Draft

NOAA NESDIS
CENTER for SATELLITE APPLICATIONS and RESEARCH

ALGORITHM THEORETICAL BASIS DOCUMENT

Convective Initiation

John R. Walker, University of Alabama in Huntsville
John R. Mecikalski, University of Alabama in Huntsville

Version 2.0

June 27, 2011

Table of Contents

Draft

LIST OF FIGURES

Draft

LIST OF TABLES

Draft

LIST OF ACRONYMS

ABI: Advanced Baseline Imager

AIT: Algorithm Integration Team

ATBD: Algorithm Theoretical Basis Document

AWG: Algorithm Working Group

CI: Convecive Initiation

CIRA: Cooperative Institute for Research in the Atmosphere

DMW: Derived Motion Winds

FAR: False Alarm Ratio

F&PS: Functional and Performance Specification

GOES: Geostationary Operational Environmental Satellite

IDN: Identification Number

LZA: Local Zenith Angle

MSG: Meteosat Second Generation

NCDC: National Climatic Data Center

OTA: Object Tracking Algorithm

POD: Probability of Detection

PQI: Product Quality Information

QF: Quality Flags

RAMS: Regional Atmospheric Modeling System

SEVIRI: Spinning Enhanced Visible Infrared Radiometer Imager

t1: Time 1

t2: Time 2

Draft

ABSTRACT

The GOES-R Algorithm Working Group (AWG) Convective Initiation (CI) algorithm is based upon the findings from several previous studies. It uses a multi-spectral approach to integrate many of the satellite-based methods for detecting cloud top properties, such as relative cloud top height, cloud top glaciation, and associated updraft strength. The many channel sensors that will be available from the GOES-R Advanced Baseline Imager (ABI) will improve the capability to diagnose such cloud properties, which can be used to infer future cloud development and generate near-future forecasts of CI for individual clouds across a broad spatial domain. This algorithm theoretical basis document (ATBD) will provide details about the algorithm processing, algorithm performance and validation, and the background work leading into its development.

Draft

1. INTRODUCTION

1.1 Purpose of This Document

The convective initiation ATBD provides a high level description of and the physical basis for the assessment of CI that is derived using the ABI aboard the GOES-R series of NOAA geostationary meteorological satellites. The CI algorithm provides an assessment of which convective clouds are most likely to precipitate. It does so by monitoring the growth of non-precipitating clouds, using a series of spectral and temporal threshold tests created to identify clouds that are growing at such a rate that they are likely to produce convective rainfall with radar-derived reflectivities ≥35 dBZ within approximately 0-2 hours (Mecikalski and Bedka 2006, Mecikalski et al. 2008, Mecikalski et al., 2010a).

1.2 Who Should Use This Document

The intended users of this document are those interested in understanding the physical basis of the CI algorithm and how to use the output of this algorithm to determine which clouds are likely to produce convective rainfall in the near future. This document also provides information useful to anyone maintaining or modifying the original algorithm.

1.3 Inside Each Section

This document is broken into the following main sections.

- **Observing System Overview**: Provides relevant details of the ABI and provides a brief description of the product generated by the algorithm.

- **Algorithm Description**: Provides a detailed description of the algorithm, including its physical basis, its input, and its output. Validation will also be addressed.

- **Assumptions and Limitations**: Provides an overview of the current limitations of the algorithm approach and gives the plan for overcoming these limitations with further algorithm development.

1.4 Related Documents

This document currently does not relate to any other document outside of the Functional and Performance Specification (F&PS) specifications or to the references given throughout.

1.5 Revision History

Version 1.0 of this document was created by Wayne M. MacKenzie, Jr., John R. Walker, and John R. Mecikalski of the University of Alabama in Huntsville, and its intent was to accompany the delivery of the version 1.0 algorithm to the GOES-R AWG Algorithm Integration Team (AIT). This updated, version 2.0, document has, since, been updated by John R. Walker and John R. Mecikalski to clarify certain sections and to add details relevant to the 100% algorithm code delivery.

Draft

Draft

2. OBSERVING SYSTEM OVERVIEW

This section gives an overview of the algorithm, including its objectives and characteristics. Specific requirements are referenced, and pertinent information about the ABI channels needed for input is provided.

2.1 Products Generated

The CI algorithm produces a binary field at 2 km spatial resolution of areas where CI has a high likelihood of occurring. The algorithm employs a cloud "object tracking" methodology, which tracks clouds within their early stages of development. Then, it monitors their spectral characteristics using a well-documented temporal/spectral differencing technique. If a majority of the spectral "interest field" thresholds are exceeded, then the group of pixels within the cloud object are flagged for having a high likelihood for CI. The F&PS requirements for the CI algorithm are given in Table 1.

Name	User & Priority	Geographic Coverage (G, H, C, M)	Vertical Resolution	Horizontal Resolution	Mapping Accuracy	Measurements Range	Measurements Accuracy	Product Refresh Rate/Coverage Time	Vendor Allocated Ground Latency	Product Measurement	Temporal Coverage Qualifiers
Convective Initiation	GOES-R	C	N/A	2 km	1 km	Binary Yes/No detection	70% Probability of Correct Detection	5 min	159 sec	N/A	Day and night
Convective Initiation	GOES-R	M	N/A	2 km	1 km	Binary Yes/No detection	70% Probability of Correct Detection	5 min	159 sec	N/A	Day and Night

Table 1. F&PS CI Algorithm Requirements. "C" is for CONUS, and "M" is for mesoscale.

2.2. Instrument Characteristics

The CI algorithm will employ many of the various spectral channels available from the GOES-R ABI. The channels used directly by the algorithm are listed in Table 2. Note that this list does not include the channels required to generate the Cloud Type algorithm output, which is a necessary input for the CI algorithm. A list of channels required for the Cloud Type algorithm may be found in another document.

Channel Number	*Wavelength (µm)*	*Projected to be used in CI processing*
1	**0.47**	
2	**0.64**	
3	**0.86**	
4	**1.38**	
5	**1.61**	
6	**2.26**	
7	**3.9**	
8	**6.15**	**X**
9	**7.0**	
10	**7.4**	**X**
11	**8.5**	**X**
12	**9.7**	
13	**10.35**	
14	**11.2**	**X**
15	**12.3**	**X**
16	**13.3**	**X**

Table 2. ABI channel numbers and the wavelengths used directly in the CI algorithm.

The algorithm relies only on infrared channels available from the ABI, so that day/night continuity will be present. The performance of the algorithm may be sensitive to any instrument noise; however, these effects would likely be minimal.

3. ALGORITHM DESCRIPTION

This section gives a complete description of the algorithm at its current level of maturity.

3.1 CI Algorithm Overview

Mecikalski and Bedka (2006) first showed that one can track growing cumulus clouds, monitor their spectral properties, and use a set of threshold-based indicators to determine the likelihood that a particular cumulus will precipitate in the near future, using only a satellite-based remote sensing approach. These indicators incorporate spectral and temporal differences that provide information about cloud phase and the relative location of cloud-tops within the troposphere. That information can be used to determine the maturity of convective clouds. Additionally, the growth of clouds through the troposphere between two successive satellite image times can be detected using several of the spectral channels listed in Table 2. Knowledge of this process can yield information relevant to the developmental stage of a cumulus cloud and help to identify whether a cloud will precipitate in the near future (0-2 hours).

3.2 Processing Outline

The processing outline of the CI algorithm is summarized in the flowchart from Figure 1. The CI algorithm currently uses satellite data in netCDF format for input into the FORTRAN-based processing code. ABI data (the most recent imagery, “t2”, along with imagery from the previous image scan time, “t1”) and the Cloud Type algorithm output (most recent output, “t2”, along with that from the previous scan time, “t1”) are required to begin processing the CI algorithm.

Draft

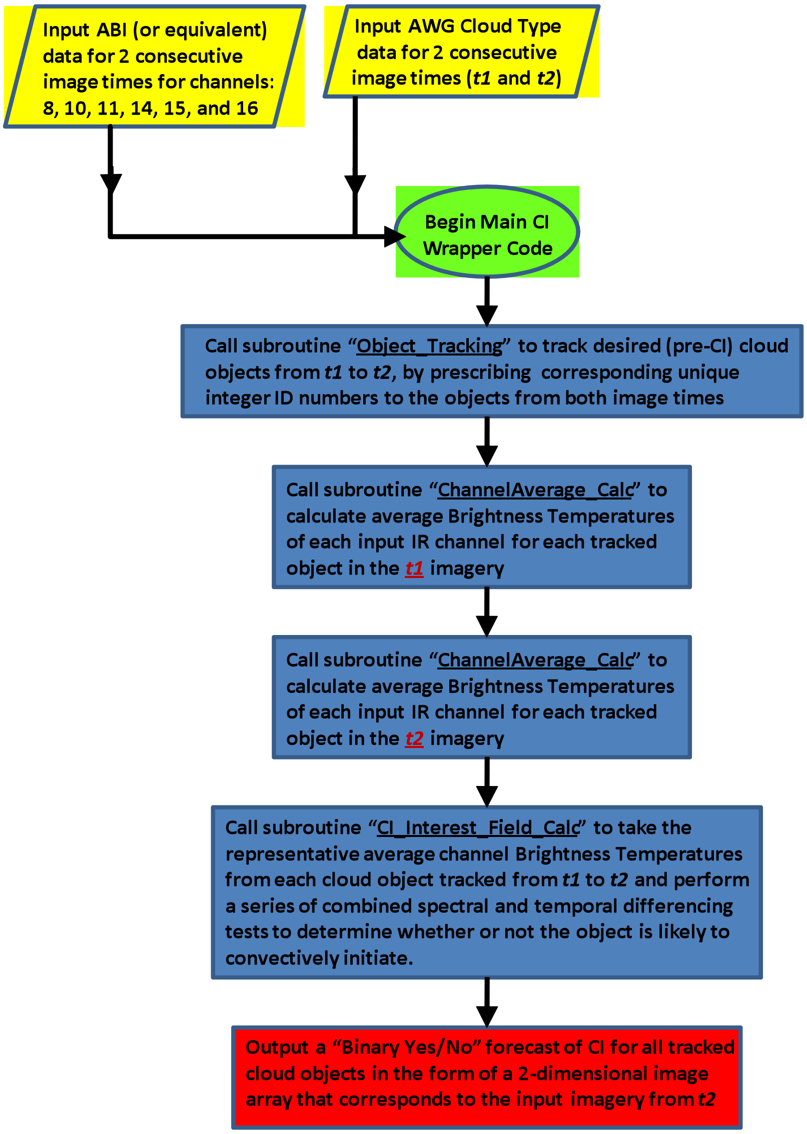
Input ABI (or equivalent) data for 2 consecutive image times for channels: 8, 10, 11, 14, 15, and 16
Input AWG Cloud Type data for 2 consecutive image times (*t1* and *t2*)
Begin Main CI Wrapper Code
Call subroutine "Object_Tracking" to track desired (pre-CI) cloud objects from *t1* to *t2*, by prescribing corresponding unique integer ID numbers to the objects from both image times
Call subroutine "ChannelAverage_Calc" to calculate average Brightness Temperatures of each input IR channel for each tracked object in the *t1* imagery
Call subroutine "ChannelAverage_Calc" to calculate average Brightness Temperatures of each input IR channel for each tracked object in the *t2* imagery
Call subroutine "CI_Interest_Field_Calc" to take the representative average channel Brightness Temperatures from each cloud object tracked from *t1* to *t2* and perform a series of combined spectral and temporal differencing tests to determine whether or not the object is likely to convectively initiate.
Output a "Binary Yes/No" forecast of CI for all tracked cloud objects in the form of a 2-dimensional image array that corresponds to the input imagery from *t2*

Figure 1. High Level Flowchart of the CI Algorithm, illustrating the main processing sections.

Draft

From Figure 1, it can be seen that the algorithm is broken into 5 main pieces of code. Below, is a summary of these components, and a more detailed explanation of these items will follow later in the document.

1. CI_Code_Main.f90
   The main "wrapper" code that coordinates the initialization of important arrays and parameters and calls the other necessary subroutines.

2. Object_Tracking.f90
   Subroutine that takes in the AWG Cloud Type output data, along with the 11.2µm ABI channel information, both for two consecutive satellite image scans, "*t1*" and "*t2*", labeled in chronological order. Then, it uses that information to define pre-convective "cloud objects" (see description of "Define_Objects.f90", below) and employs an iterative temporal-overlapping technique to track the objects from "*t1*" to "*t2*", assigning each object a unique integer identification number at each respective image time. The main output from this subroutine is a pair of 2-dimensional image arrays, one for each of the two input satellite image times, filled with tracked cloud objects that are represented by unique integer IDs. As an example, if an unbroken group of pixels in the "*t1*" output array is filled with the number "3", then the group of pixels filled with the number "3" in the corresponding "*t2*" array, though displaced, is considered to represent the same cloud object as in the first array. That particular cloud object has now been tracked between the two consecutive input images.

3. Define_Objects.f90
   Subroutine called at the beginning of "Object_Tracking.f90" to define pre-convective cloud objects, using a combination of the AWG Cloud Type output and the 11.2µm ABI channel data for times "*t1*" and "*t2*". The main purpose of this subroutine is to quickly and efficiently acquire a count of how many pixels belong to each defined object and, then, break up those objects deemed as "too large" (delineated by a parameter set at the top of subroutine) into smaller objects that are centered over localized areas of potentially convective elements embedded within the larger cloud objects.

4. ChannelAverage_Calc.f90
   This subroutine is called within the main wrapper code, "CI_Code_Main.f90", after object-tracking has been accomplished. It reads in all input ABI channel data for the given domain and input time, along with the corresponding "Object_Tracking.f90" output array of integer-identified tracked objects (either for "*t1*" or "*t2*", individually), then it calculates an average Brightness Temperature from each channel for each object at the given input image time. The averages are derived from only a subset of pixels belonging to each object, represented by the coldest 25% of 11.2µm (or equivalent) pixels from that given object. The reason for using such a subset is to hone in on potential developing updrafts, without washing out the signal over the entire cloud object and without

succumbing to possible noise-issues that might affect the single coldest 11.2 μm pixel of an object. All calculated averages are output to a tabular array for the corresponding input image time, such that each row represents a different ABI channel and each column number represents a tracked cloud object's unique integer ID number (using "fill values" where certain integer ID numbers were omitted in "Object_Tracking.f90" for various reasons). The subroutine is individually called two times, once for each of the consecutive input image sets, "*t1*" and "*t2*".

5. CI_Interest_Field_Calc.f90
   This subroutine takes in the two tabular object average Brightness Temperature arrays that were output from "Channel_Average.f90" (one for each input image time, "*t1*" and "*t2*"). Then, it performs a series of combined spectral and temporal differencing tests ("interest field" tests) to assess the potential that each cloud object is growing and/or is in a position to grow vertically through the troposphere. If enough of the interest field tests are passed for a given tracked cloud object, based on pre-determined test thresholds, then that cloud object is flagged for a "positive" CI forecast. If too few of the interest field tests are passed for a given tracked cloud object, then that cloud object is flagged for a "null" CI forecast. Therefore, the current CI forecast output is given in the form of a "Binary Yes/No" forecast for CI, regarding each cloud object. The output CI forecasts from this subroutine are given in a single 2-dimensinonal image array, filled with "0s" and "1s" that correspond to the placement of clouds in the "*t2*" input satellite imagery. The "0's" represent no objects and, therefore, no forecasts. The "1s" represent positive CI forecasts and fill the space occupied by entire individual cloud objects that were tracked and tested (these are the binary "Yes" CI forecasts, indicating that the cloud object is growing and will likely produce a radar detected echo of ≥35 dBZ within the next 2 hours).

3.3 Algorithm Input

This section describes the input needed to process the CI algorithm. This will include, both, the set of required ABI channels and any necessary ancillary data that is required to produce the desired product output.

3.3.1 Primary Imager Data

- The CI algorithm requires the use of the infrared Brightness Temperatures from ABI channels 8, 10-11, and 14-16 (Table 2). These channels are required for both the current image time and the previous image time in order to process all stages of the algorithm outlined in Figure 1.

- The Local Zenith Angle (LZA), also known as the "Satellite Zenith Angle", is required input, mainly for inclusion in the output Quality Flags, Product Quality Information, and product specific Metadata. It is necessary to pass on this LZA information, since the quality of the CI algorithm output is reduced at higher LZAs as a result of more cloud-side satellite detection rather than cloud-top satellite detection.

Draft

3.3.2 Ancillary Data

- The main ancillary data input for the AWG CI algorithm is the AWG Cloud Type Product. The current time dataset along with the previous time dataset are required for processing. Any cloud type data dependencies as outlined in the Cloud Type ATBD are also inherently and indirectly necessary for the CI algorithm.

- Furthermore, there are certain datasets needed primarily for the generation of Quality Flags (QF), Product Quality Information (PQI), and product specific Metadata. Specifically, these datasets include the Level 1B satellite channel PQI data output and the Cloud Type algorithm PQI data output, both of which should be automatically passed into the algorithm.

3.4 Theoretical Description

3.4.1 Physical Approach to the Problem

The CI algorithm tracks moving clouds using an object identification and tracking methodology, and it monitors the growth of the clouds using a spectral and temporal differencing technique that incorporates many of the infrared ABI spectral channels listed in Table 2. It is important to note that this algorithm is designed for identifying clouds, which have the potential for growth, thus mature clouds are omitted from processing. Many other studies have used similar methods for monitoring mature mesoscale convective complexes (Carvalho and Jones 2001, Machado and Laurent 2004, and Vila et al. 2008).

Cloud objects are identified using output from the AWG Cloud Type algorithm. If clouds are identified as water cloud, supercooled water cloud, or mixed phase cloud, those pixels are retained for further processing in the CI algorithm, since they are likely candidates for potential convective growth. Next, the algorithm iteratively searches around each pixel to determine whether or not there is a space between contiguous clusters of “cloud” pixels. This is the method used to group pixels into individual cloud objects. If any cloud object is deemed “too large”, the algorithm will break it into smaller objects, focusing on the potentially convectively active regions of the larger object. A predefined size threshold will be used to determine whether an object is too large, and a peak detection technique that uses the 11.2 μm channel will be used to extract the convectively active regions. This will also help to mitigate any false cloud detections by the AWG Cloud Type algorithm output, since regions of falsely identified large cloud objects will be removed if there are no minimum temperature peaks.

Once cloud objects are defined, the CI algorithm uses an object tracking technique, which employs a temporal overlap methodology similar to that used by Zinner et al. (2008). This simple overlap technique avoids the extra processing required by atmospheric motion derivations by exploiting the high temporal resolution of the GOES-R ABI. However, in its current state, the tracking algorithm does not perform well with fast-

Draft

moving clouds and suffers if the time between subsequent satellite images is significantly greater than 5 minutes.

After the cloud objects have been identified and tracked, the coldest 25% of 11.2 μm pixels within each tracked object are averaged for "t1" and "t2". To accomplish this, a "quicksort" routine is used to list all the 11.2 μm Brightness Temperature pixels in order from coldest to warmest for each cloud object, then, this subset of coldest 25% of pixels are averaged for each input spectral channel (Table 2), cloud object, and input image time. This subset of cold pixels is used as a focus for potential updraft regions in clouds.

Using the averaged object Brightness Temperatures as input from each object, time, and spectral channel, a series of infrared spectral and temporal differenced threshold tests are performed (sumarized in Table 3). These tests, or "interest fields", cover a combination of static spectral differencing from the most recent imagery alone—which provides information on current cloud-top height—and temporal differencing—which provides information about the rate of vertical cloud-top growth.

If an object meets 7 of the 12 spectral tests, then it is flagged as having a high likelihood for CI in the near future (i.e. the cloud object is forecast to produce a rainfall intensity of ≥35 dBZ on radar within the next 2 hours). In doing so, all the all pixels belonging to that object will be highlighted for CI, corresponding to the cloud object's position from the most recent input image time, "t2".

In the following sections, the four main components of the algorithm will be discussed in detail. The four main components are:

1) Cloud Object Identification
2) Object Tracking Methodology
3) Spectral Tests
4) CI Forecast Determination

Interest Field Tests	**Physical Basis (Mecikalski et al. 2010)**	**Critical Value**
6.15-11.2 μm	Cloud Depth	-30^{o}C to -10^{o}C
6.15-7.4 μm	Cloud Depth	-25^{o}C to -5^{o}C
11.2 μm	Cloud Depth/Glaciation	-20^{o}C to 5^{o}C
8.5-11.2 μm	Glaciation	-10^{o}C to -1^{o}C
Tri-channel Diff	Glaciation	-10^{o}C to 0^{o}C
5 min Tri-Channel	Glaciation Trend	>0^{o}C
5 min 12.3-11.2 μm	Cloud Depth	>0.5^{o}C
12.3-11.2 μm	Cloud Depth	-3^{o}C to 0^{o}C
5 min 11.2 μm	Cloud Growth	< -1.33^{o}C
5 min 6.15-7.4 μm	Cloud Depth Trend	>0^{o}C
5 min 6.15-11.2 μm	Cloud Depth Trend	>0.5^{o}C
13.3-11.2 μm	Cloud Depth	-20^{o} to -5^{o}C

Table 3. Spectral/temporal differencing tests and critical thresholds used in CI algorithm

Draft

3.4.1.1 Cloud Object Identification

The purpose of the Cloud Object Identification portion of the algorithm is to sort through the cloud elements identified by the AWG Cloud Type algorithm and identify candidate cloud objects that possess the potential for CI. Mainly, the routine is used to break up any large cloud objects into subsets of smaller, potentially convective elements. Additionally, the outer portions of many defined clouds are trimmed away using a dynamic Brightness Temperature threshold to reduce the risk of surface pixels, falsely identified as cloud, from being input for further processing in the CI algorithm. These tasks are handled by a subroutine known as "Define_Objects", which is called at the beginning of the "Object_Tracking" subroutine (Figure 1). For the input, it uses the AWG Cloud Type algorithm output and the ABI 11.2 μm array of Brightness Temperatures (individually, for the two consecutive input image times, "t1" and "t2"), and the output is a two-dimensional array of defined cloud objects (one array for each image time).

Certain parameters, such as a maximum object size (number of pixels) that defines what is considered as "too large" for an object, are set at the beginning of the routine. Another important parameter is the "Max_Object_Distance" parameter, which defines the box-radius of pixels surrounding any embedded convective elements that can be retained when overly large objects must be broken into smaller objects centered over these potentially convective cloud elements.

Below that section, the first functional portion of the algorithm copies the input Brightness Temperature array elements into a one-dimensional array and sorts them from coldest to warmest using a "quicksort" routine. The warmest 40% of 11.2 μm pixels across the entire domain will be disregarded from processing, in order to remove potential surface pixels from being defined as cloud objects (this is the dynamic Brightness Temperature threshold test). Only those pixels that remain colder than that threshold and still fall within the input AWG Cloud Type array are retained for further cloud object definition.

Cloud object definition then continues with the identification of the first object, starting with coldest retained point, referenced from the temperature-sorted one-dimensional array, and working down the list later. If it is determined that the pixel has not already been assigned to an object, the point is then assigned to a structure. This structure is designed to be used in a linked list that allows for tracking which points need to be visited and considered to join the object. These points are assigned further down in the algorithm and the double check that happens for "point #1" is the only check for any point to follow. The point is then checked for the desired type of cloud (water, super-cooled water, mixed phase), and its status as "not belonging to another object" is also confirmed. It is also checked again to make sure it falls within the group of pixels retained by the dynamic Brightness Temperature threshold test. If all these criteria are met, then the pixel is assigned a positive, non-zero counter number, and the pixels that are immediately up, down, left, and right of this point are added to the linked list as the next points to be checked and potentially added to the object. This process continues until all pixels belonging to the object have been discovered and assigned.

Draft

After the entire object has been identified (all contiguous pixels), the count of the total number of pixels belonging to this object is determined. If that count is larger than the maximum object size parameter, then the object is checked over again to identify the "most-defined peaks" in the cloud (i.e. the cooler, potentially convective elements embedded within a larger cloud object). This is determined by summing the difference between each object Brightness Temperature pixel and that of all the points around it within a given box-radius. Then, the average of those differences is used to quantify a Brightness Temperature "peak". The locations of the highest 10 peak magnitudes and nearby pixels are retained, and the rest of the pixels belonging to the overly large object are reset to a non-object status and flagged so that they will not be reconsidered later as part of an object.

Once this process is complete, the algorithm takes the second coldest point from the sorted one-dimensional array of Brightness Temperatures and iterates through until all potential cloud objects are identified. A two-dimensional array covering the domain is output, and all non-zero pixel values within that array are deemed to be potential CI cloud object pixels.

3.4.1.2 Object Tracking Methodology

The object tracking portion of the CI algorithm is based upon the simple concept of temporal overlapping. Because of this restriction, there is a known weakness in the method, such that, if the mean cloud motion is very fast and the object size is small, there may not be temporal overlap of the same clouds between the two input image times. However, this problem is mitigated by the fact that convectively growing will increase in horizontal size. Figure 2 shows the threshold line, identifying when an object may be missed from this method of tracking for a given object size and speed, assuming that the object size remains constant between times "t1" and "t2".

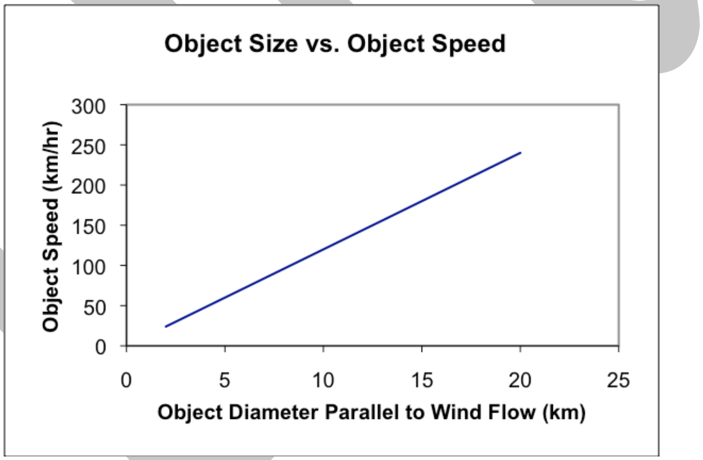

Figure 2. Threshold for simple temporal overlap object-tracking, given the speed and diameter of an object, parallel to the wind flow

Draft

Temporal overlap occurs when an object that occupies a space at Time 1 ("t1") can be assumed to be the same object at Time 2 ("t2") as long as its position at t2 partially coincides, or "overlaps", with part of the space it occupied at t1 (Figure 3).

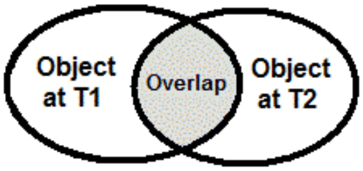

Figure 3. Schematic diagram showing how a single object can be tracked through time via the temporal overlapping technique.

Figure 4 shows a detailed flowchart of the "Object_Tracking" subroutine, which drives this portion of the algorithm. Note that the "Define_Objects" subroutine (mentioned in detail in the Cloud Object Identification portion of the algorithm description, above) is part of the "Object_Tracking" subroutine, and is called before any other object-tracking processing is performed.

The first step to applying this method to the tracking of cloud features is to identify all desired cloud types resulting from a trustworthy satellite-based cloud classification scheme, and then mask out all other irrelevant or undesired cloud types. This is accomplished through the AWG Cloud Type algorithm and the Cloud Object Identification portion of the CI algorithm, described above. In the given algorithm, near the beginning of the "Object_Tracking" subroutine, all potential CI cloud feature pixels from both t1 and t2 are assigned the integer, "-1" (these are the pixels that were assigned a positive, non-zero value in the "Define_Objects" subroutine as part of the Cloud Object Identification portion of the overall CI algorithm). Next, the resulting arrays are summed, so that all temporally overlapping regions can be identified wherever the integer "-2" is present (Figure 5). Non-overlapping region pixels are left with values of "-1" after the arrays are summed. Figure 6 better demonstrates this overlap region identification with real, scaled down output from the computer algorithm, using simplified input case data.

Draft

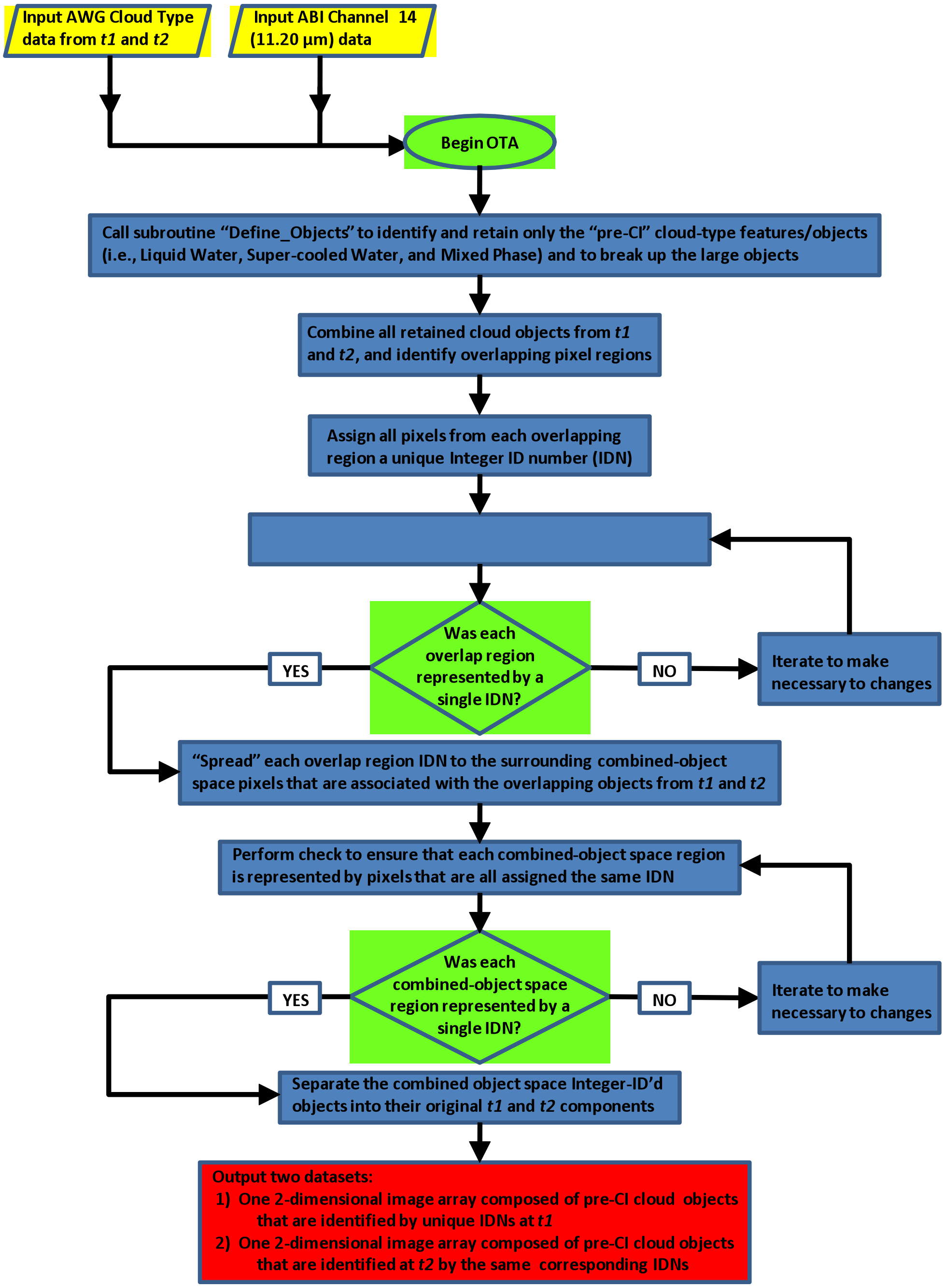
Input AWG Cloud Type data from *t1* and *t2*
Input ABI Channel 14 (11.20 µm) data
Begin OTA
Call subroutine "Define_Objects" to identify and retain only the "pre-CI" cloud-type features/objects (i.e., Liquid Water, Super-cooled Water, and Mixed Phase) and to break up the large objects
Combine all retained cloud objects from *t1* and *t2*, and identify overlapping pixel regions
Assign all pixels from each overlapping region a unique Integer ID number (IDN)
Was each overlap region represented by a single IDN?
YES
NO
Iterate to make necessary to changes
"Spread" each overlap region IDN to the surrounding combined-object space pixels that are associated with the overlapping objects from *t1* and *t2*
Perform check to ensure that each combined-object space region is represented by pixels that are all assigned the same IDN
Was each combined-object space region represented by a single IDN?
YES
NO
Iterate to make necessary to changes
Separate the combined object space Integer-ID'd objects into their original *t1* and *t2* components
Output two datasets:
1) One 2-dimensional image array composed of pre-CI cloud objects that are identified by unique IDNs at *t1*
2) One 2-dimensional image array composed of pre-CI cloud objects that are identified at *t2* by the same corresponding IDNs

Figure 4. Flowchart schematic of the "Object Tracking" Algorithm (OTA) subroutine

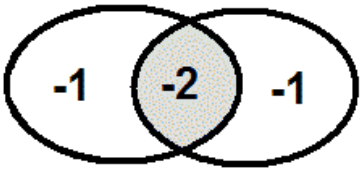

Figure 5. Schematic showing that the summing of object arrays at t1 and t2 result in values of "-2" for all overlap region pixels, when all object pixels are initially assigned values of "-1". The ellipse on the left represents an object at t1, while the ellipse on the right represents the same object at t2.

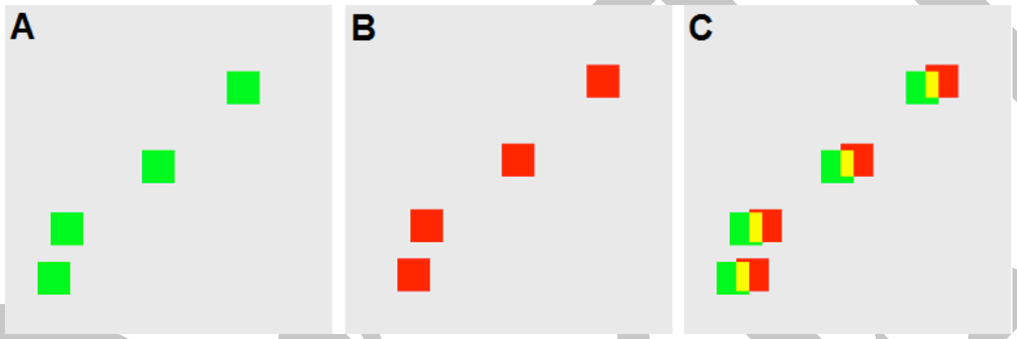

Figure 6. Illustration of how the overlap regions are identified. Object array pixels from t1 (a) and from t2 (b) are summed. The result is a single array of integers (c) with values of "0" where no objects exist (grey), values of "-1" where objects exist but there is no overlap (green or red), and values of "-2" where there is overlapping between objects from t1 and t2 (yellow).

The next step performed is to assign each individual overlap region a unique, positive integer identification number (IDN). The algorithm loops through the summed array, searching for overlap pixels, valued at "-2". Then, whenever the first group of overlap pixels is encountered, a counter, which is initialized at "0", is incremented up to "1", and each pixel in that overlap group is assigned an IDN of "1". The next time a separate group of overlap pixels is encountered, the counter is incremented up by one more integer value, and the process continues until each overlap region is assigned a unique integer IDN (Figure 7).

Draft

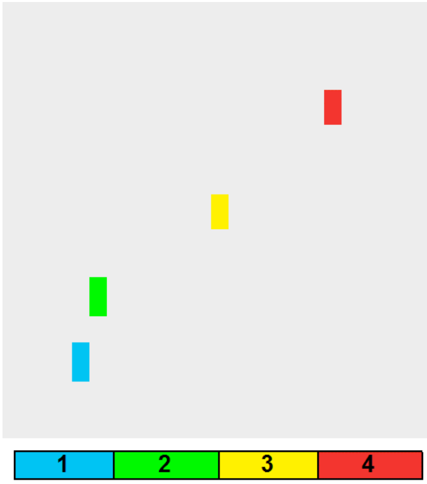

Figure 7. Continuing the example from Figure 6, the overlap regions have all been assigned unique integer IDNs.

lkoval 7/6/11 1:47 PM
Deleted:

The Object_Tracking subroutine then loops through the IDN overlap array, first, iteratively checking to make sure each contiguous overlapping group of pixels is all assigned the same IDN. Next, it iteratively "spreads" each overlap region's unique IDN left, right, up, and down across the entire space occupied by each object from t1 and t2 that is contributing to a given overlap region. The final product from this step is a single collective array, where the pixel space occupied by overlapping objects from both t1 and t2 are, accordingly, assigned an integer IDN that is unique for each object (Figure 8).

After one more series of iterations to make sure each set of overlapping object pixels is assigned the same unique ID number, the final step in the Object_Tracking subroutine is to separate the recently created t1/t2 collective array into its original t1 and t2 components (Figure 9). From there, it will be possible to monitor cloud top characteristic trends and other derived interest fields consistently for whole, individual objects.

Draft

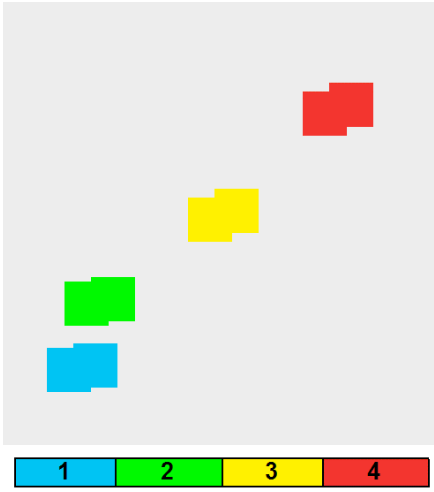

Figure 8. Illustration showing the result after “spreading” IDNs from the overlap regions to the rest of the space occupied objects both at T1 and at T2.

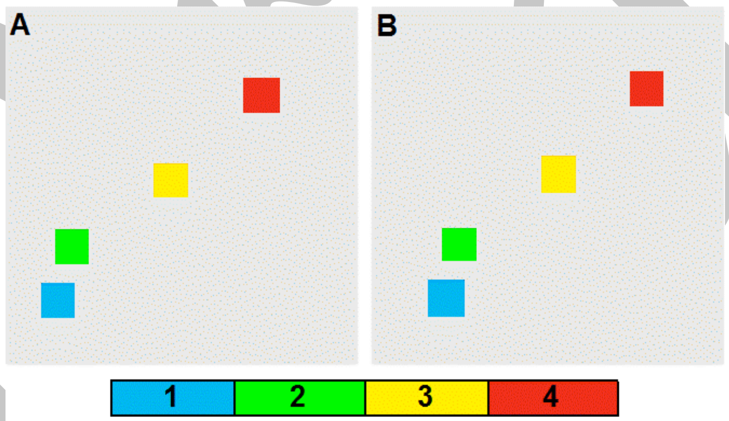

Figure 9. Final output from the Object Tracking portion of the algorithm. Each object has been assigned a unique integer IDN that remains consistent from t1 (a) to t2 (b).

lkoval 7/6/11 2:09 PM

Deleted:

Draft

3.4.1.3 Spectral Tests

In order to perform the spectral tests, representative Brightness Temperatures must first be selected for each object, for all spectral channels, and for both input image times (t1 and t2). This process is handled by the "ChannelAverage_Calc" subroutine. In this part of the algorithm, all pixels within each object are sorted from coldest to warmest using a "quicksort" routine, according to the 11.2 μm spectral channel. Then the 25% of coldest pixels from each object are averaged to derive a representative Brightness Temperature for that channel. It is assumed that using this subset of pixels from the 11.2 μm channel will help to better focus on any potential updraft regions within cloud objects; whereas, using an average of all pixels within an object may lead to a "washing out" of any CI spectral signals, and using the single coldest 11.2 μm pixel of each object may introduce CI signal errors from potential satellite instrument noise. However, in situations where an object is so small that 25% of the total count of its pixels is less than 1, the 11.2 μm Brightness Temperature from the single coldest pixel is used as the representative Brightness Temperature of that object. Once this subset is established for an object, then, the *same* subset of pixels is used to derive the representative average Brightness Temperature for each of the other input spectral channels, as listed in Table 2. This data is output into two separate tabular arrays, one for each image input time, t1 and t2.

After the representative Brightness Temperatures are established for each cloud object at both input times, the data is handed off to the "CI_Interest_Field_Calc" subroutine. For each object, each of the 12 spectral tests defined in Table 3 are performed. As can be seen from the table, this list of spectral tests, or CI "interest fields", is comprised of a combination of static spectral differencing—using the spectral data available from the most recent input imagery, t2—and temporal differencing—using spectral data from both input image times, t1 and t2, set at 5 minutes apart. If the calculated value from a given test falls within the range of critical threshold values for a cloud object, then the object receives a score of 1 for that test. Otherwise, the object receives a score of 0 for the given test. These scores are summed for each object to arrive at a total test score, so that every object receives a final score that ranges from 0 to 12.

3.4.1.4 CI Forecast Determination Overview

Empirically derived results have shown that when 7 or more of any of the above spectral tests have been passed, then there is a high likelihood for CI. When a given cloud object receives a summed spectral test score of at least 7, the "CI_Interest_Field_Calc" subroutine flags that object in one-dimensional array with a passing binary value of "1", indicating a positive forecast for CI. If the score is less than 7, the object is flagged with a failing binary value of "0". This one-dimensional array represents all tracked cloud objects, and each position in the array corresponds to its assigned unique IDN. So, the first element in this array represents tracked object #1, the second element represents tracked object #2, and so forth. Where an object IDN does not exist, a "fill value" is used, instead of a binary yes (1) or no (0) forecast for CI.

Finally, after all cloud objects have been tested, the information from this one-dimensional binary yes/no array of CI forecasts is copied into a two-dimensional image

Draft

format. This is accomplished by combining the placement of tracked cloud objects from the most recent input imagery, at t2, with information from the newly created one-dimensional array of CI forecasts. The result is a two-dimensional image covering the entire domain that is filled with pixel values of “0s” and “1s”. The pixels with values of “1” all belong to cloud objects that are forecast to convectively initiate in the near future, while the pixels with values of “0” represent, either cloud objects that are not forecast to convectively initiate (based on the given pair of input image data), or no tracked cloud objects at all. This type of two-dimensional image output allows for the final cloud object CI forecasts to be compared with or overlaid onto the most recent satellite imagery.

3.4.2 Physical/Mathematical Description

3.4.2.1 Interest Field Development

Meciklaksi and Bedka (2006) outlined several spectral CI interest fields, which can be used with the *current* GOES satellite instrument (Table 4). Note that there are fewer spectral tests available for current GOES, as opposed to with the GOES-R ABI. These interest fields provide information about cloud-top heights and the vertical growth of clouds. Knowledge of this information can lead to lower false alarm rates when attempting to forecast near-term CI, because knowing the vertical location of cloud-tops within the atmosphere will add to the information provided by cooling rates alone, as derived from the 11.2 μm channel on the ABI (or equivalent, such as the 10.7 μm channel aboard current GOES). Mecikalski and Bedka (2006) show that the spectral information provided by the current GOES satellite can yield such information and allows for effective monitoring of growing cumulus clouds. Validation efforts with the CI algorithm have revealed that having even more spectral information available from the GOES-R series of satellite imagers will facilitate the use of more cloud property information, which can further reduce the number false alarms.

CI interest field	Critical value
10.7μm T	0°C
10.7 μm T time trend	< -4 °C (15 min)$^{-1}$
6.5-10.7 μm difference	-35 °C to -10 °C
13.3-10.7 μm difference	-25 °C to -5 °C
12.0-10.7 μm difference (used for GOES-11 only)	-3 °C to 0°C
6.5-10.7 μm time trend	> 3°C (15 min)$^{-1}$
13.3-10.7 μm time trend	> 3°C (15 min)$^{-1}$
12.0-10.7 μm time trend (used for GOES-11 only)	> 2°C (15 min)$^{-1}$

Table 3. Operational GOES interest fields used within the current, proxy AWG CI algorithm

Draft

The 6.5-10.7 µm spectral difference provides information on cloud top height location, relative to the tropopause (Mecikalski and Bedka 2006). Typically the difference is negative because the near surface temperature—where the 10.7 µm weighting function peaks—is warmer than the mid- to upper-troposphere, where the water vapor channel weighting function peaks. A positive difference corresponds to clouds at or above the tropopause (Ackerman 1996; Schmetz et al. 1997). This information can identify clouds which are immature (e.g. cumulus humilus) or which have grown only into the low- to mid-levels of the atmosphere. The temporal trend of this interest field allows for the vertical growth of the cloud to be monitored over time. Essentially, this field allows for the determination of how fast the cloud-top is moving upward through the troposphere. Such information cannot be retrieved by using the 6.7 µm spectral channel, alone, because its weighting function peak is so high up in the troposphere.

The 12.0-10.7 µm spectral difference, known as the "split window" technique, is typically used for identifying the presence of cirrus, volcanic ash, and deep convective clouds. Inoue (1987) has found that near-zero 12.0-10.7 µm spectral differences provide a means to identify areas of convective rainfall. This is an enhancement to the Griffith et al. (1978) method, which employs a ≤20^{o}C 10.7 µm threshold Brightness Temperature. When the value of this spectral difference is slightly negative, the cloud-top has not yet reached a height where convective rainfall is likely occurring, yet it is in a position where rainfall will likely develop from the cloud in the near future. The purpose for this interest field is to highlight areas that are evolving into a convective rainfall cloud. The temporal trend of this spectral channel allows a more effective approach to monitoring this transition.

The 13.3-10.7 µm spectral difference provides information about growing cumulus clouds, as Mecikalski and Bedka (2006) found that this spectral difference has different characteristics when derived from mature cumulus and for small, immature cumulus clouds, similar to the 6.5-10.7 µm spectral difference. Mecikalski et al. (2008) found, using a principal component analysis, that the 13.3-10.7 µm channel is one of the most important interest fields because of the unique information it added. It is hypothesized that this spectral difference was found to show so much value, ironically, because of the relatively poor 8 km spatial resolution of the 13.3 µm channel found on the current GOES instrument. When the 13.3 µm channel saturates, then it is very likely that a cloud will convectively initiate, because the cloud will have had to have grown significantly to achieve saturation on such a coarse spatial scale. Since the 13.3 µm channel will have a spatial resolution of 2 km on GOES-R, it is uncertain how much added information this channel will provide, due to the relatively few studies performed on this spectral channel.

Both, Siewert et al. (2009) and Mecikalski et al. (2010) provide a detailed explanation of the best uses for the infrared satellite fields for pre-convective clouds. Siewert et al. (2009) discusses how to use Meteosat Second Generation (MSG) Spinning Enhanced Visible Infrared Radiometer Imager (SEVIRI) data for CI forecasting purposes over South Africa, using a different tracking methodology than that which is used by the AWG CI algorithm. In that study, the importance of using multiple spectral tests within a CI algorithm is highlighted and demonstrated. Mecikalski et al. (2010) examined all the

Draft

possible spectral tests and divided them into three physical categories: 1) Cloud depth, 2) Cloud-top glaciation and 3) Updraft strength. From these three physical categories, tests were performed to determine which spectral tests are redundant and which ones contain the most information, the latter of which are included in the CI algorithm. This study provided the spectral interest fields (or spectral tests) that are currently being used in the GOES-R AWG CI algorithm. The purpose of using the additional spectral channels available from MSG is to exploit as much information as can be provided about the three physical categories mentioned above. Additionally, the results from that study also provided the necessary information needed to develop the critical threshold values for the newest set of spectral tests, included in the CI algorithm. Some slight modifications were required, however, in order to account for the change from the 15-minute temporal resolution available from MSG to the 5-minute temporal resolution that will be available with the GOES-R ABI. Yet, these changes were simple to implement, and will allow the spectral tests to provide the same amount of useful information for forecasting CI in the GOES-R era.

The Mecikalski and Bedka (2006) algorithm is currently running operationally, using the spectral channels available from the current GOES series instruments. That algorithm has a high probability of detection (upwards of 90% when all interest field thresholds are met), however, the high false alarm rates are relatively high (Mecikalski et al. 2008). The high false alarm rates result, largely, from a pixel-based tracking and verification method, but an object-based tracking and verification technique allows for better probability of detection and false alarm rate statistics, as we have found from the validation of the current GOES proxy version of the AWG CI algorithm. This is because the object-based method allows for better accuracy in tracking, and it is far less stringent when comparing CI forecast with validation radar data. Furthermore, the GOES-R ABI will allow for the addition of more spectral interest fields in order to help constrain such false alarm rates.

3.4.2.2 Binary CI Forecast Determination

Empirical results have shown that using 7 or greater of the 12 spectral tests that will be available with the GOES-R ABI as a CI forecast cutoff line yields optimal statistics for determining a binary "Yes/No" forecast for CI. Using 184 cases over Europe during the summer of 2007, statistical tests were performed to determine this optimal CI test score threshold, covering the full range of possible positive spectral test scores (1 through 12). Table 5 contains the statistical accuracies derived when each of the possible scores was used as a minimum for determining the binary CI forecasts. Accuracy is defined as the sum of "Hits" plus "Correct Negatives", divided by the total of all four values within the validation contingency table (see Table 6, below). Notice that the statistical accuracy is maximized when 7 is used as the minimum spectral test score (number of passed spectral tests) for determining a positive forecast for CI.

Draft

Number of spectral tests passed	Accuracy
1 or greater	57.7%
2 or greater	60.56%
3 or greater	61.97%
4 or greater	67.6%
5 or greater	69.95%
6 or greater	76.06%
7 or greater	**80.75%**
8 or greater	73.24%
9 or greater	53.05%
10 or greater	48.83%
11 or greater	43.2%
12 or greater	42.25%

Table 4. Comparison of the minimum spectral test scores required for a positive CI forecast and the resulting impacts on statistical accuracy.

Dichotomous Forecast Verification	
Was CI Forecasted? **YES** Did CI Occur? **YES** ***Hits***	Was CI Forecasted? **YES** Did CI Occur? **NO** ***False Alarms***
Was CI Forecasted? **NO** Did CI Occur? **YES** ***Misses***	Was CI Forecasted? **NO** Did CI Occur? **NO** ***Correct Negatives***

Table 5. Dichotomous forecast verification contingency table.

3.4.3 Algorithm Output

The final output of this algorithm is a two-dimensional binary array that indicates which cloud objects are likely to convectively initiate in the near future (generating rainfall at an intensity great enough to produce a ≥35 dBZ radar reflectivity). In this output array, all pixels within objects forecast to CI will be highlighted in the form of a mask at 2 km spatial resolution, so that it matches the ABI infrared image data resolution.

Quality Flags, Product Quality Information, and Metadata will also be included as output. The Quality Flags and Product Quality Information are stored as 2 arrays. The elements of both arrays are structures, with each structure containing logical variables for each "bit" of information.

Draft

Because the main input for the AWG CI algorithm is the Level 1B ABI channel data and the AWG Cloud Type data, much of the information to be included in the CI algorithm output of Quality Flags, Product Quality Information, and product specific Metadata (Tables 10-12) comes from the Quality Flags and Product Quality Information data that is output from those two datasets. Therefore, that information must be passed into the main CI algorithm for processing and generation of the CI algorithm quality output data. Furthermore, the LZA satellite data must also be passed into the algorithm, so that it can be used to set the correct bits in the Quality Flags and Product Quality Information output. The variable structures for this input quality information have been declared in the code, but because the data has not yet been made available, the flag checks for this information have not yet been included.

Quality Flags

Bit 1:	0=good data, 1=bad or missing data exists from any of the 4 Quality Flags below (bits 2-5)
Bit 2:	0=good Level 1B data, 1=bad Level 1B data
Bit 3:	0=cloudy, 1=clear
Bit 4:	0= Local Zenith Angle (LZA) ≤ 65 degrees, 1= LZA > 65 degrees
Bit 5:	0=data is present, 1=missing data

Table 7. AWG CI algorithm output Product Quality Information

Product Quality Information

Byte	Bit	Flag	Source	Value
0	0	Local Zenith Angle block-out zone	L1B	1=local zenith angle>65° or lat>66°; 0=OK
	1	Cloud Type Algorithm Input	Cloud Type	1=bad data; 0=OK
	2	Level 1B data	L1B	1=bad data; 0=OK
	3	Pre-convective Cloud Object Flag	CI	1=No Cloud Object; 0=Cloud Object
	4	CI Yes/No	CI	1=No CI Likely; 0=CI Likely
	5-7	Not Used		
1	0-7	Number of CI Interest Fields Triggered	CI	Number of CI Interest Field Triggered within Object ranging from 0 to 12

Table6. AWG CI algorithm output Product Quality Information

Draft

Metadata

Percent of domain affected by "bad" Level 1B data
Percent of domain affected by "bad" Cloud Type data
Percent of domain in Local Zenith Angle block-out zone (>65 degrees)
Total number of output Tracked Cloud Objects
Average number of pixels within all output Tracked Cloud Objects
Average number of passed spectral tests for all objects
Average value of each spectral test calculated over all objects (total of 12 test values)

Table 7. AWG CI algorithm output product specific Metadata

Near the end of the main wrapper code, “CI_Code_Main”, outside of much of the main CI algorithm processing, the internal algorithm data needed to generate the output quality and metadata information is checked and processed. The first few lines of code initiate the logical flags in each array. In the FORTRAN syntax, the “TRUE” logical statement equates to the binary bit set of “1”, while the “FALSE” logical statement equates to the binary bit set of “0”. While iterating through the entire two-dimensional domain, each pixel of the input cloud type data is checked to determine if it is a cloud pixel. If it is deemed cloudy, the appropriate Quality Flag bit is set accordingly. Then, based on whether or not that pixel is part of an actual processed “cloud object”, the appropriate bit is set for the Product Quality Information that gives information on whether or not the pixel is considered part of a “Pre-convective Cloud Object”. Then using the array of information about the number of passed spectral tests, 4 bits are set in the second byte of Product Quality Information that represent the binary form of the number of passed spectral tests for that pixel. Also, there is a check to determine whether or not the given pixel belongs to an object that produced a positive CI forecast, and then the corresponding bit in the structure of the Product Quality Information array is set accordingly.

Certain Metadata is also gathered and calculated as secondary processes within various subroutines during the algorithm’s run. This data, which is passed back to the “CI_Code_Main” program for output display, is listed in Table 12. Some of this information includes the total number of tracked objects, the average number of pixels belonging to tracked objects (averaged over all tracked objects), and the average number of passed spectral tests for all objects. Also given, are the average values of the spectral test calculations for all tracked objects. This actually produces 12 different lines of information, one for each average test value, and there is no regard for the pass/fail test thresholds for this section of Metadata. Additionally, just like the Quality Flag and Product Quality Information, some of the Metadata output is dependent upon the data passed into the algorithm, such as quality information on the Level 1B ABI channel and cloud type data, as well as the LZA data. This information is used to calculate what percentage of the domain is affected by bad data. As with some of the quality information data, the output variables for these calculations have been declared, but no processing has yet to be implemented in the code, since the necessary input data has not yet been made available.

Draft

4. TEST DATA SETS AND OUTPUTS

4.1 Simulated/Proxy Input Data Sets

Three unique data set types have been used as proxies to run and to validate the AWG CI algorithm:

1) MSG SEVIRI data over Europe with 5-minute temporal resolution and available radar data within the domain.

2) Regional Atmospheric Modeling System (RAMS) Model simulated ABI radiances with 5-minute temporal resolution and simulated radar reflectivities (courtesy of the Cooperative Institute for Research in the Atmosphere; CIRA).

3) Current GOES-East data over the CONUS with 15-30 minute temporal resolution and available WSR-88D radar data within the domain (CI forecasts generated by the AWG CI proxy algorithm, which is limited by the relatively poor spectral, temporal, and spatial resolutions of the current GOES instrument).

While much validation has been performed with the second two proxy datasets listed, still much more needs to be performed using our primary ABI proxy, the MSG SEVIRI instrument. Research is currently underway to expand this group of validation statistics, using the spectral channels aboard this instrument. Nevertheless, the exhaustive validation studies that have already been performed show that the current AWG CI algorithm and method work very well, meeting and going beyond all required specifications.

4.2 Output from Proxy Data Sets

The output from the proxy datasets will resemble the planned output for GOES-R. The main differences are that MSG SEVIRI data has a horizontal spatial resolution of 3 km at nadir, and the infrared channels available from the current GOES instrument have spatial resolutions of 4 km. Also, when statistics are derived using these datasets, it is important to note the coarser spatial resolution may cause some areas of small-scale convective initiation to be missed. Figures 10 and 11 demonstrate sample output from the algorithm, using proxy data from the MSG SEVIRI instrument and the GOES-13 instrument, respectively (the latter, using the AWG CI proxy algorithm). Since radar data was not made available for the MSG case, 10.8 µm infrared satellite imagery is used to show how the clouds significantly deepened where the algorithm forecasted CI from two of the processed cloud objects.

Draft

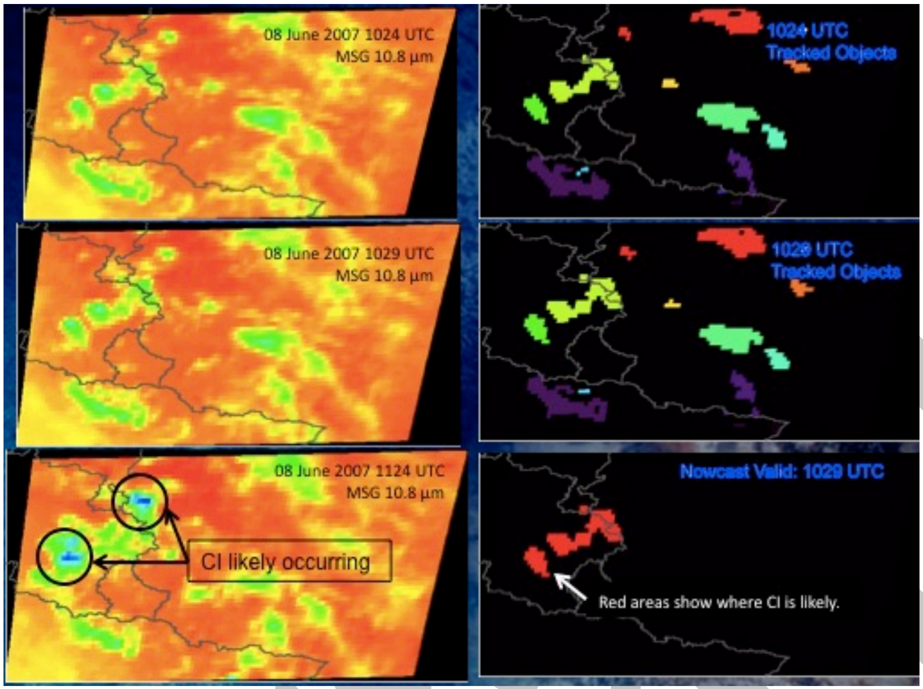

Figure 10. Examples of CI Algorithm output forecasts from 5 minute MSG SEVIRI data over southern Germany from 08 June 2007. The top two images are valid at time 1 (1024 UTC), the middle two images are valid at time 2 (1029 UTC). For the top two rows, the images on the left are the 10.8 µm channel data, and the images on the right show the defined cloud objects (the different object colors represent different unique integer ID numbers). The bottom row image on the right is algorithm output from the 1029 UTC input set of images, and the bottom row image on the left is the actual 10.8 µm imagery from nearly 1 hour into the future, valid at 1124 UTC.

Draft

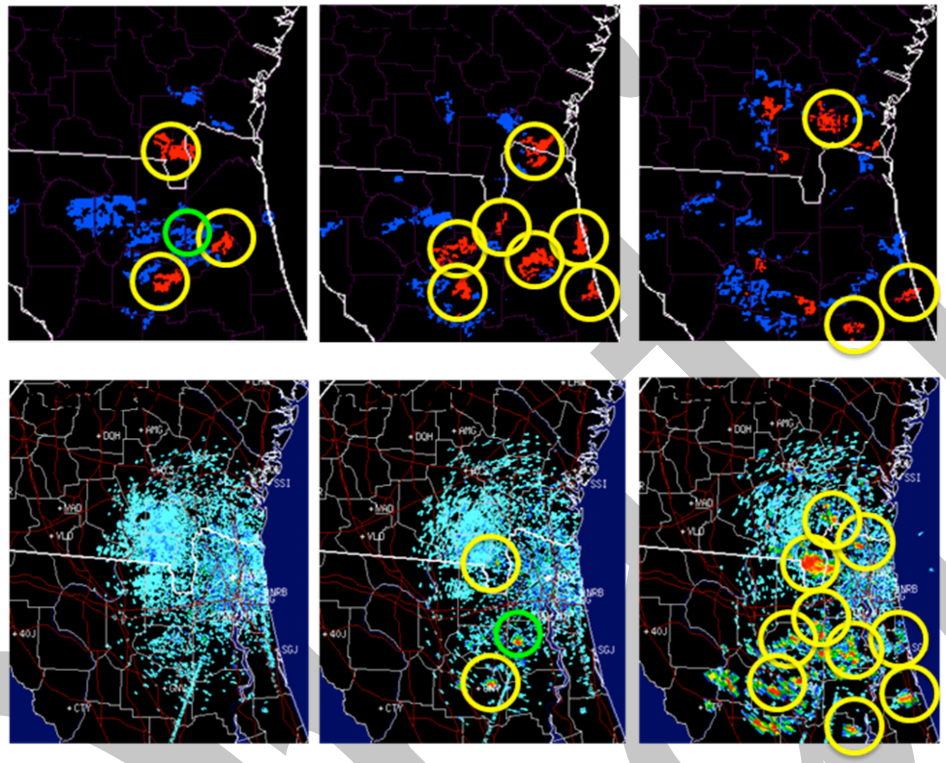

Figure 11. CI forecast output and validating WSR-88D radar data corresponding to the AWG CI proxy algorithm near the Georgia/Florida border in the U.S. on 06 June 2011. The images of forecast output and radar data in the far left column are both valid for ~1702 UTC, indicating that no rainfall was being detected at the time of the initial set of CI forecasts. The images of forecast output and radar data in the middle column are both valid for ~1715 UTC The output forecast image at the top-right is valid from the 1732 UTC satellite input, while the radar image at the bottom-right is valid at nearly 1 hour after the initial CI forecasts, at 1758 UTC. The red objects indicate positive CI forecasts, the blue objects indicate "null" CI forecasts, the yellow rings indicate "hits", and the green ring indicates a "miss" (since there was no positive CI forecast made for the storm cell that formed in that location).

Draft

4.2.1 Algorithm Validation

The validation of the CI algorithm is object-based and performed subjectively. Since we are concerned with whether a particular cloud will convectively initiate, a cloud object-based approach to validating the algorithm is the best approach, as this will give accurate statistical information on the algorithm performance.

lkoval 7/6/11 3:32 PM
Deleted:

The validation strategy includes a full contingency-based statistical analysis. Since validation of this product is a dichotomous forecast, a 2x2 contingency table validation can easily be performed. Not only can the probability of detection (POD) and false alarm ratio (FAR) be derived by this approach, but also the probability of false detection (false alarm rate), the bias score, and, most importantly, the statistical accuracy (required to be >70% for this algorithm) can be calculated. Deriving all of these variables will allow for a more complete validation, which is important for the purpose of developing and testing the robustness of any forecast algorithm.

To accomplish this, all CI object forecasts (including null forecasts) are compared with up to 2 hours of radar data in a domain encircling a given radar site with a radius of ~75 km. A set radius is used to ensure consistency and to make sure that no low-topped convective events are missed due to overshooting of the radar beam at greater distances from the radar site. Also, the ≥ 35 dBZ threshold is used from radar reflectivity to determine what is classified as a CI event, as is commonly found in the literature (Mecikalski and Bedka 2006, Mueller et al. 2003, Roberts and Rutledge 2003). Because the validation is approached subjectively, it is usually quite easy to determine which occurrences of ≥ 35 dBZ radar echoes are convective in nature, as opposed to stratiform or melting layer radar echoes. Furthermore, this approach allows for easy identification of new CI events without confusing them with convection that is ongoing.

Base radar reflectivity is used, as this allows for a larger radial coverage for validation within a given distance of a radar site without overshooting any potential developing storms farther away from the radar; furthermore, it is not uncommon to use low-level radar reflectivities in validation studies (Mueller et al. 2003). More importantly, however, it is difficult enough to obtain any radar data at all from within the European domain, which resides under the coverage of the currently best available GOES-R ABI proxy, the MSG SEVIRI instrument. Many times, the only radar that could be retrieved from this region came from the base reflectivity scans in the form of GIF image files. Because the validation studies have extended to other regions, such as the U.S., where radar data is freely available, it was deemed best to remain consistent with our validation approach, even in situations where a higher radar elevation scan or composited reflectivity would have favored the algorithm's performance. Therefore, base radar reflectivity has been adopted and used consistently for the CI algorithm validation studies.

To identify the hits, misses, false alarms, and correct negatives needed to fill the contingency table, the forecast objects are compared to radar data in the following manner, accounting for mean cloud object motion beyond the time of the forecasts:

Draft

- Positive forecasts of CI upstream of corresponding radar-detected CI events are considered as "HITS".
- Null forecasts of CI just prior to and upstream of corresponding radar-detected CI events are considered as "MISSES".
- Positive forecasts of CI upstream of non-existent CI events are considered as "FALSE ALARMS".
- Null forecasts of CI upstream of non-existent CI events are considered as "CORRECT NEGATIVES".
- Diagnostic (aka: "Zero-Lead time") and Negative Lead Time forecasts are considered as "MISSES". The first step of validation is to check if a CI forecast is given for convection that has already initiated

As mentioned before, the MSG SEVIRI instrument provides the best ABI proxy dataset for the CI algorithm. Yet, the validation dataset presented herein shows very little in the way of results from using this data as input for the algorithm. A very large issue that has been encountered with validating the CI algorithm with MSG data is gaining access to radar data over Europe. So far, for the validation study, we have only been able to gain access to radar data for several convective days over southern Germany during the summer of 2007. Nevertheless, we will continue our attempts to gain access to radar data within this region so that the validation statistics will be more meaningful and complete.

To mitigate some of the issues with gaining access to radar data over Europe, the CIRA group at Colorado State University has performed several RAMS model simulations for convective days and provided ABI simulated datasets to run as input in the CI algorithm. Model derived simulated radar reflectivity has been used to validate the algorithm for the case days that were provided. Furthermore, the current GOES proxy version of the AWG CI algorithm has also been used to assist with validation efforts. This has been a tremendous help, since GOES data and WSR-88D radar data are so readily available over the U.S. For this, several convective days were used from the summer of 2010 over various regions in the U.S., including coastal Florida, the central Great Plains, the Mid-south, and the Northeast. The logic for including the current GOES proxy algorithm in these validation efforts has been, if the output statistics meet the required specifications with the algorithm being applied to the relatively reduced capabilities (lower spectral, spatial, and temporal resolutions) of the heritage GOES instrument, then the algorithm will likely exceed the required specifications by a larger margin when run with the more advanced GOES-R ABI instrument. With this combined strategy to provide more significant validation statistics, a total of 14,671 CI forecasts have now been validated for this algorithm.

4.2.2 Validation Results

An exhaustive validation study has shown that the AWG CI algorithm exceeds the F&PS requirement of greater than 70% statistical accuracy, which is defined as the sum of "Hits" plus "Correct Negatives", divided by the total of all four values within the validation contingency table (refer to Table 6, above). Three different datasets were used to evaluate and validate the algorithm performance.

Draft

For our first proxy dataset, 184 CI forecasts were generated by the algorithm using input from 5-minute MSG SEVIRI data, covering several convective days in the summer of 2007 over southern Germany. The contingency table showing the results from this validation study can be found in Table 7.

Hits **107**	**False Alarms** **20**
Misses **16**	**Correct Negatives** **41**

Table 10. Contingency table from validation using MSG SEVIRI data

The second validation dataset came from two convective case days of simulated ABI channel data, generated by the RAMS model. Using this dataset allowed for testing with high spatial and temporal resolution data that will match the resolutions of the operational ABI data. The case days were simulated for 08 May 2003 and 27 June 2005, which were convectively active days over the central U.S. Great Plains and the front range of eastern Colorado, respectively. Table 8 shows the contingency table of results from this dataset, which encompasses a total of 4,544 validated CI forecasts. Note the high number of False Alarms and the low number of Misses from this validation dataset. It is unclear why the algorithm produced such extreme results with the model-simulated data. However, this was not found to be the case when real-world data was used for validation.

HITS **427**	**False Alarms** **1044**
MISSES **18**	**Correct Negatives** **3055**

Table 11. Contingency table from validation using RAMS simulated datasets.

The third dataset used to validate the algorithm's approach to forecasting CI in the near-term came from the current GOES-East satellite (GOES-13), which was used as input to the AWG CI proxy algorithm (designed to work with the coarser temporal, spatial, and spectral resolutions of the current GOES imager). Several case days from many diverse locations within the eastern U.S. GOES domain were taken from the summer of 2010 to facilitate this validation study. A total of 9,943 CI forecasts were validated for this dataset, and the results are given in the contingency table in Table 9. It worth noting that the results from this particular validation study most likely suffer, in a relative sense, from the use of the current GOES imager, because of the significantly reduced resolutions and capabilities of this heritage instrument. As indicated in the sections above, the higher temporal, spatial, and spectral resolutions provided by the GOES-R ABI instrument will considerably improve the performance of the AWG CI algorithm.

Draft

Hits **255**	**False Alarms** **308**
MISSES **99**	**Correct Negatives** **9281**

Table 12. Contingency table validation results using current GOES and the proxy CI algorithm

4.2.3 Error Budget and Accuracy Estimates

The F&PS statistical requirement for the AWG CI algorithm validation is an accuracy of 70% or greater. From the three different independent validation datasets listed above, the accuracy was 80.4% for the MSG dataset, and 76.6% for the RAMS model-simulated dataset, and 95.9% for the current GOES instrument dataset. These numbers all show that the CI algorithm meets and exceeds the given 70% F&PS accuracy requirement. In terms of lead time, the range was anywhere from 3 minutes up to 102 minutes, with an average of ~27 minutes. This value was derived from the time difference between the "current" input satellite image timestamp and the radar image time of the first ≥ 35 dBZ radar echo, used to validate each CI forecast.

5. PRACTICAL CONSIDERATIONS

5.1 Numeric Computational Considerations

In some situations, the object tracking code may take longer to run when there is large number of defined objects, or if any of the objects are extraordinarily large in size. However, this may be mitigated by future code enhancements and the advent of faster computer processors that will be available in the coming years.

5.2 Programming and Procedural Considerations

Since the given Object Tracking method requires the use of the AWG Cloud Type algorithm output, it is necessary that it be processed first. Once output from that algorithm is received, Cloud Object Tracking and CI Interest Field calculations can be performed.

Furthermore, the Cloud Object Tracking and CI Interest Field calculation portions of the code require the use of data from the current image time *and* the previous image time. Therefore, the ABI channel data and Cloud Type algorithm data from the previous image time need to be stored for use as input in the "current" run of the algorithm.

Draft

5.3 Exception Handling

Errors in the program are handled in the cases of reading data, writing data, allocating memory, deallocating memory, and opening files. They are handled by using the "STAT" and "IOSTAT" arguments available as intrinsic functions in FORTRAN and in other forms for many other programming languages. When the status number that STAT or IOSTAT returns is anything other than 0, the program sets a given return variable to 2, prints a description of the error out to the standard output, and returns control to whichever part of the code called it. The return variable is passed all the way back up, such that whenever an error occurs the entire algorithm ends and returns control up to the main calling routine or ends the program. It should be noted that this method of error handling does not cease the processing of any other processes outside of the CI algorithm, itself.

5.4 Graceful Degradation

It goes without saying that, because the CI algorithm requires input from both the Level 1B ABI channel data and the output from the Cloud Type algorithm, it is important that both datasets be as error-free and complete as possible in order for the output CI forecasts of the CI algorithm to be as accurate as possible. There is very little room for error, with respect to producing a high-level product such as this. The algorithm could possibly run alright with less than ideal results if one or, perhaps, even two of the required input ABI channels were missing, since the algorithm runs 12 separate spectral tests from a combination of 6 input ABI channels and only needs 7 out of 12 tests to be passed in order to produce a positive CI forecast for any given cloud object. However, missing cloud type information could prove much more detrimental to the algorithm's performance, since fundamental steps of Cloud Object Identification and Object Tracking are much more dependent on this input.

However, a couple of fall back methods could be used in certain situations where data is missing or known to be distorted to a point beyond use. In the case where the 11.2 µm channel data is not available, the 10.35 µm ABI channel could act as a reasonable substitute, so long as this data was used for both image input times, "t1" and "t2". This is possible since the weighting functions between these two channels are so similar (Schmit et al. 2005). Another possible fall back method might be used when the cloud type data is unavailable. Though, this would be a less ideal scenario, a simple Brightness Temperature threshold could be used with either the 11.2 µm or the 10.35 µm channel to generate a temporary cloud mask in these situations. Once again, it would be necessary for this type of cloud mask to be generated and input for both image times, "t1" and "t2". Also, because the Cloud Object Identification part of the algorithm retains only the cloud types of "water cloud", "super-cooled water cloud", and "mixed phase cloud" (represented in the cloud type data arrays by the integers, 2, 3, or 4, respectively), it would be a requirement to fill the substitute cloud mask with one of these integer values where ever "cloud" pixels are determined in the mask.

Draft

6. ASSUMPTIONS AND LIMITATIONS

The following sections describe the current limitations and assumptions in the current version of the CI algorithm.

6.1 Performance

The following assumptions have been made in developing and estimating the performance of the CI algorithm. This list contains the current assumptions and proposed mitigation strategies.

- The primary assumption relating to this algorithm is that significantly vertically growing clouds, as detected from only two input reference points in time, will continue to grow to a point in the near future, such that they will produce at least moderate intensity convective rainfall. However, there are several instances where clouds grow to a certain point and, then, stop well short of producing any rainfall. All other potential errors aside, this produces false alarms that cannot be avoided with the current algorithm's approach to solving the problem. Nevertheless, the developers believe that this is the best "stand-alone" satellite-based method for working toward a solution to the issue of near-term CI forecasting. Perhaps, other datasets, such as Numerical Weather Prediction short-term stability forecasts, can be added to the operational algorithm at a later time to help mitigate such potential false alarms issues, which would improve the validation statistics even more.
- We assume that the WSR-88D radar reflectivity values are accurate. The WSR-88D radar available from the National Climatic Data Center (NCDC) is quality controlled and, thus, the risk is mitigated.
- For correction of the satellite parallax issue, we assume that the data used for cloud-top pressure from GOES sounder is accurate. This has been validated in the literature and, for our purpose of radar-based validation, the error is minimal.
- It is assumed that, in general, clouds, which are growing vertically over time, are also growing horizontally. Yet, there could be missed CI events in situations where the object sizes are relatively small and are moving fast (refer to Figure 2). However, it is unlikely that any small-scale clouds will actually produce significant convective rainfall in most situations.

6.2 Assumed Sensor Performance

It is assumed that the sensor will meet its current specifications. However, the algorithms will be dependent on the following instrumental characteristics.

- Errors in navigation will affect the ability of the temporal overlap Object Tracking technique to identify overlap regions.

Draft

- 5-minute temporal resolution is required as a minimum for effective object tracking.
- Sensor accuracy is important since the spectral threshold tests require accurate measurement especially when using 5-minute data. The changes within cloud top Brightness Temperatures will be on the order of the sensor accuracy over 5 minutes in most cases.

6.3 Pre-Planned Product Improvements

Currently taking place is the examination of a methodology used with the current GOES instrument (adapted from Zinner et al. 2008) to track objects using a motion vector field. This would allow for a more effective operation of the CI algorithm, even at a coarser temporal resolution (up to 30 minutes). It would require the input of the GOES-R AWG Derived Motion Winds (DMW) within the CI algorithm. Research is ongoing within this area, exploring the possibility of implementing the DMW data within the Object Tracking portion of the CI algorithm.

Draft

7. REFERENCES

Ackerman, S.A., 1996: Global Satellite Observations of Negative Brightness Temperature Differences between 11 and 6.7 µm. *J. Atmos. Sci.*, **53**, 2803–2812

Carvalho, L.M.V., and C. Jones, 2001: A Satellite Method to Identify Structural Properties of Mesoscale Convective Systems Based on the Maximum Spatial Correlation Tracking Technique (MASCOTTE). *J. Appl. Meteor.*, **40**, 1683–1701.

Inoue, T., 1987: An instantaneous delineation of convective rainfall area using split window data of NOAA-7 AVHRR. *J. Meteor. Soc. Japan*, **65**, 469-481.

Machado, L.A.T., and H. Laurent, 2004: The Convective System Area Expansion over Amazonia and Its Relationships with Convective System Life Duration and High-Level Wind Divergence. *Mon. Wea. Rev.*, **132**, 714–725.

MacKenzie, W.M., C. Siewert, J.R. Meciklaksi, J.R. Walker, and E.W. McCaul, 2008: Enhancements to nowcasting convective initiation within the 0-1 hour timeframe. In Preparation for *Wea. Forecasting*.

Mecikalski, J.R., and K.M. Bedka, 2006: Forecasting Convective Initiation by Monitoring the Evolution of Moving Cumulus in Daytime GOES Imagery. *Mon. Wea. Rev.*, **134**, 49–78.

Mecikalski, J.R., K.M. Bedka, S.J. Paech, L.A. Litten, 2008: A Statistical Evaluation of GOES Cloud-top Properties for Nowcasting Convective Initation. *Mon. Wea. Rev.*, In Press.

Mecikalski, J.R., W. M. MacKenzie, Jr., M. Koenig, and S. Muller, 2010: Cloud-top properties of growing cumulus prior to convective initiation as measured by Meteosat Second Generation. Part 1: Infrared Fields. *J. Appl. Meteor.*, **49**, 521–534.

C. Mueller, T. Saxen, R. Roberts, J. Wilson, T. Betancourt, S. Dettling, N. Oien, J. Yee, 2003: NCAR Auto-Nowcast System. *Wea. Forecasting*, **18**, 545-561.

Roberts, R.D., and S. Rutledge, 2003: Nowcasting Storm Initiation and Growth Using GOES-8 and WSR-88D Data. *Wea. Forecasting*, **18**, 562–584.

Timothy J. Schmit, Mathew M. Gunshor, W. Paul Menzel, James J. Gurka, Jun Li, A. Scott Bachmeier, 2005: Introducing the Next-Generation Advanced Baseline Imager on GOES-R. *Bull. Amer. Meteorol. Soc.,* **86**, 1079-1096, doi: 10.1175/BAMS-86-8-1079

Schmetz, T.J., S.A. Tjemkes, M. Gube, and L. van de Berg, 1997: Monitoring deep convection and convective overshooting with METEOSAT. *Adv. Space Res.*, **19**, 433-441.

Draft

Siewert, C. W., M. Koenig, and J. R. Mecikalski, 2009: Application of Meteosat second generation data towards improving the nowcasting of convective initiation. Meteor. App., doi:10.1002/met.176

Vila, D.A., L.A.T. Machado, H. Laurent, and I. Velasco, 2008: Forecast and Tracking the Evolution of Cloud Clusters (ForTraCC) Using Satellite Infrared Imagery: Methodology and Validation. *Wea. Forecasting*, **23**, 233–245.

Zinner, T., H. Mannstein, and A. Tafferner, 2008: Cb-TRAM: Tracking and monitoring severe convection from onset over rapid development to mature phase using multi-channel Meteosat-8 SEVIRI data. *Meteorol. Atmos. Phys.*, doi:10.1007/s00703-008-0290-y.

Draft