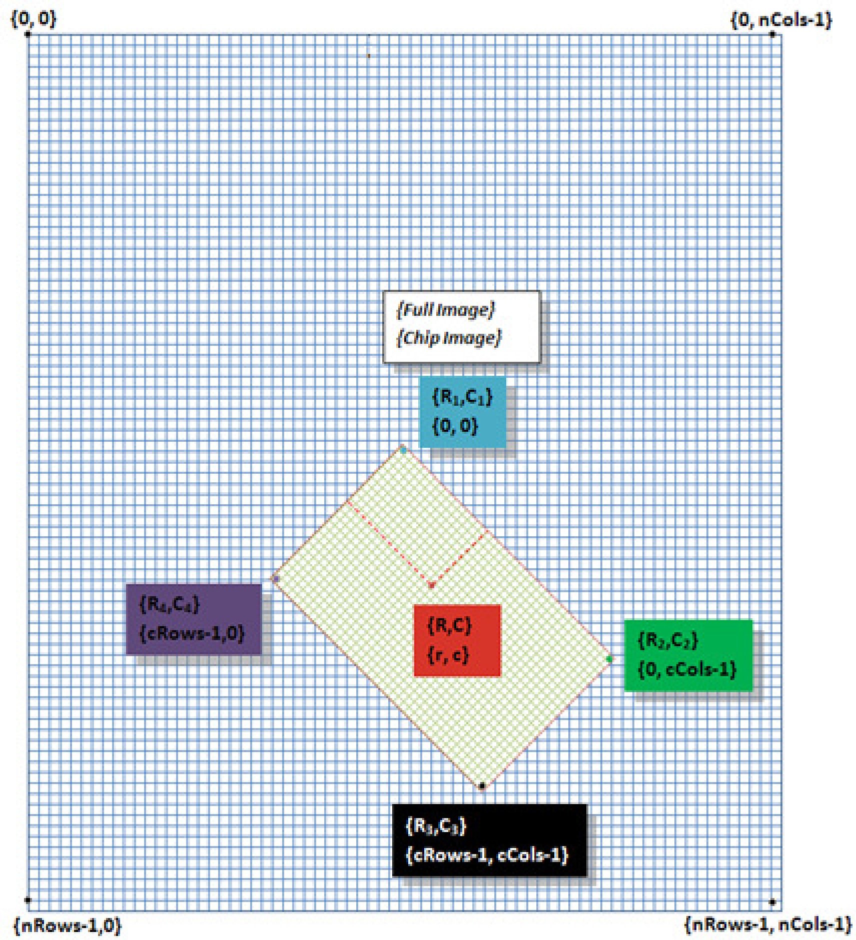


Figure 5-1 Chipping Diagram

The steps for computing the original full image coordinates from the chipped product are outlined below.

Step 1: Normalize the chip coordinates

$$u \equiv \frac{r}{cRows - 1}, v \equiv \frac{c}{cCols - 1}$$

Step 2: Compute original full image row coordinate bi-linear coefficients

$$A_r = R_1$$
$$B_r = R_4 - R_1$$
$$D_r = R_2 - R_1$$
$$F_r = R_1 + R_3 - R_2 - R_4$$

Step 3: Compute original full image column coordinate bi-linear coefficients

$$A_c = C_1$$
$$B_c = C_4 - C_1$$
$$D_c = C_2 - C_1$$

$$F_c = C_1 + C_3 - C_2 - C_4$$

Step 3: Compute the full image row and column coordinate

$$R = A_r + uB_r + vD_r + uvF_r$$

$$C = A_c + uB_c + vD_c + uvF_c$$

6 ExploitationFeatures Calculations

The purpose of this section is to provide definitions for the angle metadata defined in the *ExploitationFeaturesType* grouping. In addition to the definitions, derivations are provided for each angle defined in the grouping.

The following Table 6-1 defines the variables used in the derivations. The following figures diagram the angles and variables defined in the Table 6-1.

Table 6-1 Variables & Angles

Variables	Definitions	Units
ψ	Grazing Angle	Degrees
ψ_0	Slope Angle	Degrees
ϕ_s	Doppler Cone Angle	Degrees
ϕ_g	Ground Plane Squint Angle	Degrees
$\mathbf{P_a}$	Antenna Position in ECEF coordinates	Meters
$\mathbf{V_a}$	Antenna Velocity in ECEF coordinates	Meters per second
$\mathbf{P_o}$	Scene Center Point in ECEF coordinates	Meters
$\mathbf{Z_g}$	Earth Tangent Plane Normal in ECEF coordinates	Unitless
$\hat{}$	Indicates a vector of unit magnitude	N/A
φ	Latitude at Scene Center Point	Decimal Degrees
λ	Longitude at Scene Center Point	Decimal Degrees
$\mathbf{r}$	Unit Vector in increasing rows, in ECEF coordinates	Unitless
$\mathbf{c}$	Unit Vector in increasing columns, in ECEF coordinates	Unitless
θ_r	CCW rotation angle of output	Degrees

Table 6-1 Variables & Angles		
Variables	**Definitions**	**Units**
	product	
$\rho_{r,s}$	Slant Plane resolution in the range direction	Meters
$\rho_{c,s}$	Slant Plane resolution in the azimuth direction	Meters
$\rho_{r,g}$	Ground Plane resolution in the row direction	Meters
$\rho_{c,g}$	Ground Plane resolution in the column direction	Meters

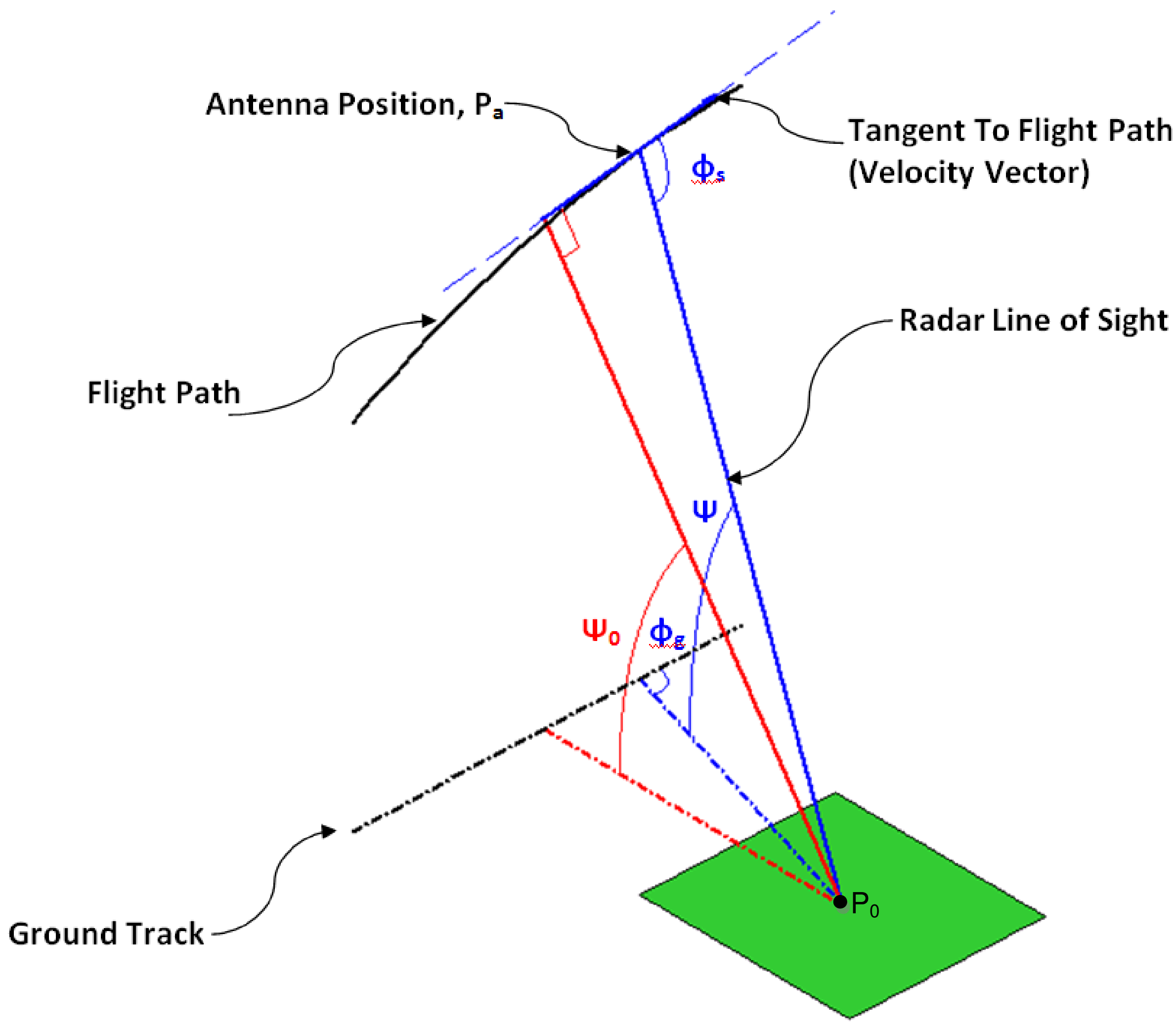


Figure 6-1 Three-Space Angle Definition Diagram

6.1 Slant Plane Definition

To simplify the derivations, it is necessary to define the slant plane. The slant plane is a plane that contains the radar line-of-sight vector and the instantaneous velocity vector. For convenience, the normal of this plane always points away from the earth.

$$\widehat{\boldsymbol{X}}_s = \frac{\boldsymbol{P}_a - \boldsymbol{P}_o}{|\boldsymbol{P}_a - \boldsymbol{P}_o|}$$

$$\hat{\boldsymbol{n}} = \frac{\widehat{\boldsymbol{X}}_S \times \boldsymbol{V_a}}{\left|\widehat{\boldsymbol{X}}_S \times \boldsymbol{V_a}\right|}$$

$$\widehat{\boldsymbol{Z}}_S = \boldsymbol{sgn}(\boldsymbol{P_o} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}$$

$$\widehat{\boldsymbol{Y}}_S = \widehat{\boldsymbol{Z}}_S \times \widehat{\boldsymbol{X}}_S$$

6.1.1 Image Plane Definition

The image plane normal is defined below.

$$\boldsymbol{z} = \boldsymbol{r} \times \boldsymbol{c}$$

6.2 Image Angle

Image angles are measured counter-clockwise (CCW) from the first column of pixels, or increasing rows. This is off the **r** vector toward the **c** vector. The four quadrant arc-tangent function should be used throughout these computations.

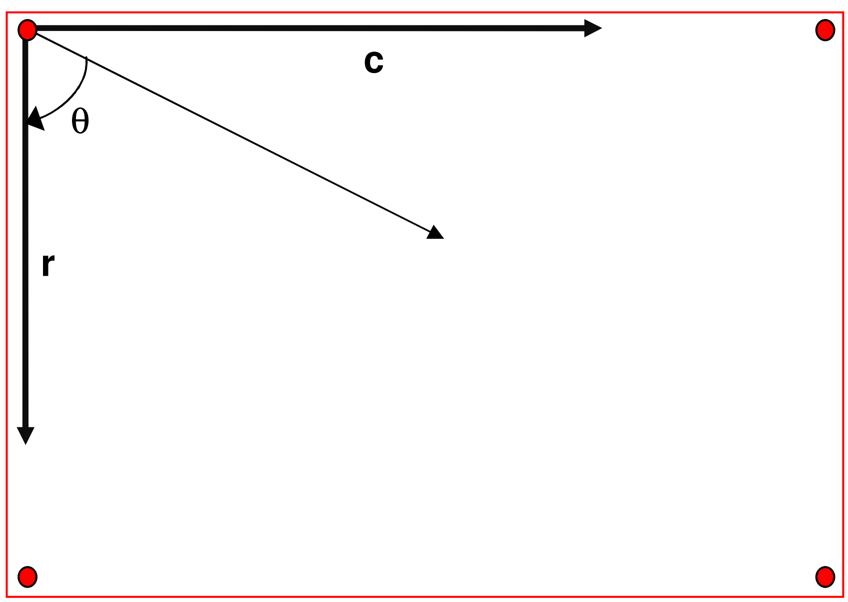


Figure 6-2 Image Angle Diagram

6.3 ExploitationFeatures

This section contains the derivations for the angles contain in the *ExploitationFeatures* grouping.

6.3.1 Polarization Angle

The polarization angle is defined from the *SICD PolarizationHVAnglePoly* evaluated at the center of aperture reference time.

6.4 Geometry

This section contains the geometry parameters independent of product processing.

6.4.1 Azimuth Angle

The azimuth angle indicates the radar line-of-sight vector on the earth. This is angle is measured clockwise from north.

$$\hat{\boldsymbol{E}} = \hat{\boldsymbol{N}} \times \hat{\boldsymbol{U}}$$

$$\theta_A = \operatorname{atan2}\left(\hat{\boldsymbol{E}} \cdot \hat{\boldsymbol{X}}_s, \hat{\boldsymbol{N}} \cdot \hat{\boldsymbol{X}}_s\right)$$

6.4.2 Slope Angle

Angle between the earth geodetic tangent plane (EGTP) and the slant plane (sometimes referred to as the broadside grazing angle).

$$\psi_0 = \cos^{-1}\left(\hat{\boldsymbol{Z}}_s \cdot \hat{\boldsymbol{Z}}_g\right)$$

6.4.3 Squint or Doppler Cone Angle

The Doppler Cone Angle is the angle between the velocity vector and the radar line-of-sight vector (see Figure 6-1). The angle is sometimes referred to as the slant plane squint angle

$$\phi_s = \cos^{-1}\left(-\hat{\boldsymbol{X}}_s \cdot \hat{\boldsymbol{V}}_a\right).$$

The ground plane squint angle is the angle between the velocity vector and the radar line-of-sight vector measured in a geocentric plane orthogonal to the ECEF vector $\mathbf{P}_a$.

$$\hat{\boldsymbol{Z}}_p = \frac{\boldsymbol{P}_a}{|\boldsymbol{P}_a|}$$

$$\boldsymbol{X}_s' = \hat{\boldsymbol{X}}_s - \left(\hat{\boldsymbol{X}}_s \cdot \hat{\boldsymbol{Z}}_p\right)\hat{\boldsymbol{Z}}_p$$

$$\boldsymbol{V}_a' = \boldsymbol{V}_a - \left(\boldsymbol{V}_a \cdot \hat{\boldsymbol{Z}}_p\right)\hat{\boldsymbol{Z}}_p$$

$$\phi_g = \cos^{-1}\left(-\hat{\boldsymbol{X}}_s' \cdot \hat{\boldsymbol{V}}_a'\right)$$

6.4.4 Grazing Angle

The Grazing angle is the angle between the earth geodetic tangent plane, EGTP, and the line-of-sight vector.

$$\psi = \sin^{-1}(\widehat{\boldsymbol{X}}_s \cdot \widehat{\boldsymbol{Z}}_g)$$

6.4.5 Tilt Angle

The tilt angle (also known as the twist angle) is the angle between the earth geodetic tangent plane, EGTP, and the cross range vector.

$$\eta = \tan^{-1}\left(\frac{\widehat{\boldsymbol{Z}}_g \cdot \widehat{\boldsymbol{Y}}_s}{\widehat{\boldsymbol{Z}}_g \cdot \widehat{\boldsymbol{Z}}_s}\right)$$

6.5 Phenomenology

This section contains the phenomenology related to both the geometry and the final product processing.

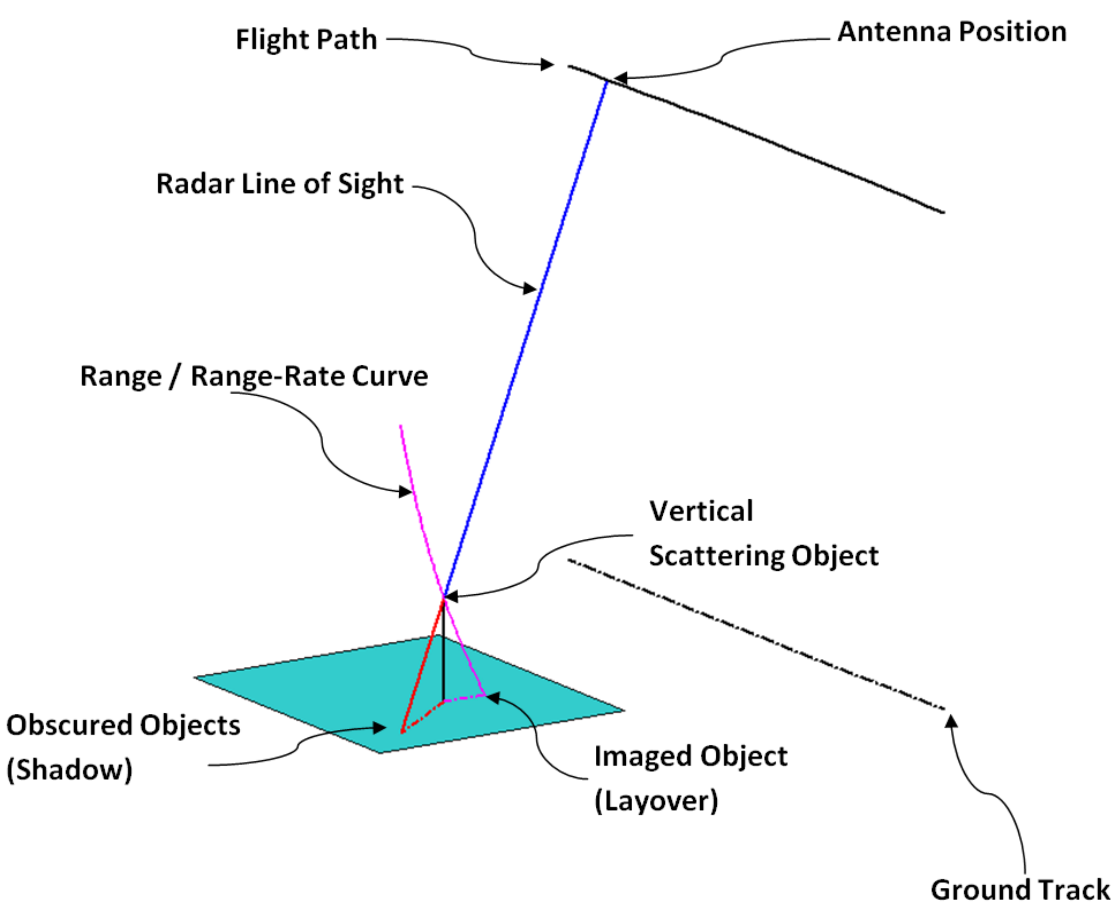


Figure 6-3 Phenomenology Diagram

6.5.1 Shadow

Shadow is an area in an image which is obscured by large vertical objects. The shadow vector derivation is shown below.

$$\boldsymbol{S} = \widehat{\boldsymbol{Z}}_g - \frac{\widehat{\boldsymbol{X}}_s}{\widehat{\boldsymbol{X}}_s \cdot \widehat{\boldsymbol{Z}}_g}$$

$$\boldsymbol{S}' = \boldsymbol{S} - \frac{\boldsymbol{S} \cdot \hat{\boldsymbol{z}}}{\widehat{\boldsymbol{Z}}_s \cdot \hat{\boldsymbol{z}}} \widehat{\boldsymbol{Z}}_s$$

$$\theta_S = \tan^{-1}(\boldsymbol{c} \cdot \boldsymbol{S}', \boldsymbol{r} \cdot \boldsymbol{S}')$$

The shadow magnitude is then

$$S = \sqrt{\boldsymbol{S}' \cdot \boldsymbol{S}'}$$

6.5.2 Layover

Layover is the phenomenon in which vertical objects appear as ground objects with same range/range-rate. The layover vector derivation is shown below. It assumes that the range/range-rate circles are nominally linear over the length of the ambiguity.

$$\boldsymbol{L} = \hat{\boldsymbol{z}} - \frac{\widehat{\boldsymbol{Z}}_s}{\widehat{\boldsymbol{Z}}_s \cdot \hat{\boldsymbol{z}}}$$

$$\theta_L = \tan^{-1}(\boldsymbol{c} \cdot \boldsymbol{L}, \boldsymbol{r} \cdot \boldsymbol{L})$$

The layover magnitude is then

$$L = \sqrt{\boldsymbol{L} \cdot \boldsymbol{L}}$$

6.5.3 North Direction

The north direction points toward the north pole and is defined at the scene center point. The computation is shown below.

$$\widehat{\boldsymbol{N}} = \begin{bmatrix} -\sin\varphi\cos\lambda \\ -\sin\varphi\sin\lambda \\ \cos\varphi \end{bmatrix}$$

$$\boldsymbol{N}' = \widehat{\boldsymbol{N}} - \frac{\widehat{\boldsymbol{N}} \cdot \hat{\boldsymbol{z}}}{\widehat{\boldsymbol{Z}}_s \cdot \hat{\boldsymbol{z}}} \widehat{\boldsymbol{Z}}_s$$

$$\theta_N = \tan^{-1}(\boldsymbol{c} \cdot \boldsymbol{N}', \boldsymbol{r} \cdot \boldsymbol{N}')$$

6.5.4 Up Direction

The up direction is also referred to as $\widehat{\boldsymbol{Z}}_{\mathbf{g}}$, which is normal to an earth geodetic tangent plane. This specific plane is a plane passing through the scene center point and is parallel to a plane that is tangent to the WGS-84 ellipsoid at the geodetic values φ, λ, for the scene center point. The computation is shown below.

$$\widehat{\boldsymbol{Z}}_{\boldsymbol{g}} = \boldsymbol{U} = \begin{bmatrix} \cos\varphi\cos\lambda \\ \cos\varphi\sin\lambda \\ \sin\varphi \end{bmatrix}$$

6.5.5 Multi-Path

Multi-path, or multi-bounce, is a phenomenon in which energy from a single scatter returns to the radar via more than one path. This is a range dependent phenomenon and results in a nominally constant direction in the image plane.

The multi-path vector is computed below

$$\boldsymbol{M} = \widehat{\boldsymbol{X}}_{\boldsymbol{s}} - \frac{\widehat{\boldsymbol{X}}_{\boldsymbol{s}} \cdot \widehat{\boldsymbol{z}}}{\widehat{\boldsymbol{Z}}_{\boldsymbol{s}} \cdot \widehat{\boldsymbol{z}}} \widehat{\boldsymbol{Z}}_{\boldsymbol{s}}$$

The multi-path angle is computed in the standard fashion.

$$\theta_M = \tan^{-1}(\boldsymbol{c} \cdot \boldsymbol{M}, \boldsymbol{r} \cdot \boldsymbol{M})$$

6.5.6 Ground Track (Image Track) Angle

The image track angle is the projection of the velocity vector into the image plane.

$$\boldsymbol{T} = \boldsymbol{V}_{\boldsymbol{a}} - (\boldsymbol{V}_{\boldsymbol{a}} \cdot \widehat{\boldsymbol{z}})\widehat{\boldsymbol{z}}$$

$$\theta_T = \tan^{-1}(\boldsymbol{c} \cdot \boldsymbol{T}, \boldsymbol{r} \cdot \boldsymbol{T})$$

7 Product Resolution

Ground plane resolutions are defined in the row and column directions and can be computed from the slant plane resolutions and collection geometry.

Ground plane resolutions are defined in the row and column directions and can be computed from the slant plane resolutions and collection geometry.

$$\rho_{r,g} = \sqrt{\kappa_{r1}\rho_{r,s}^2 + \kappa_{r2}\rho_{c,s}^2}$$

$$\rho_{c,g} = \sqrt{\kappa_{c1}\rho_{r,s}^2 + \kappa_{c2}\rho_{c,s}^2}$$

Where

$$\kappa_{r1} = \cos^2\theta_r \sec^2\psi + (\sin^2\theta_r \tan\psi \tan\eta - \sin 2\theta_r \sec\psi)\tan\psi \tan\eta$$

$$\kappa_{r2} = \sin^2\theta_r \sec^2\eta$$

$$\kappa_{c1} = (\sin^2\theta_r \sec\psi - \sin 2\theta_r \tan\psi \tan\eta)\sec\psi + \cos^2\theta_r \tan^2\psi \tan^2\eta$$

$$\kappa_{c2} = \cos^2\theta_r \sec^2\eta$$

The rotation angle is defined below:

$$\boldsymbol{X_g} = \boldsymbol{\widehat{X}_s} - \left(\boldsymbol{\widehat{X}_s} \cdot \boldsymbol{\hat{z}}\right)\boldsymbol{\hat{z}}$$

$$\theta_r = -\tan^{-1}\left(\boldsymbol{c} \cdot \boldsymbol{X_g}, \boldsymbol{r} \cdot \boldsymbol{X_g}\right)$$

8 Acronym List

Table 8-1 Acronyms

Acronym	Definition
ARP	Aperture Reference Point
BE	Basic Encyclopedia
CCW	Counter-clockwise
DRA	Dynamic Range Adjustment
D&I	Design and Implementation Description Document
ECEF	Earth Centered, Earth Fixed Coordinate
EGTP	Earth Geodetic Tangent Plane
ELT	Electronic Light Table
GGD	Geodetic Gridded Display
IERS	International Earth Rotation Service
PGD	Planar Gridded Display
ROI	Region of Interest
RRDS	Reduced Resolution Datasets
SAR	Synthetic Aperture Radar
SFA	Simple Feature Access
SICD	Sensor Independent Complex Data
SIDD	Sensor Independent Derived Data
SIPS	Softcopy Image Processing Standard
UTC	Universal Time Coordinate
XML	eXtensible Markup Language