NGA.STND.0017_3.0
2010-11-15

NGA STANDARDIZATION DOCUMENT

Community Sensor Model (CSM)
Technical Requirements Document (TRD)
(2010-11-15)

Version 3.0

NATIONAL CENTER FOR GEOSPATIAL INTELLIGENCE STANDARDS

Revision History

Revision	Date	Status	Description
3.0	29 Sep 2010	Modified	Adds CCB approved changes from 31 October 2006 through 17 September 2010 Remove MTI Target Model and MTI TargetPlugin
3.0	15 November 2010	Peer Reviewed	Adds changes from review
3.0	30 November 2010	Approved	Put under configuration management

The Revision History Table is used to record the history of changes to this document, as well as the current status of a revision in the review-approval cycle. All entries in this table are tagged with one of the following statuses:

Status	Description
New	New, not approved
Modified	Modified – needs to be peer reviewed.
Peer Reviewed	Peer Reviewed – needs to be (re-) approved for release
Released	Approved and moved put under configuration

Summary of Changes between previous Version 2.A and 3.0

This version of the CSM TRD has been enhanced and benefits from a general evolution and refinement over the period that the CSM TRD Version 2.A has been the Mandated baseline. The text has been clarified and corrections have been made. New calls have been added to optimize use of the Sensor Model and to incorporate functionality that was not present in Version 2.A.

Among the major enhancements in CSM TRD Version 3.0 is that it allows for expandability and backwards compatibility between future versions of the TRD. The means used to gain this capability is the addition of new classes to the API without altering existing class definitions. This is done by either defining a completely new base class or a new derived class of an existing base class. The addition of a new base class will allow for the creation of a completely new model. Further, this approach allows for easy expandability of the standard without affecting the current models. Adding new capability and still providing backward compatibility to an existing model is done with derived classes. For example, adding a new method to a model and making modifications to existing methods is done this way. Additional software calls were added to the CSM TRD API Version 3.0 to assist with this process (e.g. getVersion).

A new section on Compiling and Linking with the CSMPlugin Base Classes has been added to assist the Sensor Model and SET developer.

Previous integration of Sensor Models with SETs that use multithreaded code has caused problems. A new section on Thread Safety has been added to identify and rectify this problem.

All of the “TSM” references have been removed and converted to “CSM”. This will remove any confusion with the legacy Air Force Tactical Sensor Model program that was the forerunner of the Community Sensor Model standardization undertaken by NGA within the Geospatial Intelligence Standards Working Group (GWG), Community Sensor Model Working Group (CSMWG).

The ‘SystematicErrorCorrection’ functionality has been renamed ‘GeometricCorrection’ to better reflect the true nature of its purpose. This functionality allows for the ability to turn on/off geometric corrections. Along with the clarified naming, the description and use of this functionality has been clearly specified in the new document.

The ‘getValidAltitudeRange’ command has been replaced by the better-defined and more useful commands of ‘getValidHeightRange’ and ‘getValidImageRange’.

The explanation of Parameter Types has been greatly expanded along with the addition of new functionality to get the value of and set the value of both the original and the current parameter types. Previously, only the current parameter type was available.

A new section discussing the Covariance Model has been added. Previously, there was one call to get the Covariance Model that was not sufficiently defined. The call returned only a simple pointer address. The content of the returned pointer address was not well enough defined which caused confusion. This call has been augmented with a much more robust and better-defined set of two calls, ‘getCurrentCrossCovarianceMatrix’ and ‘getOriginalCrossCovarianceMatrix’.

The valuable information learned from and during the acquisition of new commercial sensor models has been incorporated into the example Statement Of Objectives in Appendix E. Appendix E has been extensively revised to better assist the government in its procurement and contracting of sensor model development.

Table of Contents

List of Figures

List of Tables

1 Introduction

The Community Sensor Model (CSM) will provide the Department of Defense (DoD) and the Intelligence Community (IC) with the capability to create and maintain a standard digital library of models, algorithms and software for developing, testing, and evaluating a collection of current and future sensor models. The models support Sensor Exploitation Tools (SETs) and other application tools that require a precise understanding of the image (data) and ground coordinate relationships. The CSMs are dynamically linked (or loaded) libraries that do not require re-compilation of the SET. Models may be added or removed from the SET without impact on the SET or other models. This capability will be used to accurately map a pixel (e.g., target location) on an image to a geo-referenced coordinate and provide rigorous error estimates.

For questions and/or to provide beneficial comments (recommendations, additions, and/or deletions) or other pertinent data which may be of use in improving this document, please use the following website to contact the Community Sensor Model Working Group: http://csmwg@seicorp.com.

1.1 Identification

The Community Sensor Model provides a precise understanding of the image and ground coordinate relationship for a specific sensor or sensor mode. The main CSM functions are the transformations between image space to ground space (ground to image, image to ground). These transformations and associated capabilities provide inputs used by the SETs to complete other photogrammetric and exploitation operations.

The TRD provides the technical requirements for the CSM. A CSM is a dynamically linked (or loaded) software library that supports, but does not perform, other photogrammetric operations on images. Underlying a CSM is a mathematical model described by equations, an algorithm, and a process that defines a coordinate transformation from a sensor’s image space (2-dimensional) to ground space (3-dimensional). The CSM can be based on the phenomenology, physics, and geometry of the image sensing/formation process--modeling the imaging ray from the sensor, through the optics (or antenna), down to the ground with a set of rigorous equations. It can correct for system or sensor specific aberrations, if needed.

The CSM is intended for applications where the earth's surface is imaged; nevertheless, it may also accommodate the imaging of features or phenomena above or below the earth's surface. Although CSM might appropriately be used with underwater sensors, such applications have not yet been sufficiently explored to recommend this usage.

The SET operations may include mensuration, feature projection, extraction, registration, uncertainty propagation and any other operation that uses the functions provided by the CSM within the confines of the interface definition.

1.2 Philosophy

An objective of these CSM requirements is to standardize the coordinate transformations of Community imagery--ensuring consistent, accurate coordinates are provided to the warfighter. A consideration in the CSM development is to minimize changes to existing

SETs and other application tools that require a precise understanding of the image and ground coordinate relationships.

Initially, the SETs may require modifications to access the functionality of the CSMs in a sensor independent manner, but will not require additional changes as more CSMs are produced and used by the SETs.

The CSM's API (Appendix C) is a standardized method for communicating between the CSM and the SETs. The CSM API appendix defines a library of functions that can be dynamically loaded by the SET.

The acquisition strategy is, as new sensors are developed or existing sensors are revised, the sensor developer must deliver a sensor model built in accordance with the requirements of the TRD. These sensor models will be dynamically linked (or loaded) libraries that do not require re-compilation of the SET.

1.3 Architecture/Product Perspective

Figure 1 - CSM Context Diagram displays the perspective of the CSM in its operational environment—specifically, showing its relationship to the SET via the API. The figure shows example data that may be passed between a given CSM and the SET and example functions. The figure is not all-encompassing.

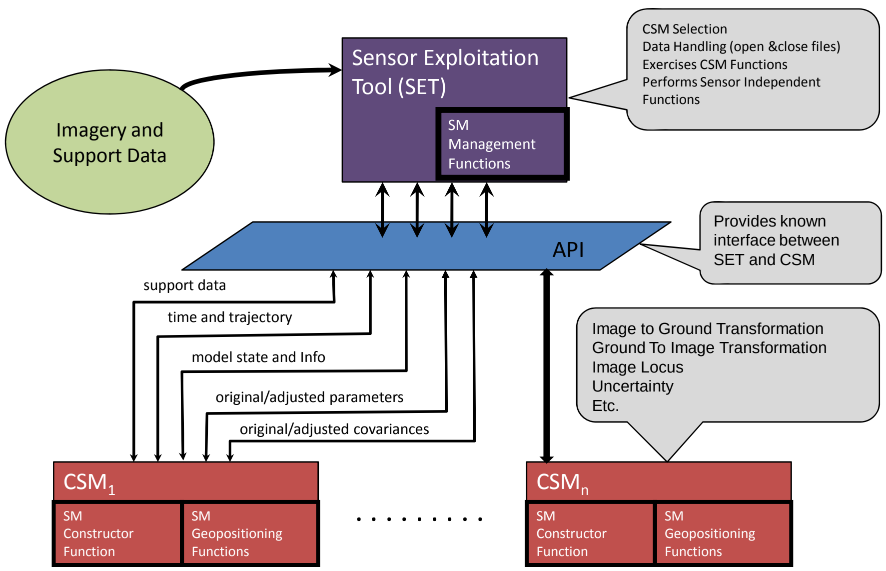

Figure 1 - CSM Context Diagram

Below the API, the CSM has two distinct sets of functions. The CSM constructor functions include those functions required to instantiate the sensor model and prepare it to respond to inputs from the SET. The CSM geopositioning functions perform the image to ground and ground to image transformations. Associated functions support these transformations and provide additional information used by the SET to perform its exploitation functions. The CSM provides a list of specific parameters, their values and uncertainties (variances and covariances) and a means for adjusting these parameters to obtain more accurate solutions. The CSM also integrates information regarding the time of collection and the trajectory.

Above the API, the SET understands the local environment and performs data handling functions (i.e. opens/closes files, opens/closes data streams, etc.). The SET uses the CSM constructor functions to create the required CSM. And the SET exercises the transformation functions between the ground and image spaces to exploit the imagery. The SET can also use this information to perform other sensor independent functions such as mensuration, registration, feature extraction, etc. Using the associated CSM functions, the SET adjusts selected CSM parameters to obtain a more accurate solution.

Furthermore, the SET performs sensor model management functions as required. The SET also selects the appropriate CSM if more than one model is available for the imagery data in use.

The API provides the structure and definition required for the CSM and SET to communicate.

1.4 Sensor Types and Assumptions

1.4.1 Sensor Types

Several types of CSMs exist depending primarily on the method or phenomenology by which the imaging sensor collects an image. This effort is focused on 2-D imagery initially but with an extension to 3-D imagery in the future. Details on the following sensor types can be found in Appendix A. Note that Multi-Hyper/Ultra Spectral Imagery (MSI, HSI, USI), and infrared (IR) are special cases of EO. Moving target indicator (MTI) data is a capability of some special cases of EO and of some SAR sensors.

1. Electro Optic (EO/IR)
2. Synthetic Aperture Radar (SAR)
3. Video (EO/IR)

Sensor type and mode of collection is reported by a sensor model through the getSensorTypeAndMode API.

1.4.2 Sensor Model Assumptions

The Community sensor model provides transformations between image and ground spaces. The following assumptions are made:

1. The sensor model operates on a single image/frame.
2. The sensor model operates on multi-band data as single frames, which treats each band as a “single” frame and the model addresses one frame at a time.
3. Video frames are handled in the same manner; each video frame is a single frame, addressed by the sensor model as a single frame.
4. The CSM does not perform any file input/output operations. The SET will perform all required file input/output functions.

1.5 Growth

The initial effort includes the development of CSMs for EO, IR and SAR sensors. Follow-on efforts may add MSI and HSI as well as LIDAR and other sensors including support for MTI capabilities. Other improvements may include identifying common sensor model functions for migration to a primary CSM SET API.

2 References and Applicable Documents

If a requirement in this TRD is in conflict with a referenced document, the contents of this TRD shall have precedence with regard to the CSM implementation. All specification references, including military, in this or any other CSM document are for guidance only.

2.1 Government Documents

Table 1 – Applicable Government Documents

Document No.	Title
MIL-STD-2500A	National Imagery Transmission Format Version 2.0 for the National Imagery Transmission Format Standard NOTE: Version 2.0 is for legacy data and has been updated to Version 2.1.
MIL-STD-2500C	National Imagery Transmission Format Version 2.1 for the National Imagery Transmission Format Standard
NATO STANAG 4545	NATO Secondary Imagery Format (NSIF) Edition 1 – 27 November 1998
STDI-0001 V1.3	National Support Data Extensions (SDE) Version 1.3 for the National Imagery Transmission Format (NITF)
STDI-0002 V3.0	Compendium of Controlled Extensions (CE) for the National Imagery Transmission Format (NITF), Version 3.0
N0105-98	NITFS Standards Compliance and Interoperability Certification Test and Evaluation Program Plan
TR 8350.2	NGA Technical Report 8350.2, DoD World Geodetic System 1984 – Its Definition and Relationship with Local Geodetic Systems
DoDI 5000.61	DoD Modeling and Simulation Verification, Validation, and Accreditation
NIST Special Publication 811	NIST Guide for the Use of the International System of Units (SI)
NUG-B	USIGS Glossary Revision B
	Applicable Platform Developer Documents ORDs
	Applicable Application Developer Documents APIs
DCID 6/3 Manual	Director Central Intelligence Directive (DCID) 6/3 - Protecting Sensitive Compartmented Information Within Information Systems
JDCSISS	Community DODISS / Cryptologic SCI Information Systems Security Standards
CSM TRD Appendix C	Community Sensor Model Application Program Interface
BIIF Profile NSIF01.01	ISO/IEC BIIF Profile NSIF01.01, NATO Secondary Imagery Format, Version 01.01

	Geopositioning Accuracy Validation Working Group Validation Plan, 30 May 2002
Air Force Pamphlet 14-210 Intelligence	USAF Intelligence Targeting Guide – 1 February 1998

2.2 Commercial Standards

Table 2 – Commercial Standards

Document No.	Title
ANSI IEEE 754-1985	Floating Point Arithmetic
ISO 8601:2000	ISO 8601:2000 (international standard for date representation)
ISO/IEC 14882:1998	ISO/IEC 14882:1998: Programming Languages - C++
ISO/IEC 14882:2003(E)	ISO/IEC 14882:2003(E): Programming Languages - C++

2.3 System Specifications and ICDs

Table 3 – Applicable System Specifications and ICDs

Document No.	Title
Not Applicable	CSM Design document
Not Applicable	SOW
Not Applicable	Sensor TEST plan or Test Plan Report
NATO STANAG 4607	NATO STANAG 4607- **NATO** Ground Moving Target Indicator Format (GMTIF)

3 Community Sensor Model Requirements

3.1 Application Program Interface (API)

The CSM shall be implemented in accordance with the API (Appendix C), which is the interface between the CSM and the SET. The CSM API document defines in detail the methods and their syntax for accessing model information and performing basic photogrammetry operations.

3.2 Community Sensor Model (CSM)

The CSM shall be a dynamically linked/shared library that does not require re-compilation of the SET. The CSM shall be added or removed from the SET without impact on the SET or other models.

3.3 Measurement Units

With the exception of image support data and any exceptions in the API, the CSM shall utilize standard metric units (base and derived) in accordance with the International Systems of Units (SI). Note that this requirement only applies to values passed across the interface between the CSM and SET. It does not mandate the units that may be reported to the SET user or used by the CSM internally.

3.4 Photogrammetry

3.4.1 Single Frame Operation

Each CSM operates as a single sensor model per image. For most sensor models, this assumes a single image/frame/band per instantiation of the sensor model. It is the responsibility of the SET to structure the imagery and support data accordingly. For example, a SET may need to subdivide multi-band data into multiple sets treating each band as a single “frame” and instantiating multiple sensor models. Video data may be handled in the same manner; the SET may extract a single video frame to instantiate a sensor model.

3.4.2 Ground to Image

The CSM shall transform a 3-D point in ground space to a 2-D point in image space.

3.4.3 Image to Ground

The CSM shall transform a 2-D point in image space to a 3-D point in ground space for a given elevation.

3.4.4 Imaging Locus

The CSM shall compute an imaging locus (in ground space coordinates) from a 2-D image point. There is a one-to-one mapping going from ground space to image space. However, going from image to ground space the mapping is one to many. Each pixel in image space corresponds to a line of sight that traverses a set of ground points. That set of ground points is termed the imaging locus.

3.5 Coordinate System

3.5.1 Ground Space Coordinate System

The CSM shall use a rectangular Earth Centered Earth Fixed (ECEF) coordinate frame referenced to WGS-84.

3.5.2 Image Space Coordinate System

Any point in an image can be described by two coordinates, the line (or row) and the sample (or column). The origin of the coordinate system is taken to be at the upper left corner of the upper left pixel. The line coordinate is positive in the downward direction on the image, and the sample coordinate is positive to the right. The pixel at the origin will have the coordinates of (0,0).

Image coordinates are measured in units of pixels. Only coordinates referenced to the full image resolution are used in the Sensor Model interface.

The image coordinates at the center of any pixel will have a fractional part of 0.5.

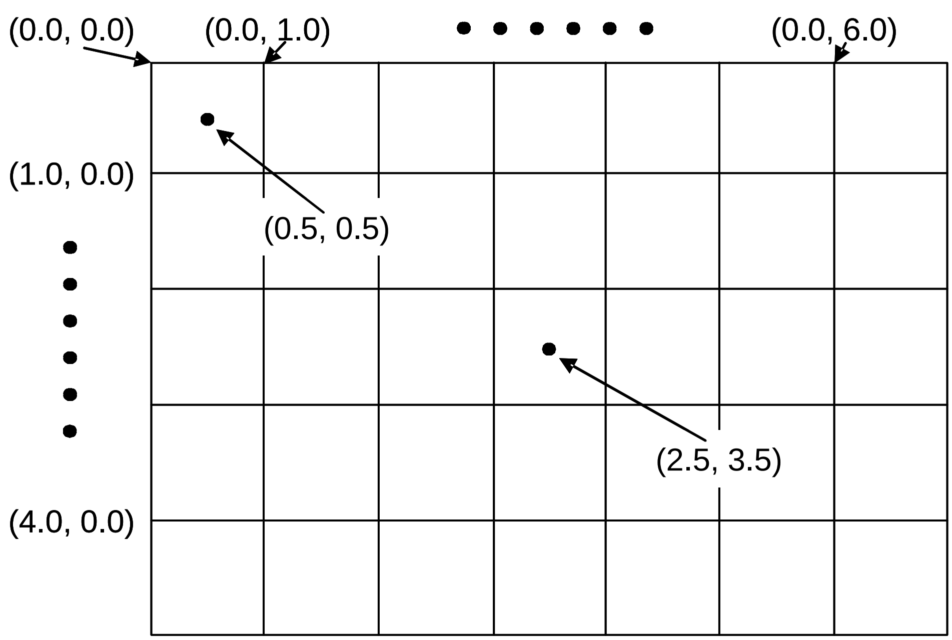

Figure 2 - Image Coordinate System

3.6 Time

The CSM shall provide image collection time in accordance with the API (Appendix C). Time shall be provided in Coordinated Universal Time (UTC). The required granularity of this data (e.g., once per frame, once per line, etc.) and its association with the image depend on the sensor mode as described in Appendix A. The time/date format shall comply with ISO 8601:2000.

3.7 Trajectory Data

The CSM shall provide the sensor position and velocity for sensor in accordance with the Appendix C.

3.7.1 Position

The 3-D sensor position shall be provided as defined in paragraph 3.5.1.

3.7.2 Velocity Vector

The 3-D sensor velocity vector shall be provided in meters/second units relative to the coordinate system in paragraph 3.5.1.

3.8 Model Identification

The CSM shall provide sensor model type and identification. Since multiple CSMs may be applicable to a given image, the CSM shall provide information to allow the SET or SET operator to select the appropriate model, if needed.

3.9 Model State

The CSM shall provide information on the state of the model. The state of a sensor model is the set of data needed to instantiate the sensor model.

The saved sensor model state data is used to instantiate a model to some condition other than the original state. This allows the operator to save changes made to the model, such as registration with known ground points in the image to improve accuracy, and then restart the model with the changes.

3.10 Model Parameters

3.10.1 Parameter Availability

The CSM shall provide information regarding the availability of model parameters as defined in the API.

3.10.2 Parameter Adjustability

Selected CSM sensor model parameters shall be adjustable in order to refine the reported ground coordinate corresponding to a given image coordinate, i.e. allow registration type operations.

1. The great majority of all possible image geometry error must be removable through the use of the adjustable parameters (Goal).

2. The uncertainty ascribed to the collection of all adjustable parameters along with the reported unmodeled error shall properly represent the potential image geometry error associated with the aggregate of all possible sensor model error sources (including those that have no corresponding adjustment parameter). This requirement is only in effect when all of the geometric corrections match the configuration of the delivered sensor model.

3.10.3 Parameter Initialization

It is strongly recommended that adjustable parameters be defined in such a manner that their initial value be zero. This can be very helpful to numerical algorithms that work with adjustable parameters.

3.10.4 Parameter Format

The CSM shall transfer sensor model parameters as defined in the API appendix.

3.11 Uncertainty Propagation

3.11.1 Covariance Availability

The CSM shall provide uncertainty estimates of the adjustable model parameters in the form of error covariances.

3.11.2 Covariance Adjustability

The CSM shall accept adjusted covariance values to optimize performance of photogrammetric operations. The CSM shall provide access to these updated values.

3.12 Partial Derivative Computation

The CSM shall compute partial derivatives of the image position with respect to the ground coordinates at the given ground position. The CSM shall compute partial derivatives of the image position with respect to the given sensor parameter at the given ground position.

3.13 Support Data Ingest

The CSM shall be capable of ingesting necessary support data (including SDEs) delivered by the sensor through the SET in accordance with the Appendix C.

3.13.1 Image Support Data (ISD) Reading Methods

The CSM shall support all of the suitable ingest methods described in Appendix C that are technologically feasible. NITF compliant ISD shall support reading by Filename method, by Bytestream method and by NITF 2.0 and/or by NITF 2.1 method. Non-NITF ISD shall support reading by Filename method and by Bytestream method. For large size imagery Bytestream is the least preferred method. This is due to operating system memory constraints and the amount of physical RAM available. During the Bytestream read there are at least four tasks that are competing for the available RAM: the operating system, the SET, the Sensor Model and the dynamically allocated space required to hold the image until it has been completely streamed. A system resource allocation failure may occur if the streaming data exceeds the available RAM. To mitigate this problem try adding additional RAM and/or running on a larger bit size operating system (i.e. 64 bit vs. 32 bit).

3.14 Performance

3.14.1 Geometric Correction Switches

The ability to turn on/off geometric corrections is optional unless given a specific requirement by the Government. Using commercial spaceborne optical imagery as an example, these corrections would include but are not limited to: atmospheric refraction and velocity aberration. The sensor model vendor is at liberty to include others, as desired, to facilitate the verification and validation of their sensor model. It is the responsibility of the government acquisition program manager to specify which geometric corrections, if any, are required to be included by a sensor model builder.

Regardless of the current settings of the geometric correction switches, the sensor model builder shall implement error propagation such that it matches the configuration of the delivered sensor models. The sensor model developer may turn those geometric switches to the 'off' position when it is determined that the support data is so inaccurate that it either degrades the solution or provides no benefit. It should also be noted that the geometric correction switches and adjustable parameters are to be treated independently of each other.

SET developers must understand that turning any of the geometric correction switches to the off position can degrade the quality of the groundToImage(), imageToGround(), imageToProximateImagingLocus(), imageToRemoteImagingLocus(), computeGroundPartials(), and computeSensorPartials() methods. SET users may, for example, turn some switches off to reduce computation time; however, they are doing so at their own risk. If any of the geometric correction switches have been turned to the off position by the SET, then the error propagation associated with the groundToImage() and imageToGround() methods is not valid. Commercial SET developers should not provide the user an option, e.g. through a GUI, to turn these switches to the off position since they may not understand the consequences.

3.14.2 Accuracy

The CSM testing shall include verification of the sensor/sensor model combined accuracy compared to control points (e.g., ground survey points or other truth data). The CSM shall produce image positions that are consistent with the corresponding uncertainty estimates and surveyed ground space measurements. This requirement is only in effect when all of the geometric corrections match the configuration of the delivered sensor model.

3.14.3 Error Propagation Accuracy

The CSM shall produce uncertainty estimates that are consistent with the observed accuracy of the sensor/sensor model combination. (See paragraph above)

3.14.4 Error Estimation Calculations

The CSM shall support the calculation of Circular Error and Linear Error at a standard deviation (e.g., 50% or 90%) for all geopositioning scenarios.

Amplification: Provides the Warfighter with a statistical degree of confidence in the three-dimensional coordinate accuracy. Enables the Warfighter to determine coordinate accuracy for weapon type ,

3.14.5 Throughput

The CSM shall process at least 100 transformations per second (image to ground, ground to image or imaging locus) once the CSM is initialized, assuming the hardware/software configurations identified in Appendix B.

3.14.6 Latency

The elapsed time between a query and answer shall be less than 5 milli-seconds assuming the same hardware/software configuration described in Appendix B.

3.15 Software Design

The CSM shall be site installable and uninstallable without interference to other CSMs. The CSM shall be installable and be capable of being executed without requiring recompilation of the SET.

3.16 Software Environment

3.16.1 Programming Language

The CSM shall be coded using the ANSI standard C++ programming language.

Goal: One set of source code files that can be conditionally compiled for various UNIX or Windows operating systems.

3.16.2 Operating Systems

The CSM shall maintain independence from specific computer operating systems in order to insure maximum portability. The CSM shall be designed to support UNIX and/or Windows operating systems and compilers as specified in Appendix B. In the Microsoft Windows environment, the CSM shall be a dynamic link library (.dll) file—accessible with *LoadLibrary* and *GetProcAddress*. In the UNIX environment, the CSM shall be a shared object (.so) file—accessible with *dlopen* and *dlsym*. The CSM shall be designed to minimize the number of executable versions needed to support the range of development environments specified in Appendix B.

3.16.3 Thread Safety

Both the sensor model plugin and the sensor models they produce are required to run safely in a multi-threaded environment without significant performance degradation. A software element is considered thread safe if the outputs of method calls remain unchanged when run in a multi-threaded environment with multiple threads invoking the same object.

It is known that certain CSM methods will alter the results of other methods. For example, changing the value of an adjustable parameter will affect the image to ground and ground to image relationship. This is not considered a violation of thread safety. It is considered the responsibility of the SET to understand that these types of calls will affect all threads using the CSM model.

One significant area in which thread safety problems have been seen is in the numerical determination of partial derivatives. In this case, the sensor model is often momentarily altered for the computation and then returned to its original state. If another thread invokes the sensor model while it is in an altered state, unreliable outputs can result.

It is noted that one method to handle thread safety problems is through extensive use of mutex locking on the sensor model. However, if this use is made too extensive, performance in a multi-threaded environment may be impacted. It is not desirable to limit multi-threaded performance to single thread levels. The purpose of multi-threading is to increase performance.

Another design that can cause problems when running many multiple threads is having the plugin hold open file pointers. This can be problematic since many operating systems limit the number of files that can by simultaneously open in one application. If each sensor model retains open files for the life of the sensor model, this could potentially add up to exceed the system limit, especially on applications that may be opening many files itself.

3.17 Security

All SETs must comply with the requirements as defined in the Director Central Intelligence Directive (DCID) 6/3 and Joint DODIIS / Cryptologic SCI Information Systems Security Standards (JDCSISS) in the safeguarding of all classified elements both internally and/or externally accessed. Each CSM will be designed such that it does not impede the ability of a SET to meet these requirements.

3.18 Sensor Mode and Type

The CSM shall provide sensor mode and type in accordance with Appendix C.

4 Evaluation Methodology/Verification and Validation Process

4.1 Community Sensor Model/Sensor Exploitation Tool Evaluation Methodology

Table 4 provides an evaluation methodology for the CSM. Note that some table rows serve only as category headings in order to provide further clarification. Individual methods (demonstration, test, analysis, and inspection) are defined below.

Demonstration is defined as a method of verification denoting the qualitative determination of properties by observation. Demonstration is limited to a readily observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis. Demonstrations are used to indicate pass/fail conditions and to verify characteristics such as proper system response as a result of a specified input command, operational performance, human engineering features, service and access features

Test is defined as a method of verification wherein system performance is measured during or after the controlled application of real or simulated functional and/or environmental stimuli. Measurements of quantitative performance are often taken a sufficient number of times to provide a statistical level of confidence in the final result. System performance measurements may require the use of instrumentation or special test equipment to collect data for detailed analysis. The analysis of data derived from test is an integral part of the activity and may involve automated data reduction to produce the necessary results.

Analysis is defined as a method of verification wherein the item or component design is studied to determine if it meets specified requirements. Analysis includes the technical evaluation of drawings, software listings, equations, charts, graphs, diagrams, or representative data.

Inspection is defined as a visual method of verification that determines compliance with required characteristics without the use of special laboratory equipment, procedures, items or services. Inspection involves “looking at” an item or component, or reviewing descriptive documentation, and comparing the appropriate characteristic with a predetermined standard but does not require operation of the item.

Table 4 - Evaluation Methodology for TRD Section 3 Requirements

Req. #	Paragraph	Paragraph Title	Requirement	Function Cross Reference	Methodology Analysis	 Demon-stration	 Test	 Inspection
1	3.1	Application Program Interface (API)	CSM shall be implemented in accordance with the API, which is the interface between the CSM and the SET.		✓	✓		
2	3.1	Application Program Interface (API)	CSM API document defines in detail the methods and their syntax for accessing model information and performing basic photogrammetry operations.		✓			
3	3.2	CSM	The CSM shall by dynamically linked/shared library that does not require re-compilation of the SET.			✓		
4	3.2	CSM	The CSM shall be added or removed from the SET without impact on the SET or other models.			✓		
5	3.3	Measurement Units	CSM shall utilize standard metric units (base and derived) in accordance with the International Systems of Units (SI).		✓	✓	✓	
6	3.4	Photogrammetry	.					
7	3.4.1	Single Frame Operation	Each CSM operates as a single sensor model per ISD object. For most sensor models, this assumes a single			✓		

			image/frame/band per instantiation of the sensor model. It is the responsibility of the SET to structure the imagery and support data accordingly. For example, a SET may need to subdivide multi-band data into multiple sets treating each band as a single “frame” and instantiating multiple sensor models. Video data may be handled in the same manner; the SET may extract a single video frame to instantiate a sensor model.				
8	3.4.2	Ground To Image	CSM shall transform from a 3-D point in ground space to a 2-D point in image space.	groundToImage	✓	✓	✓
9	3.4.3	Image To Ground	CSM shall transform a 2-D point in image space to a 3-D point in ground space for a given elevation.	imageToGround ReferencePoint	✓		✓
10	3.4.4	Imaging Locus	CSM shall compute an imaging locus (in ground space coordinates) from a 2-D image point.	imageToProximateImagingLocus imageToRemoteImagingLocus	✓		✓
11	3.5	Coordinate System					
12	3.5.1	Ground Space Coordinate System	CSM shall use a rectangular Earth Centered Earth Fixed (ECEF) coordinate frame referenced to WGS-84.		✓		✓
13	3.5.2	Image Space	The CSM shall use an image coordinate system with the			✓	

		Coordinate System	origin set at the upper left corner of the upper left pixel. The coordinate system shall consist of the line (or row) component and the sample (or column) component. The line coordinate is positive in the downward direction and the sample coordinate is positive to the right. The pixel at the origin will have the coordinates of (0,0).			
14	3.6	Time	CSM shall provide image collection time in accordance with the API.	getReferenceDateAndTime getImageTime getImageIdentifier setImageIdentifier	✓	
15	3.6	Time	Time shall be provided in Coordinated Universal Time (UTC) ISO 8601.2000 Format.		✓	
16	3.7	Trajectory Data	CSM shall provide the sensor position, and velocity in accordance with the API.	getSensorPosition getSensorVelocity	✓	
17	3.7.1	Sensor Position	3-D sensor position shall be provided as defined in paragraph 3.5.1.	getSensorPosition		✓
18	Reserve					
19	3.7.3	Sensor Velocity Vector	3-D sensor velocity vector shall be provided in meters/second units relative to the coordinate system in paragraph 3.5.1.	getSensorVelocity		✓
20	3.8	Model Identification	CSM shall provide sensor	getPedigree	✓	

			model type and identification.					
21	3.8	Model Identification	CSM shall provide information to allow the SET or SET operator to select the appropriate model, if needed.	getPedigree getSensorIdentifier	✓	✓		
22	3.9	Model State	CSM shall provide information on the state of the Model. The state of a sensor model is the set of data needed to instantiate the sensor model.	getSensorModelState	✓	✓		
23	3.10	Model Parameters			N/A	N/A	N/A	N/A
24	3.10.1	Parameter Availability	CSM shall provide information regarding the availability of model parameters.	getPedigree getSensorIdentifier getTrajectoryIdentifier getCurrentParameterValue getParameterName getNumParameters getOriginalParameterValue getCurrentParameterType getOriginalParameterType		✓		
25	3.10.2	Parameter Adjustability	Selected CSM sensor model parameters shall be adjustable in order to refine the reported ground coordinate corresponding to a given image coordinate.	getcurrentParameterValue getnumParameters getoriginalParameterValue		✓		
26	3.10.3	Parameter Format	CSM shall transfer sensor model parameters using a structure defined in the API document.	getParameterName getcurrentParameterValue	✓	✓		
27	3.10.3	Parameter Format	Parameters shall comply with	getParameterName	✓	✓		

			NITF and SDEs.					
28	Reserved							
29	3.11	Uncertainty Propagation			N/A	N/A	N/A	N/A
30	3.11.1	Covariance Availability	CSM shall provide uncertainty estimates of the adjustable model parameters in the form of error covariances.	getoriginalParameterCovariance	✓			
31	3.11.2	Covariance Adjustability	CSM shall accept adjusted covariance values to optimize performance of photogrammetric operations.	getcurrentParameterValue getcurrentParameterCovariance getoriginalParameterCovariance		✓		
32	3.11.2	Covariance Adjustability	The CSM shall provide access to these updated values.	getCurrentParameterCovariance		✓		
33	3.12	Partial Derivative Computation	The CSM shall provide a method to compute partial derivatives of the image position with respect to the ground coordinates at the given ground position.	computeGroundPartials	✓	✓		
34	3.12	Partial Derivative Computation	[CSM shall provide a method to compute] partial derivatives of the image position with respect to the given sensor parameter at the given ground position.	computeSensorPartials	✓	✓		
35	3.13	Support Data Ingest	CSM shall be capable of ingesting necessary support data (including SDEs) delivered by the sensor through the SET.			✓		

36	3.13.1	ISD Reading Methods	The CSM shall support all of the suitable ingest methods described in Appendix C.	csm_ISDNITF21.h csm_ISDNITF20.h csm_ISDByteStream.h csm_ISDFilename.h		✓	
37	3.14	Performance					
38	3.14.1	Geometric Corrections	The ability to turn on/off geometric corrections is optional unless given a specific requirement by the Government. These corrections include but are not limited to: atmospheric refraction and velocity aberration. Regardless of the current settings of the geometric correction switches, the sensor model builder shall implement error propagation such that it matches the configuration of the delivered sensor models. The sensor model developer may turn those geometric switches to the ‘off’ position when it is determined that the support data is so inaccurate that it either degrades the solution or provides no benefit. It should also be noted that the geometric correction		✓		✓

			switches and adjustable parameters are to be treated independently of each other.		
39	3.14.2	Accuracy	The CSM shall include verification of the sensor/sensor model combined accuracy compared to control points (e.g. ground survey points or other truth data).	✓	✓
40	3.14.2	Accuracy	CSM results shall be consistent with the underlying math model (ground to image and image to ground).	✓	✓
41	3.14.2	Accuracy	CSM shall produce image positions that are consistent with the corresponding uncertainty estimates and surveyed ground space measurements.	✓	✓
42	3.14.3	Error Propagation Accuracy	CSM shall produce uncertainty estimates that are consistent with the actual sensor system. (See paragraph above.)	✓	
43	3.14.4	Error Estimate Calculations	The CSM shall support the calculation of Circular Error and Linear Error at a standard deviation (e.g. 50% or 90%) for all geopositioning scenarios.		✓
44	3.14.5	Throughput	CSM shall process at least 100 transformations per second (image to ground,		✓

			ground to image or imaging locus) once the CSM is initialized, assuming the following hardware/software configurations identified in Appendix B.				
45	3.14.6	Latency	Elapsed time between a query and answer shall be less than 5 milli-seconds assuming the same hardware/software configuration described in Appendix B.			✓	
46	3.15	Software Design	CSM shall be site installable and uninstallable without interference to other CSMs.	✓	✓		
47	3.15	Software Design	CSM shall be installable and be capable of being executed without requiring recompilation of the SET.		✓		
48	3.16	Software Environment					
49	3.16.1	Programming Language	CSM shall be coded using the ANSI standard C++ programming language.				✓
50	3.16.2	Operating Systems	CSM shall maintain independence from specific computer operating systems in order to insure maximum portability.	✓			
51	3.16.2	Operating Systems	CSM shall be designed to support UNIX and/or Windows operating systems and compilers as specified in Appendix B.		✓		

52	3.16.2	Operating Systems	In the Microsoft Windows environment, the CSM shall be a dynamic link library (.dll) file—accessible with *LoadLibrary* and *GetProcAddress*.	✓	✓	
53	3.16.2	Operating Systems	In the UNIX environment, the CSM shall be a shared object (.so) file—accessible with *dlopen* and *dlsym*.	✓	✓	
54	3.16.2	Operating Systems	CSM shall be designed to minimize the number of executable versions needed to support the range of development environments specified in Appendix B.	✓		
55	3.17	Security	All SETs must comply with the requirements as defined in the Director Central Intelligence Directive (DCID) 6/3 and Joint DODIIS / Cryptologic SCI Information Systems Security Standards (JDCSISS) in the safeguarding of all classified elements both internally and/or externally accessed. Each CSM will be designed such that it does not impede the ability of a SET to meet these requirements.	✓	✓	✓

4.2 Verification/Validation Process: Community Sensor Model/Sensor Exploitation Tool

This section of the TRD describes and identifies the top-level processes for the verification and validation of the CSM and the SET that uses the CSM(s). The primary emphasis here being the generation of accurate coordinates, including error propagation estimates.

For the purpose of this document, DoDI 5000.61 defines verification and validation as:

Verification: "The process of determining that a model implementation accurately represents the developer's conceptual description and specifications."

Validation: "The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model."

Verification and validation differ in that the developer of the CSM and SET performs the "in-house" verification, while an independent third party performs validation. Validation and verification also differ in the amount of test points required. For example, the verification of a CSM (for a given CSM and exploitation scenario) requires on the order of 30-50 test points while validation of the SET[1] requires on the order of 250 test points.

The four steps in verification and validation of the CSM and SET are:

A) CSM Verification:

- CSM Developer will write the verification test plan (reference the CSM TRD for requirements)
- CSM developer will conduct the verification testing and report the results to the CSM sponsor
- CSM developer may use either synthetic imagery data or actual imagery in the verification test
- Verification of the CSM will demonstrate Ground-to-Image and Image-to-Ground consistency
- The CSM verification is governed by the CSM TRD
- Outputs include the CSM, associated deliverables and verification test results

B) CSM Validation:

- A third party will perform CSM validation
- The third party will use a standard process[2] for CSM validation

---

[1] Reference Geopositioning Accuracy Validation Working Group Validation Plan, 30 May 2002 for details on the validation process (e.g., entry and exit criteria for the SET).

[2] The process (e.g., source of elevation, number of test points, etc.) for conducting the validation testing will be documented in NGA's emerging effort "Standard Process for Geopositioning Accuracy". Until such time as the Standard Process for Geopositioning Accuracy is adopted as the guiding document for CSM validation, it is recommended that all validation procedures and test plan be vetted with the Geopositioning Accuracy Validation Working Group.

- The third party may use synthetic imagery data in addition to actual imagery in the validation process; this is in recognition of the difficulty in obtaining actual or operational imagery
- The third party will make the results of CSM validation testing available for peer review by the Geopositioning Accuracy Validation Working Group (GAVWG)
- Validation of the CSM will demonstrate Ground-to-Image and Image-to-Ground consistency
- Outputs include the CSM, associated deliverables and validation test results

C) SET (via CSM) Verification:

- SET developer will write the verification test plan
- SET developer will conduct the verification testing and report the results to the SET sponsor
- SET developer will use either synthetic imagery data, actual imagery, or a combination of the two in the verification test
- Verification of the SET will demonstrate that the SET can generate accurate coordinates, including error propagation estimates (i.e., known levels of confidence)
- Outputs include the SET verification test results

D) SET (via CSM) Validation:

- The GAVWG will perform SET validation
- The GAVWG will use the process outlined in the GAVWG Validation Plan for SET validation
- The GAVWG will use imagery in the validation process
- The GAVWG will make the results of SET validation testing available for peer review by the SET developer and sponsor
- Validation of the SET will demonstrate that the SET can generate accurate coordinates, including error propagation estimates (i.e., known levels of confidence)
- Outputs include a validation message summarizing the results of the SET validation testing to the appropriate community

In support of the CSM and SET verification and validation process, it will be necessary to establish the imagery requirements (e.g., amount, collection geometries, etc.) and all of the techniques employed by the SET to generate coordinates.

The imagery collection of surveyed points is crucial to the verification and validation process. Without imagery of surveyed controlled points the CSM and SET cannot be validated. Therefore, it is desirable to have both the verification and validation testers and their sponsor(s) work with the organization requesting validation to ensure that the appropriate quantity of controlled imagery is collected for both verification and validation testing.

NGA and AFRL have established a “Golden Imagery Set” to support the verification and validation testing. The “Golden Imagery Set” will be accessible to both verification and validation testers.

5 Acronym List

Table 5 - Acronym List

Acronym	Definition
2D	Two-dimensional
3D	Three-dimensional
AFRL	Air Force Research Laboratory
AIS	Automated Information Systems
ANSI	American National Standards Institute
API	Application Program Interface
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
CCD	Charged Couple Device
CE	Circular Error
CGM	Computer Graphics Metafile
COE	Common Operating Environment
CONOPS	Concept of Operations
CSD	Covariance Support Data
CSW	Conventional Stand-off Weapon
DCID	Directory of Central Intelligence Directive
DII	Defense Information Infrastructure
DODI	Department of Defense Instruction
DODIIS	Department of Defense Index of Specifications & Standards
ECEF	Earth Centered/Earth Fixed
EM	Electromagnetic
EO	Electro-Optical
FOV	Field of View
GAVWG	Geopositioning Accuracy Validation Working Group
GEO-TIFF	Geo-referenced Tagged Image File Format
GFI	Government Furnished Information
GMT	Greenwich Mean Time
GRD	Ground Resolved Distance
HAE	Height Above Ellipsoid
HRR	High Range Resolution
HSI	Hyperspectral Imagery
ICD	Interface Control Document

Acronym	Definition
ISD	Image Support Data
I/O	Input and output
IR	Infrared
JCS	Journal of Computer Security
JDCSISS	Community DoDISS/Cryptologic SCI Information Systems
KVP	Key Value Pairs
LADAR	Laser Radar or Laser Detection & Ranging
LE	Linear Error
LGB	Laser Guided Bomb
LIDAR	Light Detection and Ranging
LWIR	Long Wave Infrared
MCP	Motion Compensation Point
MIL-STD	Military Standard
MKS	Meter-Kilogram-Second
MPEG	Moving Pictures Export Group
MSI	Multispectral Imagery
MTI	Moving Target Indicator
MWIR	Mid-wave Infrared
NGA	National Geospatial-Intelligence Agency
NIR	Near Infra-red
NIST	National Institute of Standards and Technology
NITF	National Imagery Transmission Format
ORD	Operational Requirements Document
OS	Operating System
RPC	Rapid Positioning Capability
SAR	Synthetic Aperture Radar
SCI	Sensitive Compartmented Information
SDE	Support Data Extension
SET	Sensor Exploitation Tool
SM	Sensor Model
SOO	Statement of Objectives
SWIR	Short Wave Infrared
SYERS	Senior Year Electro-Optical Reconnaissance System

Acronym	Definition
TBD	To Be Determined
TM	Thematic Mapper
TR	Technical Report
TRD	Technical Requirements Document
CSM	Community Sensor Model
USI	Ultraspectral Imagery
UTC	Coordinated Universal Time
VIS	Visible
VNIR	Visible Near Infrared
WGS-84	World Geodetic System 1984 (Referenced Ellipsoid)
XML	Extensible Markup Language

APPENDIX A – Sensor Type and Mode Definitions

A Appendix A – Sensor Type and Mode Definitions

A.1 Sensor Types

A.1.1 Electro Optical (EO)

EO image sensing for reconnaissance classically refers to fine spatial resolution broadband or panchromatic ("black-and-white") sensing in the Visible (VIS) spectral region. It may also include sensing in the Near Infrared (NIR) (VIS+NIR=VNIR) and Short Wave Infrared (SWIR) spectral regions. It usually refers to Passive EO sensing in the "reflective" mode (senses the reflections of ambient light originating from natural sources). Passive EO imaging conventionally produces a 2-D "angle-angle" image taken down-looking, ahead-looking, or oblique. Examples of EO sensors are video cameras and night vision cameras that rely on the ambient light to illuminate the imaged scene. Active EO sensor types would add and utilize a continuous or pulsed illumination source.

A.1.2 Infrared (IR)

IR sensing generally refers to broadband sensing in the mid-wave infrared (MWIR) or the long-wave infrared (LWIR) "thermal" spectral regions. These regions are also commonly referred to by their wavelength spans – the "3-to-5 micron" and "8-to-12 micron" wavelength regions. These usually passive cameras sense the intrinsic thermal emissions from objects in an imaged scene. Imaging infrared systems are common today and are used to sense both surface temperature and emissivity (material) differences. Similar to passive reflective EO cameras, IR cameras also produce 2-D angle-angle imagery. Active IR imaging sensors would also add and utilize a continuous or pulsed irradiation source to enhance detectability.

Electro-optical (EO) sensors, in general terms, can include UV, VIS, VNIR and IR sensors. However, many in the operational community think of EO relative to VIS or VNIR cameras and "other EO" being IR cameras. This latter point is the basic term definition adopted for the purposes of the CSM.

A.1.3 Synthetic Aperture Radar (SAR)

SAR utilizes active electromagnetic (EM) sensing in the radar wavelengths to form an all-weather/day-night imaging capability. SAR transmits and detects a series of wideband radar pulses to first form a High Range Resolution (HRR) capability on a pulse-to-pulse basis. SAR coherent image formation processing then corrects for platform motion and synthesizes the azimuth resolution, thus forming a radar image of a scene. A classical SAR image is a 2-D "range-angle" image. Other types of SAR data can be used to create elevation data or SAR can be geometrically and radiometrically terrain corrected by being combined with elevation data. These other forms of SAR data may be appropriate for CSM models but such applications have not yet been investigated.

A.1.4 EO/IR Special Case – Multi/Hyper/Ultra Spectral Imaging Systems

These systems are EO and/or IR imaging sensors that use multiple spectral bands to sense the "color" in a scene. They form a broad class of MSI, HSI, and USI sensors distinguished mainly by the bandwidths of the various spectral bands sensed in a system and the number of bands. A color video camera is one of the simplest examples of a 3-band multispectral imaging sensor. Multispectral systems, such as the LANDSAT Thematic Mapper ™ satellite or the SYERS-2A Community sensor have a set of 2 to 10 spectral bands with bandwidths (e.g., filters) on the order of 10's of nanometers. Hyperspectral systems, such as the AVIRIS, have around 200 narrow spectral bands approaching 1 nanometer. More novel ultraspectral sensors possess very fine spectral bandwidths of less than 1 nanometer. Spectral imaging sensors couple both spectral and spatial properties for utility to detect, classify, and identify materials or targets of interest to a variety of users. Spectral imaging systems generally produce 2-D spatial imagery and a third dimension of wavelength bands – "angle-angle-wavelength" imagery. For example, the 3D structure is referred to as an HSI image datacube or hypercube. Products can be 2-D images integrated over wavelength, "true" or "false" color image composites using weightings of different bands, or specific spectral graphs depicting the spectral response of particular pixels or features extracted from the datacube. In general, the various bands of the spectral imaging sensor must be registered spatially to a 2-D image.

Thus, the basic sensor model for passive 2-D angle-angle imagery for EO, IR, and spectral imaging sensors is expected to have great similarity. Most of the separation of these models will be in how the actual sensor builds up the imagery (see Section A.2).

A.1.5 Motion Imagery - Video

MOTION Imagery is defined as imaging sensor / systems that generate sequential or continuous streaming images at specified temporal rates (normally expressed as frames per second), ***within a common field of regard***. MOTION Imagery is defined as nominally beginning at frame rates of 1 Hz (1 frame per second) or higher.

Within the major division of MOTION Imagery, the following domains are currently specified: Electro Optical (including Video and Television), Waveforms based on Radar Imaging, including Synthetic Aperture Radar (SAR), and Moving Target Indication (MTI) and Acoustic Waterfall.

Within the Motion Imagery Electro-Optical domain, specific definitions are given for Video and Television sub-domains:

Video is defined as Electro-Optical motion imagery technologies defined by standards developed by ISO, ITU, SMPTE, EBU, etc., reviewed, adopted and profiled for DoD/IC/USIGS applications by designated DoD/IC/USIGS standards bodies such as the GWG's Motion Imagery Standards Board (MISB).

Television is defined as Video formats and implementations defined by Government Transmission Regulations such as NTSC, PAL, SECAM, FCC 4th Report and Order; reviewed, adopted and profiled for DoD/IC/USIGS applications by designated DoD/IC/USIGS standards bodies such as the VWG.

A.2 Sensor Imaging Modes

Once the CSMs are distinguished with respect to basic sensor type, the method by which imagery is formed is an important factor to distinguish different types of CSMs. In many cases, the sensor only can operate, by design, with one sensor mode (e.g., framing camera). In other cases, such as advanced SAR systems, different imaging modes (e.g., Spot, Strip, and Scan) may be selectable. Generally, all of these methods affect both the imaging-to-scene geometry and the method that the sensor temporally builds up an image scene. Within each sensor mode, various parameters may be selectable, such as the optical zoom or the SAR synthesized resolution. These selections would be made to trade-off various operational parameters such as field of view (FOV) and Ground Resolved Distance (GRD).

Table 6 - Imaging Sensor System Types and Modes

Sensor Type/Phenomenology	Sensing Geometry	Sensor Imaging Mode
EO IR Spectral Imaging	2D angle-angle	Frame Pushbroom Whiskbroom
SAR	2D range-angle	Spot Strip Mapping Scan Moving Target Indicator

A.2.1 Frame

A camera that acquires all of the data for an image essentially at one time and at a fixed exposure by using a two-dimensional detector such as film or a Charge-Coupled Device (CCD) array.

A.2.2 Pushbroom

A camera that acquires the data for an image at a sequence of times, one or more lines at a time. It relies upon the motion of the platform/sensor (e.g., along-track motion) to bring new lines of the scene into view as time passes. These cross-track lines would then be sensed rapidly in time with respect to the along-track motion and the image built-up line-by-line.

A.2.3 Whiskbroom

A camera that acquires the data for an image at a sequence of times, and not collecting a whole line at one time. It relies upon the platform/sensor motion for scanning in one direction (along-track), and may or may not have motion correction in the other scan direction. A whiskbroom camera might have a linear detector array that is aligned in the along-track direction and swept rapidly in the cross-track direction to acquire several samples of an image as the sensor travels a short distance. Whiskbroom scanning can also be used with framing cameras to greatly extend the field-of-view. One achieves the benefit of a fine-resolution framing camera, which is then rapidly scanned over a larger scene to build a larger image composed of frames sampled at sequential times.

A.2.4 Spot

Spot refers to the SAR spotlight imaging mode. Spot is the finest spatial resolution SAR mode, which is achieved by the operator and image formation processor fixing the Motion Compensation Point (MCP) at a particular spot on the ground as defined by a fixed range/time and zero Doppler shifted coordinate relative to the sensor system. As the system dwells about this point collecting radar data, all data is compensated to this point and the image is coherently formed. Often, the MCP is the central pixel in a SAR image or image subpatch.

A.2.5 Strip

Strip Mapping is a SAR imaging mode where the SAR antenna is fixed or steered at a specific squint (or azimuth) angle with respect to the platform's velocity. Typically, this is 90 degrees or broadside, but squinting to 20 degrees or better can be achieved by some systems. In the strip mapping mode, the MCP then is fixed in squint (azimuth) and depression angle forming a point on the ground that moves with the platform. Imagery is formed with respect to this MCP location, with the achieved SAR azimuth resolution limited by radar effective antenna aperture size. Long continuous SAR image strips can be formed in this mode, which is useful for radar mapping of any area.

A.2.6 Scan

Scanning is an advanced SAR imaging mode that controls the MCP in both the range and azimuth locations. As the platform moves, the MCP is controlled to image, within the SAR operating envelope and parameters, any path along the ground. The mode is like a flexible Strip Mapping mode in that a radar map is produced as the start to end MCP locations are accessed and the SAR imagery is formed. This mode allows an area not parallel to the along-tack motion to be imaged and mapped.

A.2.7 Moving Target Indicator

Moving Target Indicator is a mode in which a SAR sensor scans a coverage pattern in search for moving targets. The SAR antenna sweeps through the coverage pattern, possibly requiring multiple sweeps to cover the entire area. Moving targets are identified based on Doppler shift of the radar returns.

APPENDIX B

Hardware and Software Requirements

B Appendix B

B.1 Appendix B

Data Ingest Formats: NITF v 2.0 and/or v 2.1 (both SET and CSM)

Accuracy: Meet accuracy as defined in each sensor ORD

Operating Systems: (NOTE: The following operating systems are the oldest ones supported at the time of this document revision. Contact the government contracting agency or the CSM Working Group for the latest supported versions.)

UNIX for SPARC- Sun Solaris 9.0/10.0 or a backward compatible newer OS (32/64 bit)

Windows for x86 Intel Pentium class – Windows 2000/XP or a backward compatible newer OS (32/64 bit)

Linux for x86 Intel Pentium class – Red Hat Linux Enterprise 4.0 or a backward compatible newer OS (32/64 bit)

Compilers: (NOTE: The following compilers are the oldest ones supported at the time of this document revision. Contact the government contracting agency or the CSM Working Group for the latest supported versions.)

Sun Solaris SPARC – Sun Studio 9 or a backward compatible newer compiler

Sun Solaris SPARC – GNU Compiler Collection (GCC) version 3.0.4 or a backward compatible newer GCC compiler

Microsoft x86 Intel Pentium class – Visual C++.NET 2003 Standard Edition, Visual Studio .NET 2005 Standard Edition or a backward compatible newer Microsoft compiler

Linux x86 Intel Pentium class – GNU Compiler Collection (GCC) 3.4.4 or a backward compatible newer GCC compiler

Computer Systems: (NOTE: The following computer systems are the minimum supported at the time of this document revision. Contact the government contracting agency or the CSM Working Group for the latest supported versions.)

Sun Blade 2000, 900 MHz UltraSPARC III with 8MB External Cache

1 GB RAM, 73 GB 10000 RPM FC-AL Disk or newer backward compatible machine.

Dell Precision 340, Intel Pentium 4 Processor, 2.20 GHz, 512 Full Speed Cache

1 GM RAM, 36 GB Ultra 160/M SCSI, 10000 RPM Disk or newer backward compatible machine.

Appendix C

Application Programming Interface

(The API is available in a separate electronic file.)

Appendix D

Example Test Plan and Procedure

(The Test Plan and Procedure is available in a separate electronic file.)

Appendix E

Example of a Sensor Model Statement of Objectives

(The example SOO is available in a separate electronic file.)