As of 10 June 2016

This GOES-R Level 2+ space weather Algorithm Theoretical Basis Document (ATBD) is preliminary and subject to change as the GOES-R Program prepares for a launch in late 2016.

The NOAA National Centers for Environmental Information (NCEI) is currently developing a demonstration version of the Satellite Product Analysis and Distribution Enterprise System (SPADES) which will host the L2+ algorithms. The operational SPADES will be instantiated by the National Weather Service in FY2017 to serve the needs of the Space Weather Prediction Center (SWPC).

Updated versions of the L2+ ATBDs will periodically be made available and posted to the SPADES website.

Bill Denig
NCEI

http://www.ngdc.noaa.gov/stp/space-weather/online-publications/stp_sii/spades/

NOAA NESDIS CENTER for SATELLITE APPLICATIONS and RESEARCH

SUVI DIFFERENCE IMAGE ALGORITHM THEORETICAL BASIS DOCUMENT
Version 1.0

TITLE: SUVI DIFFERENCE IMAGE ALGORITHM THEORETICAL BASIS DOCUMENT VERSION
1.0

AUTHORS:
Dr. E. Joshua Rigler
Dr. Steven Hill

APPROVAL SIGNATURES:
____<Actual Signature Date>
<Name of Technical Approver> Date
Technical Approver
____<Actual Signature Date>
<Name of Agency Approver> Date
Agency

SUVI DIFFERENCE IMAGE ALGORITHM THEORETICAL BASIS DOCUMENT VERSION HISTORY SUMMARY

Version	Description	Revised Sections	Date
1.0	Created by Dr. E. Joshua Rigler CIRES/SWPC	New Document	Novermber 30, 2009

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AIT	Algorithm
AT	Application Team
ATBD	Algorithm Theoretical Basis Document
AWG	Algorithm Working Group
CCD	Charge Coupled Device
CDR	Critical Design Review
CIRES	Cooperative Institute for Research in Environmental Sciences
CME	Coronal Mass Ejection
EIT	EUV Imaging Telescope
EUV	Extreme UltraViolet
EUVS	EUV Sensor
HDR	High Dynamic Range
L1B	Level 1B (data)
MRD	
SEP	Solar Energetic Particle
SOHO	Solar and Heliospheric Observatory
STAR	Center for Satellite Apps and Research
SUVI	Solar Ultra-Violet Imager
SWPC	Space Weather Prediction Center
SXI	Solar X-ray Imager
AWG	Algorithm Working Group
XRS	X-ray Sensor

ABSTRACT

The Solar Ultraviolet Imager (SUVI) will provide high cadence images of the solar atmosphere to support Space Weather forecast activities at the Space Weather Prediction Center (NOAA/NWS/NCEP/SWPC). This document provides a comprehensive description of the SUVI Difference Image algorithm. The SUVI Difference Image algorithm can be used to generate two distinct L2+ data products: 1) running difference images, where the epoch image to be subtracted from the current baseline is simply the most recent compatible image; and 2) fixed difference images, where the epoch image is kept constant over a finite sequence of difference images, and defined as the most recent image at the time a so-called trigger is set. This document provides information to developers and reviewers necessary to verify the algorithm meets operational requirements. In particular, it addresses issues of image registration and navigation, as well as requirements for an operational software framework from which the algorithm must be called. Test and validation procedures are provided along with assumptions and known limitations of the algorithm.

1.0 INTRODUCTION

The GOES-R Algorithm Working Group (AWG) Space Weather (SWx) Application Team (AT) is responsible for producing the algorithms to generate products using the space environment data from the GOES-R series satellites. The algorithms shall meet operational needs of the NOAA/NWS Space Weather Prediction Center (SWPC) to observe and forecast space weather conditions impacting near-earth systems such as satellites, communications, electrical power grids, manned space missions and many others. They shall also meet the requirements of outside operational and research agencies as agreed. This ATBD provides details on the Solar Ultraviolet Imager (SUVI) Difference Image algorithm to inform potential users of the proposed methods and processes used to transform full dynamic range composite images of the sun into running and fixed difference images necessary to identify and track dynamic solar events typically associated with the most extreme forms of space weather.

1.1 Purpose of This Document

This ATBD provides details of the GOES-R SUVI Difference Image algorithm design and processing. This document shall be used to describe the SUVI algorithm that produces difference images of the sun in the extreme ultra-violet spectral range from observed GOES-R satellite data. It provides operational requirements for this product and defines how these requirements will be met with this algorithm. The algorithm inputs, processing and outputs are described in enough detail to design, develop, test and implement the necessary processing software and storage mechanisms.

1.2 Who Should Use This Document

The members of the Space Weather Forecast Office and the Research and Customer Requirements Section of the SWPC shall use this ATBD to verify their operational requirements are being met by the proposed algorithm. They should also use it to understand the strengths and weaknesses of the algorithm as well as its applicability, accuracy, and robustness. The STAR AIT group shall use this document to integrate the algorithm into their collaborative framework environment. It shall also be used by the prime development and implementation contractor to design, develop, test, validate and implement the algorithm into the final operational processing system.

1.3 Inside Each Section

Section 2.0 OBSERVING SYSTEM OVERVIEW:

- provides objectives of the SUVI Differrence algorithm including the output difference solar images and how they should be used operationally;
- discusses SUVI instrument characteristics and the fundamental algorithm approach;

Section 3.0 ALGORITHM DESCRIPTION:

- contains the complete SUVI Difference Image algorithm description including an outline of the processing, input data and a theoretical description;
- provides estimates of the algorithm performance and quality of the products;
- reviews the numerical computation, programming and procedural issues and a description of how the algorithm has been validated;

Section 4.0 TEST DATA SETS AND OUTPUTS:

- describes the test data sets used to characterize the performance of the algorithm and quality of the data product(s), including the breadth of the domain (typical versus stressing states) used in the analysis and assessment;
- discusses the results from algorithm processing on simulated input data;

Section 5.0 PRACTICAL CONSIDERATIONS:

- discusses issues involving numerical computation, programming and procedures, quality assessment and diagnostics and exception handling at a level of detail appropriate for the current algorithm maturity;

Section 6.0 ASSUMPTIONS AND LIMITATIONS:

- describes all assumptions concerning the SUVI Difference Image algorithm theoretical basis and performance;
- discusses planned product improvements for future enhancements;

Section 7.0 REFERENCES:

- Provides references to all sources cited in the ATBD.

1.4 Related Documents

1.5 Revision History

Revision Number	Date	Author	Revision Description	Reason for Revision

2.0 OBSERVING SYSTEM OVERVIEW

2.1 Product Generated

The SUVI Composite Image algorithm will generate high dynamic range images of different EUV solar scenes. However, while these images do contain most if not all of the information content that can be obtained considering instrumental limitations, it is not always possible for the human eye, or for that matter, many digital displays, to present this information in a way that maximizes the visibility of interesting features. This is particularly true of dynamic, or time-varying features when viewed as a single snapshot. By subtracting pixel values from a previous image from the most recent image, it is often possible to discern subtle changes in the EUV scene that might not have been visible otherwise.

There are two distinct types of difference images that are expected to be useful to space weather forecasters. Running differences capture the discrete time derivative of solar coronal dynamics, and are used to track migrating features like coronal waves. Such a sequence of images is presented in Figure 1. Fixed differences are more useful for a before/after view of specific events like coronal mass ejections (CMEs), which can be identified by a significant dimming of pixels across a relatively broad region of the solar disk or limb. This is demonstrated in Figure 2.

Finally, the underlying assumption of either differencing technique is that pixels sample identical regions of the solar surface. Accurate alignment of the image field of view (FOV) and point of view (POV) is critical to a valuable difference image product.

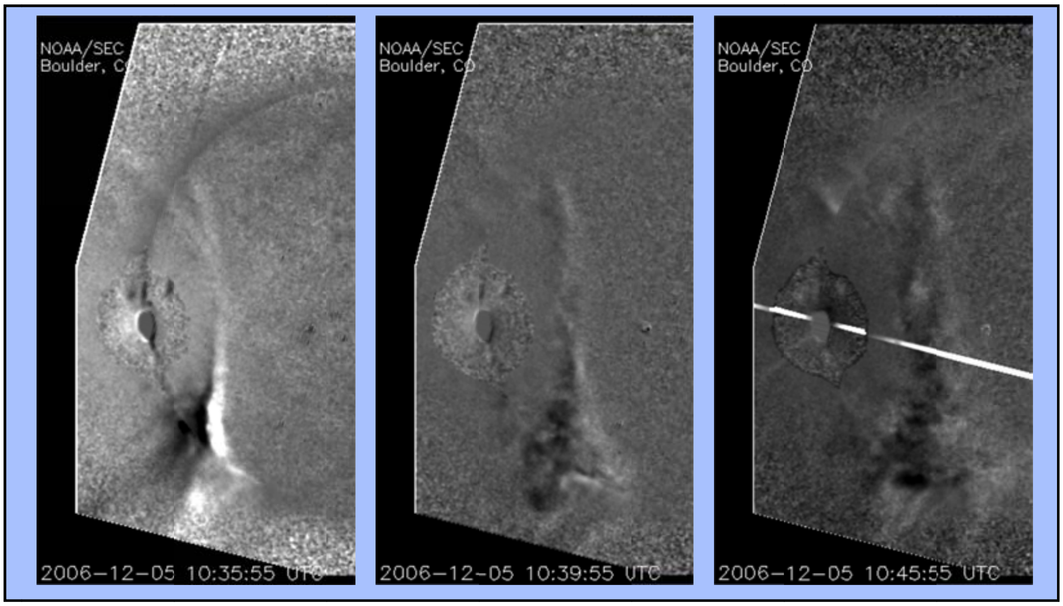


Figure 1 GOES SXI Running Difference Image Sequence
Coronal wave associated with flare observed beyond east limb on 5 December, 2006, ~10:30 UTC.

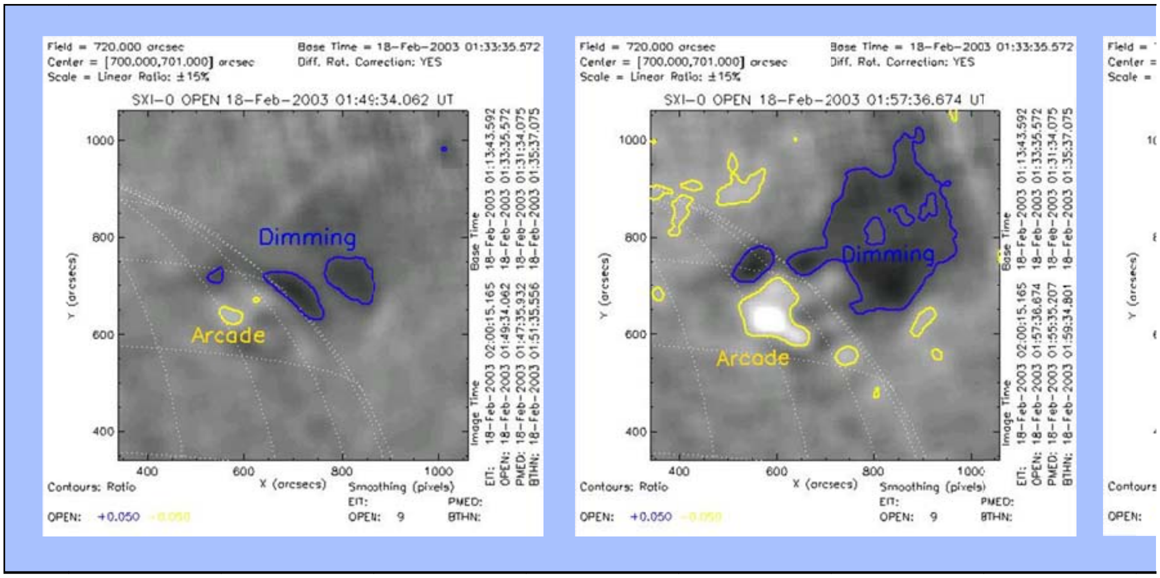


Figure 2 GOES SXI Fixed Difference Image Sequence
Coronal dimming associated with CMC on northwest limb on 18 February, 2003, ~01:33 UTC.

2.2 Instrument Characteristics

The individual SUVI channel specifications are given in the MRD section 3.4.2.4.0-4 and repeated below for ease of reference, along with the expected nominal sampling interval, and an example of the type of EUV feature that will be studied using each channel. The SWx Algorithm Team assumes the instrument meets the performance outlined in this section of the ATBD during development efforts.

Table 1 SUVI L1b image spectral channels

Channel	Sampling	Wavelength	Sample Use
SUVI (Fe XVIII)	every 4 min	9.39 nm	Flares (~6x106 °K)
SUVI (Fe XX / Fe XXIII)	every 4 min	13.1 nm	Flares (~10x106 °K) and Hot flares (~15x106 °K)
SUVI (Fe IX)	every 4 min	17.1 nm	Active regions (~6x105 °K)

SUVI (Fe XII / Fe XXIV)	every 4 min	19.5 nm	Active regions (~1x106 °K) and Hot Flares (~20x106 °K)
SUVI (Fe XV)	every 4 min	28.4 nm	Coronal holes (~2x106 °K)
SUVI (He II)	every 4 min	30.4 nm	Filaments (~6x104 °K)

3.0 ALGORITHM DESCRIPTION

3.1 Algorithm Overview

There are two distinct types of difference image generated by the Difference Image algorithm: 1) a running difference, which subtracts the most recent compatible image from the current baseline image; and 2) a fixed difference image, which subtracts the same image for a sequence of baseline images. The type of difference image generated by the algorithm depends on the definition of input parameters “trigger” and “epoch image”. The trigger is a logical variable that evaluates to either true or false. It must be set externally and passed to the algorithm by the software framework. The epoch image is a string, or some other unique identifier for the epoch image to be subtracted from the current baseline composite image.

3.2 Processing Outline

The following describe how the trigger and epoch image inputs are interpreted:

- A running difference image is always generated when the trigger is false, and the epoch image is undefined;
- if the trigger is true, but epoch image is undefined, a fixed difference sequence begins by using the most recent compatible image as its epoch image. This epoch image ID is then passed back to the calling routine to be used as an input to the Difference Image algorithm the next time it is called on to generate a fixed difference image;
- if the Difference Image algorithm is called with the trigger set to true, and the epoch image is defined, a fixed difference image is generated that continues an ongoing sequence of fixed difference images;

- if the trigger is set to false, and yet an epoch image is defined when the Difference Image algorithm is called, the last in a sequence of fixed difference images is generated, but no epoch image is passed back to the calling routine;

Figure 3 summarizes graphically the process flow of the SUVI Difference Image algorithm.

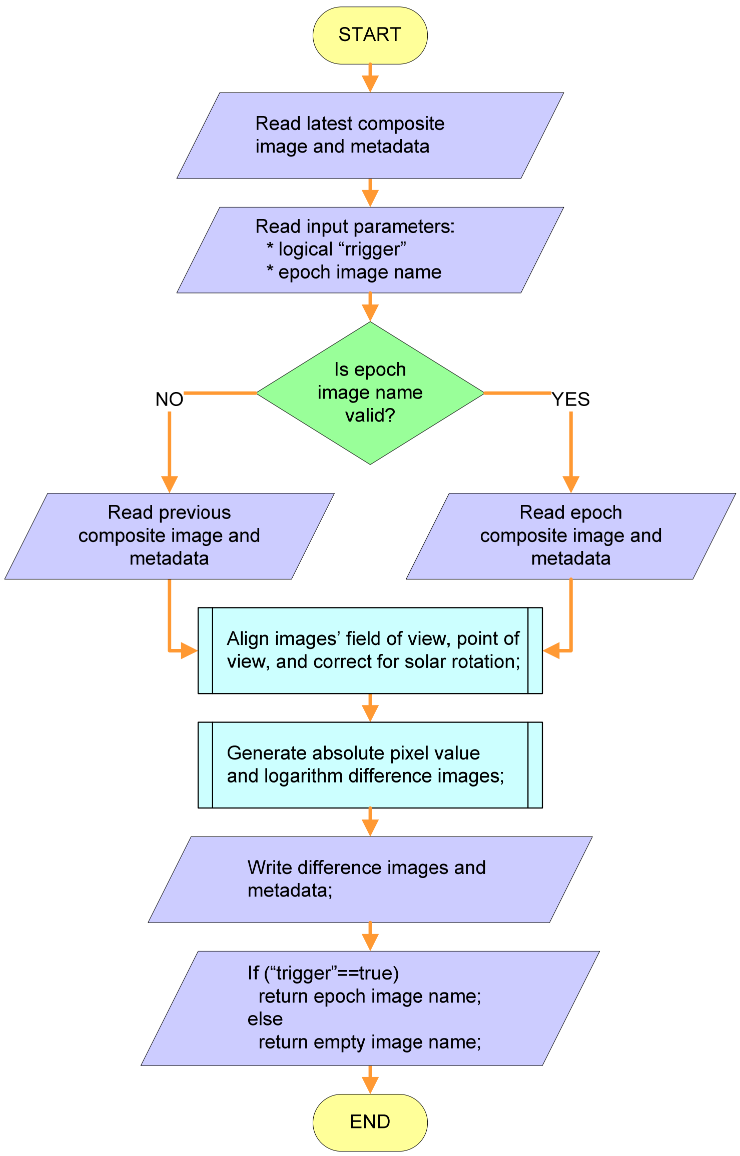


Figure 3 SUVI Differrence Image Algorithm
High level process flow of Difference Image algorithm implemented by the subroutine *sxi_image_difference*. The blue boxes correspond to sub-tasks implemented in, or called by the subroutine.

3.3 Algorithm Input

This section describes the input needed for the SUVI Difference Image algorithm.

3.3.1 Primary Sensor Data

The SUVI Difference Image algorithm does not operate on "primary sensor data", but rather uses L2 composite images as input. We consider this ancillary data, and describe it in the following subsection.

3.3.2 Ancillary Data

Ancillary data are assumed to be data that are not generated on-orbit by SUVI or the spacecraft. The only data required by the SUVI Difference Image algorithm consists of Level 2 Composite Images, a "trigger" specifying the start or continuation of a fixed difference image sequence, and the epoch image ID that was defined in the first call to a fixed difference image sequence, and is tracked between calls to the Difference Image algorithm by the space weather data product software framework.

The Difference Image algorithm will be applied to L2 composite images of each of SUVI's six EUV wavelength bandpasses. These were shown in Table 1, along with the primary spectral emission line(s) being sampled, the nominal sampling interval, and an example of the solar EUV features that the bandpass will be used to study. Each of these images must, at a minimum, contain the information detailed in Table 2 for the Difference Image algorithm to function properly.

Table 2. SUVI L2 composite image inputs to Difference Image algorithm

Name	Type	Description	TYPE (Dimension)
Date/Time	input	Image date and time tags (e.g. YYYY,MM,DD,hh,mm,ss.sss)	CHAR(1) or Integer(7)
Channel/Filter	input	Identifier for channel/filter combo	CHAR (1)
Pixels	input	L2 Composite Image pixel value	REAL (xsize, ysize)
x_0	input	Pixel *x*-coordinate of solar disk center	REAL (1)
y_0	Input	Pixel *y*-coordinate of solar disk center	REAL (1)
θ_z	input	CCW rotation angle to align image and solar y axes	REAL (1)
Δ_x	Input	Pseudo-angle describing a pixel's plate scale in the X direction	REAL (1)
Δ_y	input	Pseudo-angle describing a pixel's plate scale in the Y direction	REAL (1)
ϕ_0	input	Heliographic longitude of SUVI	REAL (1)
B_0	input	Heliographic latitude of SUVI	REAL (1)
$D_\odot$	input	Distance from sun center to SUVI	REAL (1)

The date/time and channel/filter combinations are necessary to isolate the baseline and epoch images of each difference image. The pixel values are necessary for the actual comparison and differencing. The remaining image metadata are required for field of view and point of view alignment. Note that composite images that were generated using the SUVI Composite Image algorithm will have additional information, including the exposure time and an array of pixel weights. These are not required by the SUVI Difference Image algorithm, and will in fact be modified as can be seen below.

3.4 Theoretical Description

3.4.1 Physics of the Problem

There is little to describe in terms of "physics" for the SUVI Difference Image algorithm. Pixels that have grown brighter since the last compatible epoch image will appear as positive values; pixels that have grown dimmer since the last compatible epoch image will appear as negative values. Rapidly evolving features on the solar surface will be more easily identified and tracked using running difference images, while significant and semi-permanent reconfigurations of features on the solar surface will be more apparent in fixed difference images.

3.4.2 Mathematical Description

The mathematics involved in a difference are fairly simple: subtract the value of an epoch image pixel from the value of a baseline image pixel.

$$\Delta X_i = X_{i,baseline} - X_{i,epoch} \tag{3.1}$$

In some cases it is more appropriate to calculate $\log_{10}$ pixel values, then subtract these, which amounts to a ratio of pixel values.

$$\Delta \log_{10}(X_i) = \log_{10}(X_{i,baseline}) - \log_{10}(X_{i,epoch}) \tag{3.2}$$

The epoch composite image pixels are assumed to be aligned in time and space with the baseline composite image pixels. Herein lies the primary difficulty associated with the Difference Image algorithm.

Field of View Image Alignment

Pointing knowledge of instruments is typically much more reliable than the ability to actually control exactly where the instrument is pointed. This is particularly true for SUVI, since it is mounted on an observation platform whose primary function is not solar viewing, but rather Earth viewing. This pointing knowledge is encoded into each SUVI image, and should be used to align L1b images before they are merged into composites. There are essentially two kinds of alignment we are concerned with: 1) field of view (FOV), which relates to the orientation between the target and the imager's focal plane; and 2) the point of view (POV), which relates to the location of the imager in a target-centered coordinate system. Techniques for dealing with the first are described in this subsection, while techniques for dealing with the second are described in the following subsection.

To start, the center of each pixel must be assigned a floating point Cartesian coordinate. The actual coordinate system is not important, but we choose one in which the image center is the coordinate system origin, or {0,1}, to simplify things. Image rotations, translations, and scaling are most efficiently performed using matrix operations. This requires that the coordinates of each pixel be stored as a column vector in the following format:

$$\mathbf{P} \equiv \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (3.3)$$

The reason for this special format becomes clear when one considers the forms potential 2D transformation matrices will take:

$$\mathbf{R} \equiv \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T} \equiv \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{S} \equiv \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.4)$$

A position vector must be pre-multiplied by the associated transformation matrix. The matrix **R** rotates a point θ radians about the origin in the counter-clockwise direction. The matrix **T** translates, or shifts, the position of a point by t_x in the x direction, and by t_y in the y direction. Finally, the matrix **S** multiplies a point's position by s_x in the x direction, and s_y in the y direction, effectively rescaling the entire image.

If more than one operation needs to be performed on each pixel location (e.g., the image must be rotated by the angle θ, followed by a translation { t_x, t_y }), the associative property of matrix multiplication makes it possible to multiply transformation matrices together to produce a single transformation that can be applied to the pixel location vectors just once.

Matrix multiplication is not, however, commutative, so individual transformations must be multiplied by one another in the same order they would be performed separately (i.e., **P**' = **T** x **P**, **P**'' = **R** x **P**', is the same as **P**"=(**T** x **R**) x **P**). For example, to rotate the image by the angle θ, followed by the translation { t_x, t_y }, we generate a new transformation matrix **A**:

$$\mathbf{A} \equiv \mathbf{T} \times \mathbf{R} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.5)$$

At this point, it is important to note that we have been discussing the rotation, translation, and rescaling of pixel coordinates themselves, but in reality it is the pixel coordinate frame that must be transformed before 2D interpolation can be used to resample the original image, and create the required transformed image for subsequent operations. This means we must apply the inverse of our coordinate transform. To understand this better, consider Figure 4, where the inverse of individual transforms is shown graphically. To rotate an image counter-clockwise, pixel coordinates must be rotated clockwise. To shift an image down and left, pixel coordinates must be shifted up and right. And to expand an image, pixel coordinates must be scaled down. 2D interpolation is then used to generate new pixel values for these coordinates based on the known values at the center of every pixel in the original image.

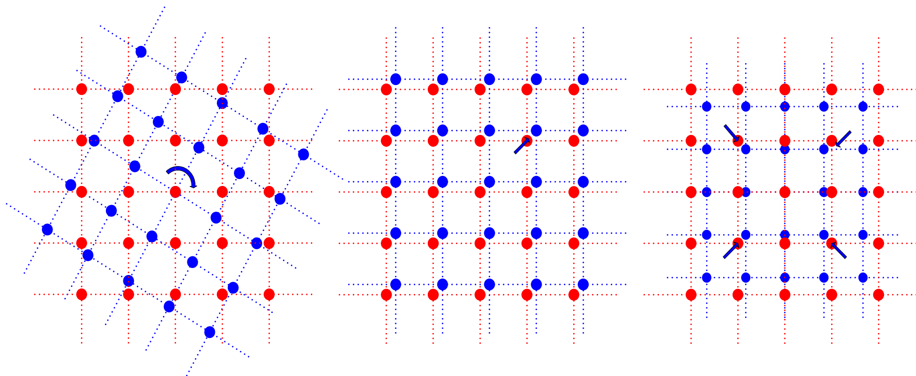
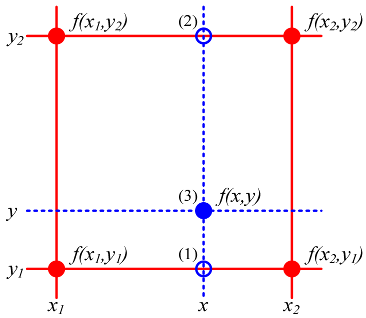


Figure 4 Transforming coordinate systems
The coordinate system of an image array (blue) is rotated by angle $-\theta$ in lieu of an actual image rotation of $+\theta$ (left); the coordinate system translates $\{+t_x,+t_y\}$ to achieve image translation of $\{-t_x,-t_y\}$ (middle); the coordinate system is scaled by $\{1/s_x,1/s_y\}$ to scale the image by $\{s_x,s_y\}$ (right). In all cases, the original image pixels remain on a fixed, rectangular grid to facilitate interpolation.

Bilinear interpolation is conceptually simple: 1) interpolate linearly to a desired x coordinate that falls between two known values with the same y coordinate; 2) interpolate linearly to the same desired x coordinate between two values with a different y coordinate; 3) interpolate linearly to a desired y coordinate that falls between the two y coordinates used in steps 1 and 2, along the vertical line passing through the desired x coordinate. Given four known pixel values f that fall at the lower-left (x_1,y_1), lower-right (x_2,y_1), upper-left (x_1,y_2), and upper-right (x_2,y_2) corners of a rectangular region of interest, interpolation to a desired point within this region is defined mathematically as:



$$f(x,y) \approx \frac{f(x_1,y_1)}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \frac{f(x_2,y_1)}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) + \frac{f(x_1,y_2)}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) + \frac{f(x_2,y_2)}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1) \quad (3.6)$$

This equation is generally applicable. However, if the coordinate system in which the four points are known can be scaled such that $x_1=y_1=0$ and $x_2=y_2=1$, a more compact and efficient bilinear interpolation scheme based on matrix operations may be used:

$$f(x,y) \approx [(1-x) \quad x] \times \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \times \begin{bmatrix} (1-y) \\ y \end{bmatrix} \quad (3.7)$$

The L1b SUVI images are expected to each possess metadata that will define the translation necessary to center the solar disk on the image array, and the rotation angle necessary to align solar spin axis with the image *y*-axis. Furthermore, each L1b SUVI image should possess metadata that define the pixel plate scale. The ratio of this plate scale and that of GOES 13 SXI can be used to derive a scaling factor that guarantees SUVI images will be directly comparable to GOES 13 SXI

Point of View Image Alignment.

The point of view (POV) of an image relates to the location in 3D space of the imager, and should be adjusted to align pixels in images that have not been obtained from identical locations.

We first correct for the distance from the Sun, rescaling the image so that the Sun's apparent radius is always equal to the apparent radius seen from 1AU.

There is currently no correction for perspective change, since variations in the imager's distance from the sun would lead to changes in actual visible solar surface area that are not detectable using 32-bit floating point pixel coordinates. If the algorithm were ever applied to non-SUVI images, this shortcoming may need to be remedied.

We also correct for changes in heliographic latitude and longitude. While latitude does not vary appreciably between L1b images, heliographic longitude varies at a rate of ~14 degrees per day. We might neglect this correction for images taken just a few minutes apart, but as the time between images approaches an hour or more, features will shift by whole pixels and more at the center of the solar disk. Normal composite images whose time stamps differ by only seconds to minutes will not suffer much, but difference images and composites of composites used in other SUVI L2+ products will be severely impacted if these POV corrections are not made.

The first step is to convert 2D pixel coordinates to 3D heliocentric Cartesian. The X and Y coordinates simply become the pixel coordinate multiplied by the pixel plate scale. The Z coordinate is determined by assuming a spherical sun with know radius, and applying (3.8).

$$p_z = \sqrt{R_S^2 - p_x^2 - p_y^2} \tag{3.8}$$

Once 3D Cartesian coordinates are available, changes in perspective not related to distance from the Sun are applied using three rotations: 1) rotate from the heliographic latitude of the observer to the solar equator; 2) rotate through the change in heliographic longitude ($\Delta\phi = \phi_{targ} - \phi_0$); and 3) rotate to the target heliographic latitude. The transformation matrices for these rotations are presented in (3.9).

$$
\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-B_0) & \sin(-B_0) \\ 0 & -\sin(-B_0) & \cos(-B_0) \end{bmatrix} \times \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}
$$

$$
\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos(\Delta\phi) & 0 & -\sin(\Delta\phi) \\ 0 & 1 & 0 \\ \sin(\Delta\phi) & 0 & \cos(\Delta\phi) \end{bmatrix} \times \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad (3.9)
$$

$$
\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(B_{targ}) & \sin(B_{targ}) \\ 0 & -\sin(B_{targ}) & \cos(B_{targ}) \end{bmatrix} \times \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}
$$

If no further POV alignment were necessary, the X and Y components of the resulting pixel coordinates would be divided by the plate scale to return to image pixel coordinates, and all pixels with negative Z coordinates would be filtered out, since these are not visible the observer. Pixels that were not visible before these rotations will be assigned “bad” values, or NaNs.

However, the Sun exhibits differential rotation in that its apparent surface spins faster at the equator and slower toward the poles. A variety of models of this differential rotation exist in the literature, and nearly all of them define the time derivative of longitude as a function of latitude, so it is necessary to convert the 3D Cartesian coordinates to latitude and longitude in order to correct for the Sun’s differential rotation. This can be done using equations in (3.10) (see Thompson (2006), *Astronomy & Astrophysics*).

$$
\begin{aligned}
r_i &= \sqrt{p_x^2 + p_y^2 + p_z^2} \\
\theta_i &= \sin^{-1}\left(\left(p_y \times \cos B_{targ} + p_z \times \sin B_{targ}\right)/r\right) \qquad (3.10) \\
\phi_i &= \phi_{targ} + \operatorname{atan2}\left(p_x, p_z \times \cos B_{targ} - p_y \times \sin B_{targ}\right)
\end{aligned}
$$

Once Cartesian pixel coordinates have been converted to radius (r_i), latitude (θ_i), and longitude (ϕ_i), the transformation in (3.11) can be applied to each pixel’s coordinates to simulate differential rotation (see Snodgrass and Ulrich (1990), Astrophysical Journal).

$$\begin{aligned} \frac{d\phi_i}{dt} &= 14.713 - \left(2.396 \times \sin^2 \theta_i\right) - \left(1.787 \times \sin^4 \theta_i\right) \\ \theta_i &= \theta_i + \frac{d\phi_i}{dt} \times \Delta t \end{aligned} \quad (3.11)$$

To convert heliographic coordinates back to Cartesian, use the relations in (3.12) (see Thompson (2006), *Astronomy & Astrophysics*).

$$\begin{aligned} p_x &= \cos\theta_i \times \sin\left(\phi_i - \phi_{targ}\right) \times r \\ p_y &= \left(\sin\theta_i \times \cos\theta_{targ} - \cos\theta_i \times \cos\left(\phi_i - \phi_{targ}\right) \times \sin\theta_{targ}\right) \times r \\ p_z &= \left(\sin\theta_i \times \sin\theta_{targ} - \cos\theta_i \times \cos\left(\phi_i - \phi_{targ}\right) \times \cos\theta_{targ}\right) \times r \end{aligned} \quad (3.12)$$

3.4.3 Algorithm Output

The SUVI Difference Image algorithm generates output images that are somewhat similar in structure to its inputs, but quite different in content. The exposure time metadata variable is retained, but set equal to NaN because it has no meaningful definition for difference images. The pixel difference array holds the linear difference between baseline image pixel values and epoch image pixel values. The $\log_{10}$ pixel difference array holds the difference in $\log_{10}$ values, which is essentially the ratio of pixel values. All FOV and POV related metadata correspond to the baseline image. These outputs are summarized in Table 3.

Table 3. Level 2 Outputs of SUVI Composite Image algorithm.

Name	Type	Description	TYPE (Dimension)
Date/Time	output	Image date and time tags (e.g. YYYY,MM,DD,hh,mm,ss.sss)	CHAR(1) or Integer(7)
Channel/Filter	output	Identifier for channel/filter combo	CHAR (1)
Exposure Time	output	Not applicable to difference images	NaN
Pixel Difference	output	Difference in image pixel values	REAL (xsize, ysize)
Difference ($\log_{10}$)	output	Difference in image pixel $\log_{10}$ values	REAL (xsize, ysize)
x_0	output	Pixel x-coordinate of solar disk center	REAL (1)
y_0	output	Pixel y-coordinate of solar disk center	REAL (1)
θ_z	output	CCW rotation angle to align image and solar y axes	REAL (1)
Δ_x	output	Pseudo-angle describing a pixel's plate scale in the X direction	REAL (1)
Δ_y	output	Pseudo-angle describing a pixel's plate scale in the Y direction	REAL (1)
ϕ_0	output	Heliographic longitude of SUVI	REAL (1)
B_0	output	Heliographic latitude of SUVI	REAL (1)
$D_\odot$	output	Distance from sun center to SUVI	REAL (1)

4.0 TEST DATA SETS AND OUTPUTS

4.1 Simulated/Proxy Input Data Sets

Composite images generated by the SUVI Composite Image algorithm using proxy images derived from SOHO/EIT images are used to test the Difference Image algorithm. These tests are described in detail in the SUVI Difference Image Test Plan document (Rigler, 2009), along with the expected test results. Here we describe the test and proxy data in some detail.

These images were chosen because there was a weak coronal mass ejection (CME) off of the west (right) limb during this period that should be captured by the fixed difference images in particular. Also, while there were no on-disk CMEs or other major events to generate coronal waves, there were several bright regions that fluctuated during the sequence which should be evident in running difference images. Figure 5 shows these proxy data.

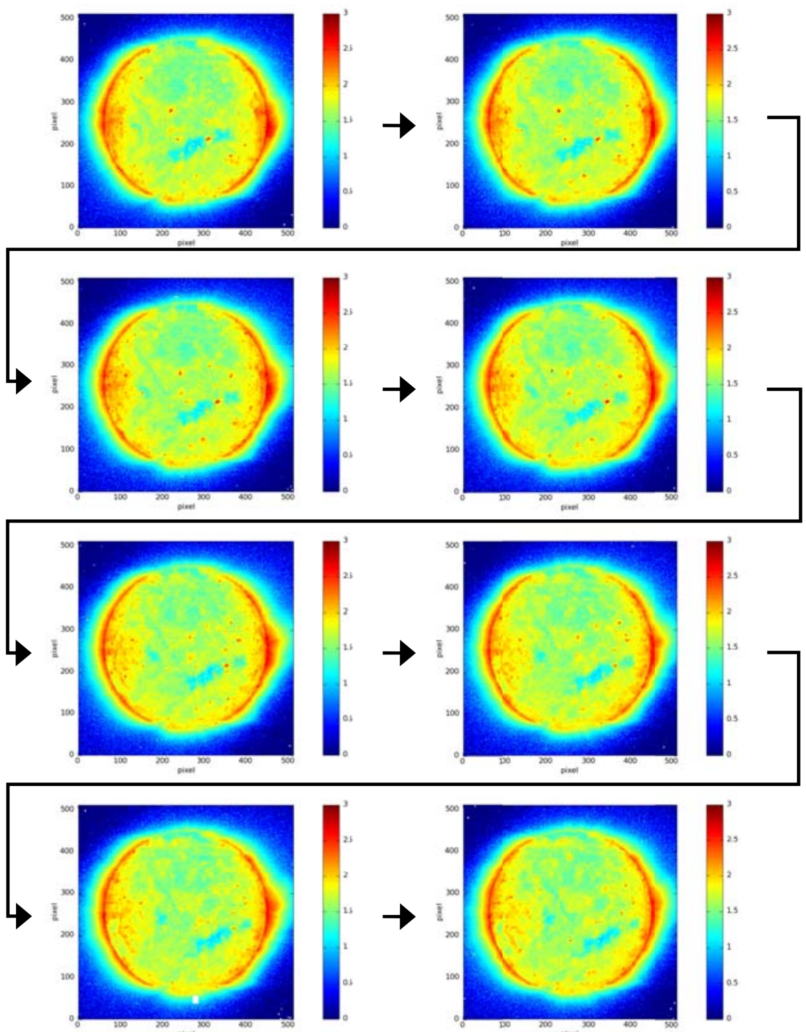


Figure 5 Composites constructed from modified SOHO/EIT 19.5 nm images and using the SUVI Composite Image algorithm.

In addition to these composites, the difference algorithm was called eight times with input trigger set equal to false, and no epoch image defined in order to generate the running difference images. It was then called eight times with input trigger set true for images 1-3, then 5-8, to generate two sequences of four fixed difference images apiece.

4.2 Output from Simulated/Proxy Inputs Data Sets

Running Difference Images

The non-log running difference images generated from the composite image inputs described previously are presented in Figure 6. Items to note:

- the first difference image in the sequence is blank because there were no valid proxy images available prior to the first image's observation date;
- the last two difference images in the sequence contain a square region of Null/NaN values because this region exists in the 7^{th} composite image, which is part of the difference equation for difference images 7 and 8;
- a small, positive blob just north and east (left) of center in the first valid difference image turns negative in the second image, then less negative with each successive image, indicating an initial brightening after the first image, followed by a gradual dimming;

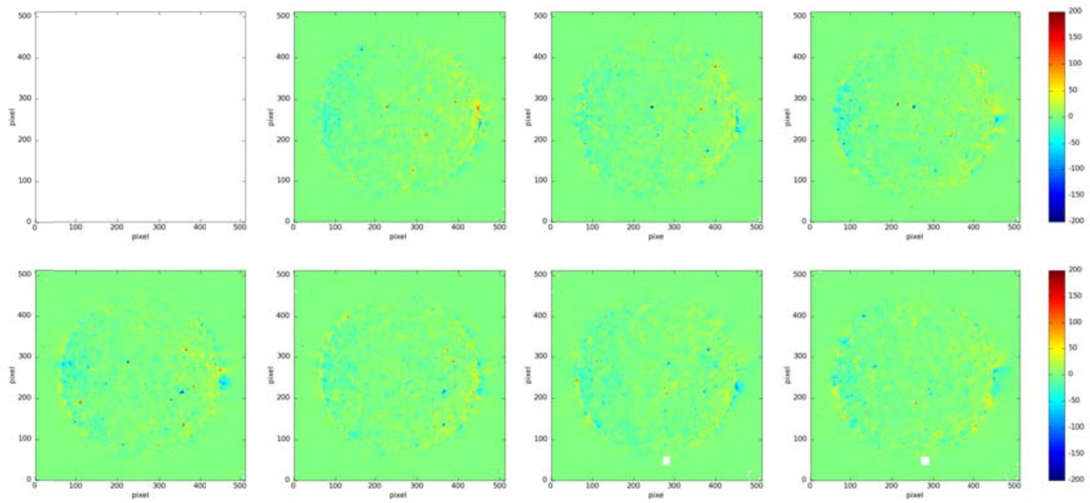


Figure 6 Example Running Difference Images
Eight running differences of composites from Figure 5, with no valid epoch image defined for the first in the sequence.

Running Difference Images

The non-log fixed difference images generated from the composite image inputs described previously are presented in Figure 7. Items to note:

- the first difference image in the sequence is blank because there were no valid proxy images available prior to the first image's observation date;
- the 7th difference image in the sequence contains a square region of Null/NaN values because this region exists in the 7th composite image; it does not exist in the 5th or 8th composite image, so there is none in the 8th difference image;
- a crescent of Null/NaN pixels evolves on the east (left) limb as differential rotation is corrected for to align the epoch image with the current baseline;
- the more easterly (left) pixels tend toward negative differences, while the more westerly (right) pixels tend toward positive differences, both due to the fact that an optically thin corona is always brighter closer to the limb;

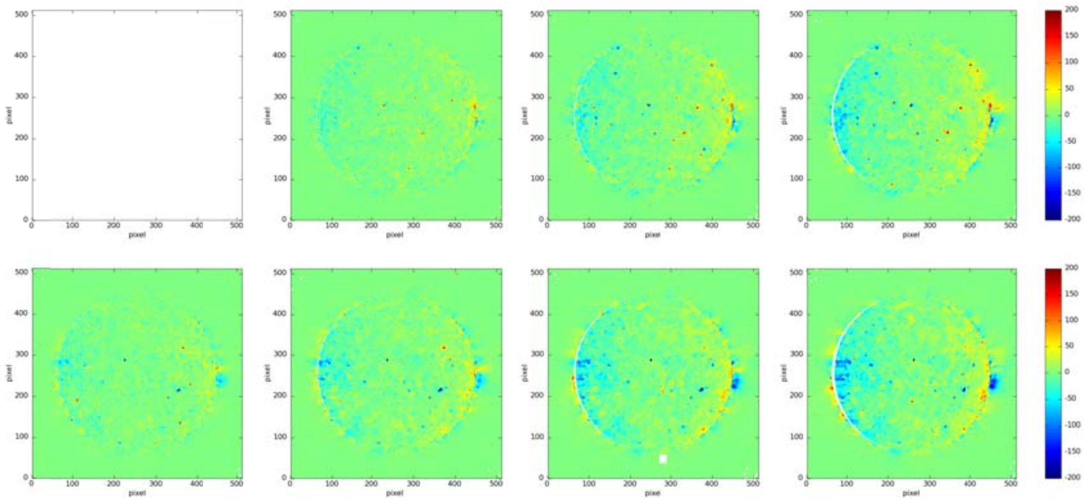


Figure 7 Example Fixed Difference Images
Eight fixed differences of composites from Figure 5, with no valid epoch image defined for the first in the sequence.

4.2.1 Precisions and Accuracy Estimates

TBD

4.2.2 Error Budget

TBD

5.0 PRACTICAL CONSIDERATIONS

5.1 Numerical Computation Considerations

There are no significant numerical computation considerations in the current implementation of the Composite Image algorithm.

5.2 Programming and Procedural Considerations

Triggers necessary for fixed difference image to be generated by the SUVI Difference Image algorithm will need to be set by some sort of space weather product software framework. Likewise, the epoch image ID passed back from the Difference Image algorithm once a fixed difference image sequence has been initiated must be accepted and stored by the framework for subsequent calls to the algorithm. This framework does not exist as of the delivery date for the SUVI Difference Image algorithm.

5.3 Quality Assessment and Diagnostics

Difference images will be periodically retrieved, examined, and compared with contemporaneous data from other solar imaging instruments (operational and scientific) operating at similar wavelengths in order to assess the quality of the output.

Space weather forecasters are trained to recognize deviations from normal conditions, and so will be consulted regularly to determine if, and to what degree, the Difference Image product quality degrades with time. Such changes will almost certainly be due to degradation in the L1B data, but it may be possible to implement changes to the L2 processing that can alleviate some of these issues.

5.4 Exception Handling

Possible exceptions, and the required actions, include:

- bad or incomplete metadata in single or both composite input data files;
    - action: generate difference image of Null/Nan values
- bad pixel in single or both composite input data files (missing or otherwise flagged data);
    - action: assign Null/NaN value to difference image pixel;

5.5 Algorithm Validation

TBD

6.0 ASSUMPTIONS AND LIMITATIONS

6.1 Performance

TBD

6.2 Assumed Sensor Performance

TBD

6.3 Pre-Planned Product Improvements

6.3.1 Improvement 1

If the FOV and POV related metadata provided along with the L2 composite image is not accurate enough to guarantee image alignment between components of a L2 difference image, an alternative form of the SUVI Difference Image algorithm will include an option to perform optimized pixel alignment. This process will involve making sub-pixel adjustments to FOV and POV metadata in the epoch image until a maximum in the correlation coefficient between two images is achieved. This is expected to be substantially more computationally expensive than the current algorithm, so additional optimization of the current algorithm for speed and memory usage will be required.

7.0 REFERENCES

Debevec, P. E. and M. Jitendra (1997), Recovering high dynamic range radiance maps from photographs, SIGGRAPH Proceedings

Hapgood, M.A. (1992), Space physics coordinate transformations: a user guide, *Planet. Space Sci.*

Lucke, R. L. and R. A. Kessel (2006), Signal-to-noise ratio, contrast-to-noise ratio, and exposure time for imaging systems with photon-limited noise, *Opt. Eng.*

Rigler, E. J., S. Hill, GOES R SUVI Difference Image Implementation and User's Guide, Version 1.0, November 30, 2009.

Rigler, E. J., S. Hill, GOES R SUVI Difference Image Algorithm Test Plan and Results, Version 1.0, November 30, 2009.

Snodgrass and Ulrich (1990), "Rotation of Doppler features in the solar photosphere, *Astrophysical Journal*

Thompson, W. T. (2006), Coordinate systems for solar image data, *Astronomy & Astrophysics*