Using McIDAS-V Libraries for Data Analysis and Visualization

Joleen Feltz[*], Mat Gunshor[*], Tim Schmit[+],

Tom Whittaker[*]

*Cooperative Institute for Meteorological Satellite

[+]NOAA/NESDIS Satellite Applications Branch

In cooperation CIMSS Scientists and McIDAS Programmers

9-12 September 2013: **McIDAS Users' Group Meeting**

CIMSS

Acknowledgements:

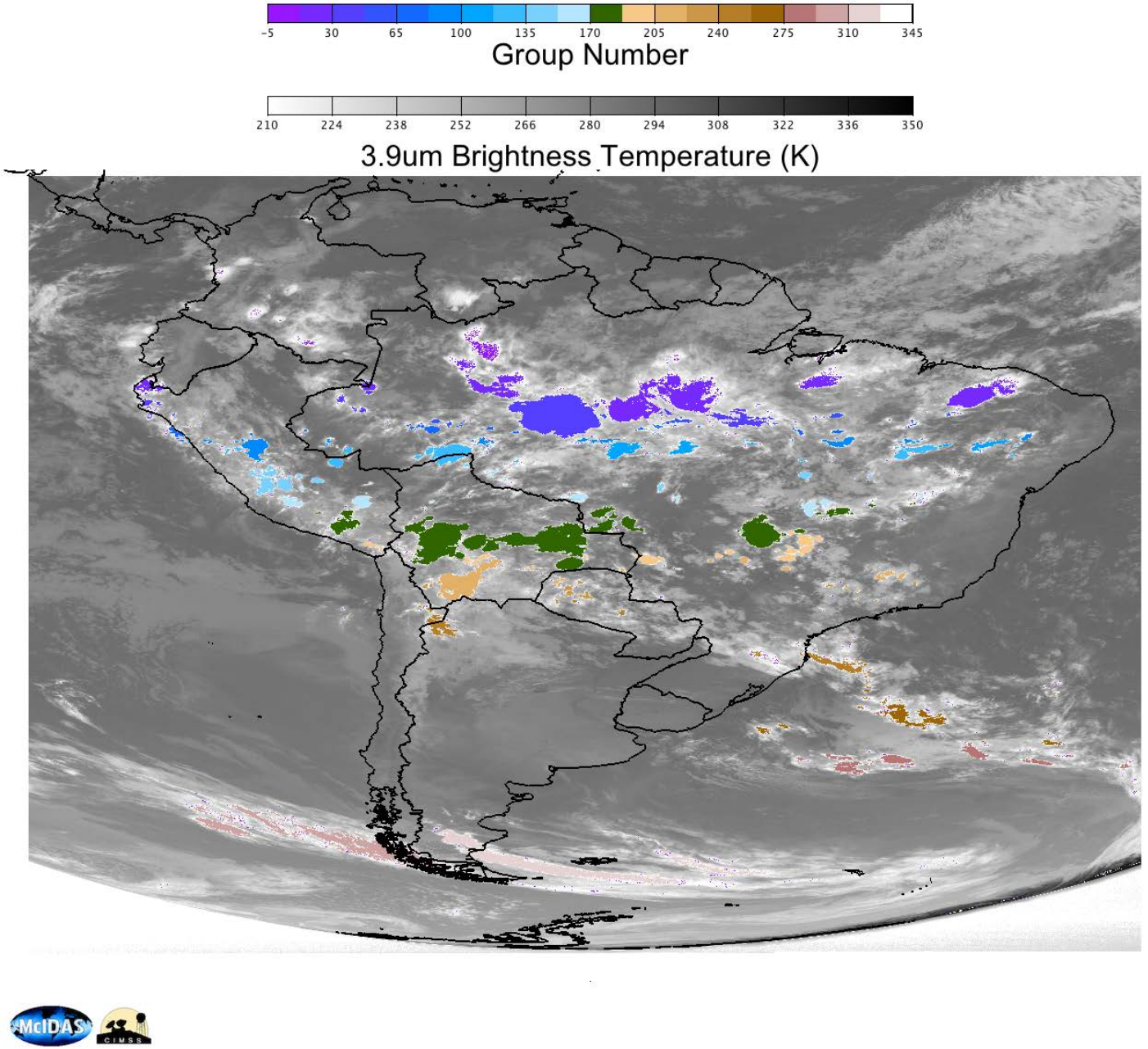
- McIDAS–V Team: Tom Rink, Tommy Jasmin, Mike Hiley, Jonathan Beavers, Bob Carp, Dave Parker
- SSEC System Administrator: Jesse Stroik
- Ghansham Sangar, Hans Peter Roesli
- CIMSS Scientist: Elisabeth Weisz, Nadia Smith, Bill Smith, Bob Knuteson, Jim Nelson, Tony Schreiner, Justin Sieglaff, John Cintineo, Steve Ackerman, Eric Tobias (UW-Madison AOS Student)
- NOAA/NESDIS Satellite Applications Branch: Gary Wade

Outline

- Analysis routine which visually highlights features of interest in satellite data and produces a simple output file

- Automated Data Analysis run on all GOES-13 Images in real-time. (Jython Shell using jar files packaged with McIDAS-V)

- Visualizations completed for education and scientific reporting

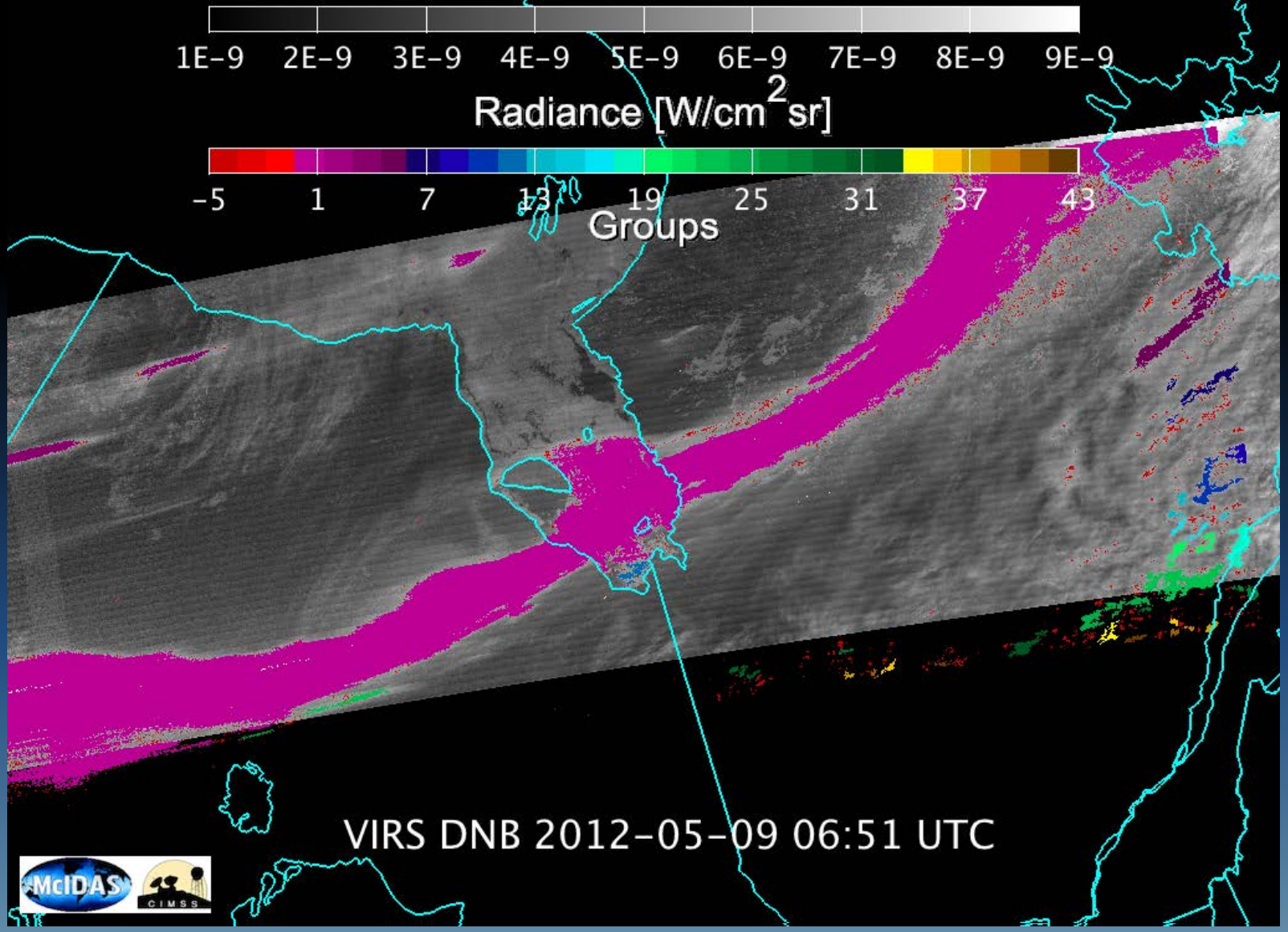


Grouping of brightness temperature values outside of range 71-339 K
GOES-12 3.9um Brightness Temperature (K): 2012-02-10 04:58:00Z

Imagery and Data Analysis

Group Number	Total Area (m2)	numPixels	mean	Lat[start]	Lon[start]	Lat[mid]	Lon[mid]	Lat[end]	Lon[end]
2012-02-10 04:58:00 UTC									
1	147.2589712	15	0	1.53566909	-75.357071	1.42623091	-75.399773	1.35316122	-75.333817
2	92.65638828	10	0	0.11205493	-64.906273	0.07577693	-64.885567	0.00323442	-64.947655
3	176.1312733	19	0	0.07578032	-65.051147	-0.0330385	-65.092575	-0.1055823	-65.009773
4	120.6602287	13	0	-0.1418774	-65.569099	-0.2144309	-65.527664	-0.2507086	-65.527664
5	398.883029	43	0	-0.1418653	-65.278984	-0.286967	-65.299736	-0.3595207	-65.299797
6	222.478528	24	0	-0.1781299	-65.071907	-0.2506742	-65.051208	-0.4320556	-65.134163
7	565.9910421	61	0	-0.3957936	-65.258392	-0.5772307	-65.527954	-0.7223119	-65.320892
8	1155.001211	120	0	-0.5429213	-72.866409	-0.8343191	-72.995979	-0.9798816	-72.825119
9	10736.96166	1159	0	-0.6497166	-65.134361	-1.4113888	-64.370087	-2.282505	-64.331322
10	101.7600107	11	0	-1.1573471	-64.080231	-1.2299068	-64.08036	-1.2661723	-64.039101
11	129.8846979	14	0	-1.3028287	-65.197716	-1.3754114	-65.218605	-1.4480064	-65.239532
12	123.5669079	13	0	-1.8879265	-49.361935	-1.9971249	-49.339924	-2.0698833	-49.360538
13	92.95865154	10	0	-2.1742427	-65.490906	-2.2469459	-65.574242	-2.2832561	-65.553665
14	4166.442819	448	0	-2.2108805	-65.947754	-2.5739572	-65.638115	-3.0465508	-65.703201
15	95.42088509	10	0	-2.8352158	-48.928989	-2.9445093	-48.948902	-2.9809947	-48.927204
16	93.0037899	10	0	-3.0461907	-65.308716	-3.1553521	-65.413132	-3.1552954	-65.350868
17	95.26773453	10	0	-3.1992068	-49.17918	-3.2719178	-49.284233	-3.3083792	-49.283737
18	152.6764135	16	0	-3.2724433	-49.008415	-3.3817627	-49.049347	-3.418314	-49.006378
19	106203.7537	11405	0	-3.2993293	-56.388458	-5.4110026	-57.785797	-7.1306581	-58.504448
20	15153.37458	1590	0	-3.3449216	-49.240829	-3.9638002	-49.782936	-4.5106688	-50.091568
21	111.1260414	12	0	-3.3718767	-63.280998	-3.4081855	-63.198463	-3.4445691	-63.239952
22	176.9060316	19	0	-3.4100223	-65.477119	-3.5191779	-65.49865	-3.5918941	-65.457596
23	24662.82551	2653	0	-3.4085183	-63.715416	-4.3555293	-64.528961	-5.1931696	-63.539207
24	41507.21691	4091	0	-3.7353632	-40.608757	-4.977469	-42.322823	-6.0083323	-42.306866
25	7288.411738	700	0	-3.7778544	-80.782448	-4.291965	-80.523911	-4.8097129	-80.910187

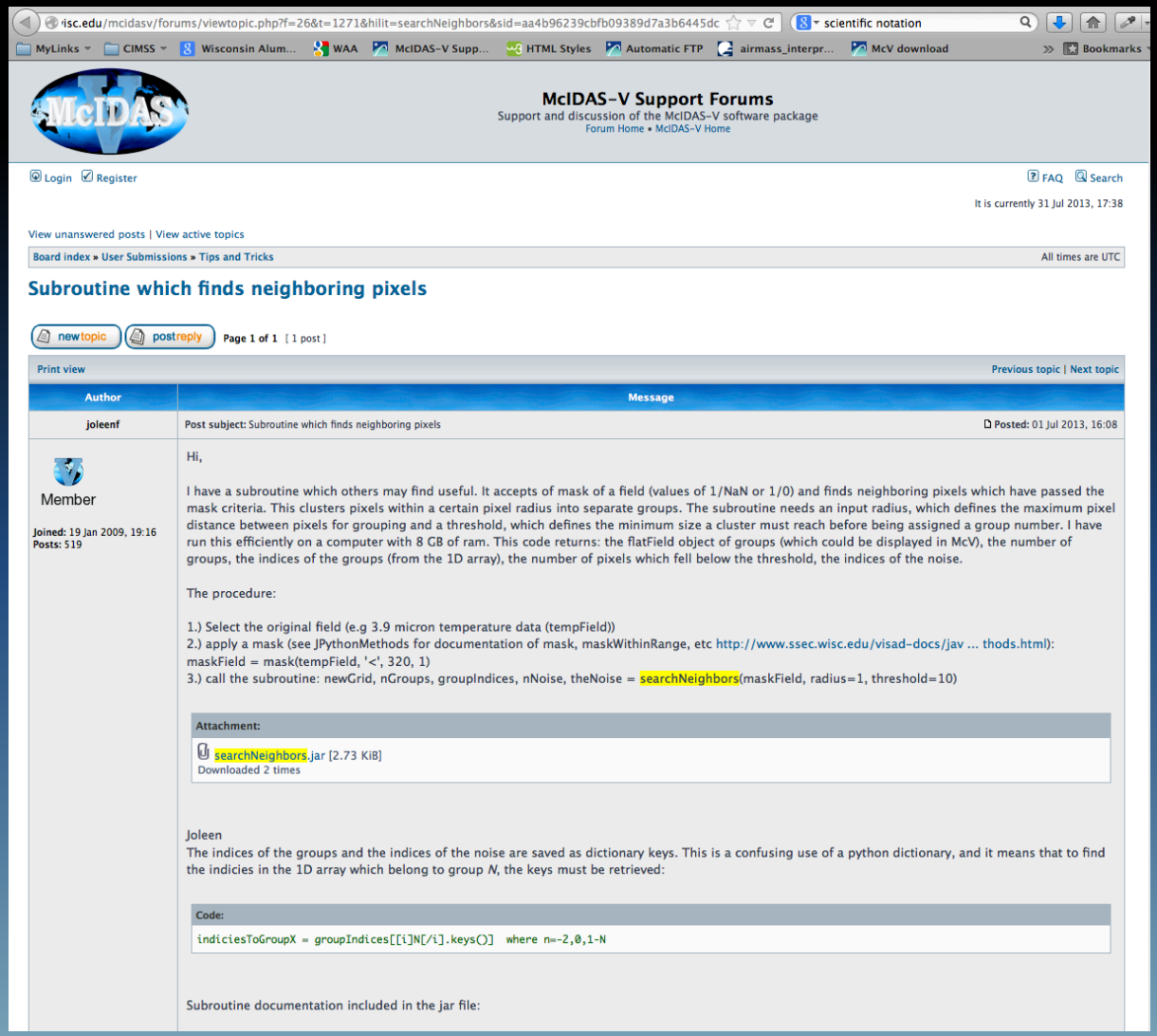
Visual and Statistical Data Analysis



Eric Tobias: UW-Madison AOS Undergraduate Student

Supervisor: Steve Ackerman

searchNeighbors.jar



isc.edu/mcidasv/forums/viewtopic.php?f=26&t=1271&hilit=searchNeighbors&sid=aa4b96239cbfb09389d7a3b6445dc

scientific notation

MyLinks | CIMSS | Wisconsin Alum... | WAA | McIDAS-V Supp... | HTML Styles | Automatic FTP | airmass_interpr... | McV download | Bookmarks

McIDAS-V Support Forums

Support and discussion of the McIDAS-V software package

Forum Home • McIDAS-V Home

Login | Register | FAQ | Search

It is currently 31 Jul 2013, 17:38

View unanswered posts | View active topics

Board index » User Submissions » Tips and Tricks — All times are UTC

Subroutine which finds neighboring pixels

new topic | post reply | Page 1 of 1 [1 post]

Print view | Previous topic | Next topic

Author	Message
joleenf	Post subject: Subroutine which finds neighboring pixels — Posted: 01 Jul 2013, 16:08

Member

Joined: 19 Jan 2009, 19:16
Posts: 519

Hi,

I have a subroutine which others may find useful. It accepts of mask of a field (values of 1/NaN or 1/0) and finds neighboring pixels which have passed the mask criteria. This clusters pixels within a certain pixel radius into separate groups. The subroutine needs an input radius, which defines the maximum pixel distance between pixels for grouping and a threshold, which defines the minimum size a cluster must reach before being assigned a group number. I have run this efficiently on a computer with 8 GB of ram. This code returns: the flatField object of groups (which could be displayed in McV), the number of groups, the indices of the groups (from the 1D array), the number of pixels which fell below the threshold, the indices of the noise.

The procedure:

1.) Select the original field (e.g 3.9 micron temperature data (tempField))
2.) apply a mask (see JPythonMethods for documentation of mask, maskWithinRange, etc http://www.ssec.wisc.edu/visad-docs/jav ... thods.html):
maskField = mask(tempField, '<', 320, 1)
3.) call the subroutine: newGrid, nGroups, groupIndices, nNoise, theNoise = searchNeighbors(maskField, radius=1, threshold=10)

Attachment:

searchNeighbors.jar [2.73 KiB]
Downloaded 2 times

Joleen
The indices of the groups and the indices of the noise are saved as dictionary keys. This is a confusing use of a python dictionary, and it means that to find the indicies in the 1D array which belong to group *N*, the keys must be retrieved:

Code:

```
indiciesToGroupX = groupIndices[[i]N[/i].keys()]  where n=-2,0,1-N
```

Subroutine documentation included in the jar file:

Automated Processing

Jython Shell

File Edit Help

```
inputField = selectData()

try:
   dataDate = inputField[0].getSampleTime()
except:
   dataDate = None

myMask = maskOutsideRange(inputField[0], [210,350])
imageLayer = activeDisplay().createLayer('Image Display', inputField)
maskLayer = activeDisplay().createLayer('Image Display', myMask)

groupNumber = reportCluster(myMask, inputField, radius=1, threshold=10, fileName='/Users/joleenf/test2.csv',fileDate=dataDate)
groupLayer = activeDisplay().createLayer('Image Display', groupNumber)

groupLayer.setEnhancement("Panoply>Group 1>haxby", range=[-5,230])
imageLayer.setEnhancement("Inverse Gray Scale", range=[210,350])
```

```
starting at  1373476919.61
SearchNeighbors Subroutine Begin
Domain and Range:  2440 1802
In searchNeighbors
SearchNeighbors Subroutine End
For all masked groups area =  4085764.45425
For groups above threshold =  4017344.14879
For individual pixels and clusters below threshold (aka:  Noise) =  52071.9988184
Group numbers with size of zero:  [] Numbers groups below threshold of  10  pixels:  736
Done in  19.2860000134  seconds
```

Evaluate:

```
inputField = se

try:
   dataDate = i
except:
   dataDate = N

myMask = maskOu
imageLayer = ac
maskLayer = act

groupNumber = r
groupLayer = ac

groupLayer.setE
imageLayer.setE
```

McIDAS-V

File Edit Display Tools History Bundles Window Help

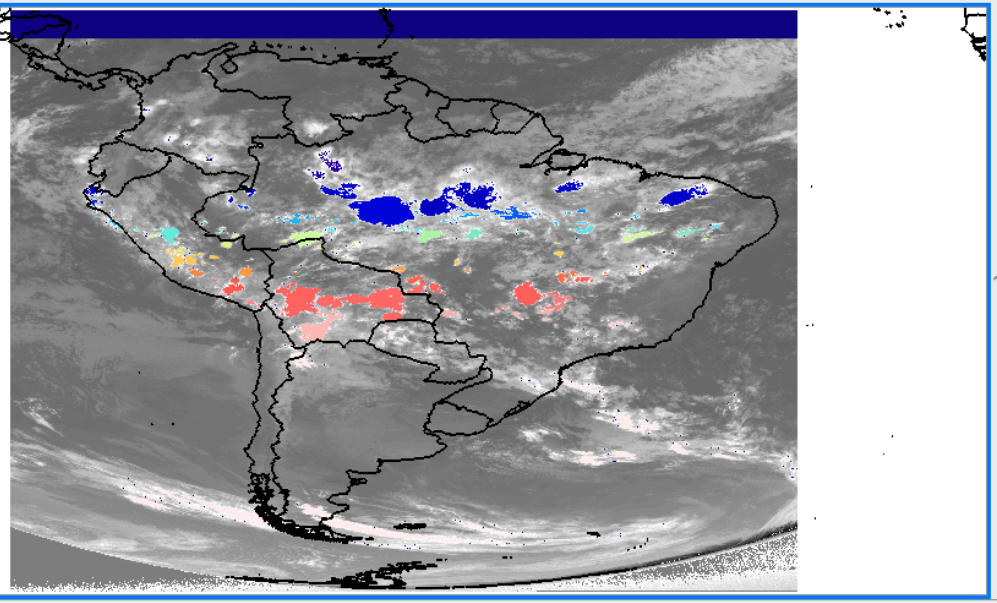
4Panel BlackBG WhiteBackground

1 untitled ×

View Projections

Panel 1

2012-02-10 04:58:00 UTC



Legend

Maps

Default Backgroun...

McIDAS World Country Outlines
North & Central America

Imagery

– Image Display

-5 230

– Image Display

1 1

– Image Display

210 350

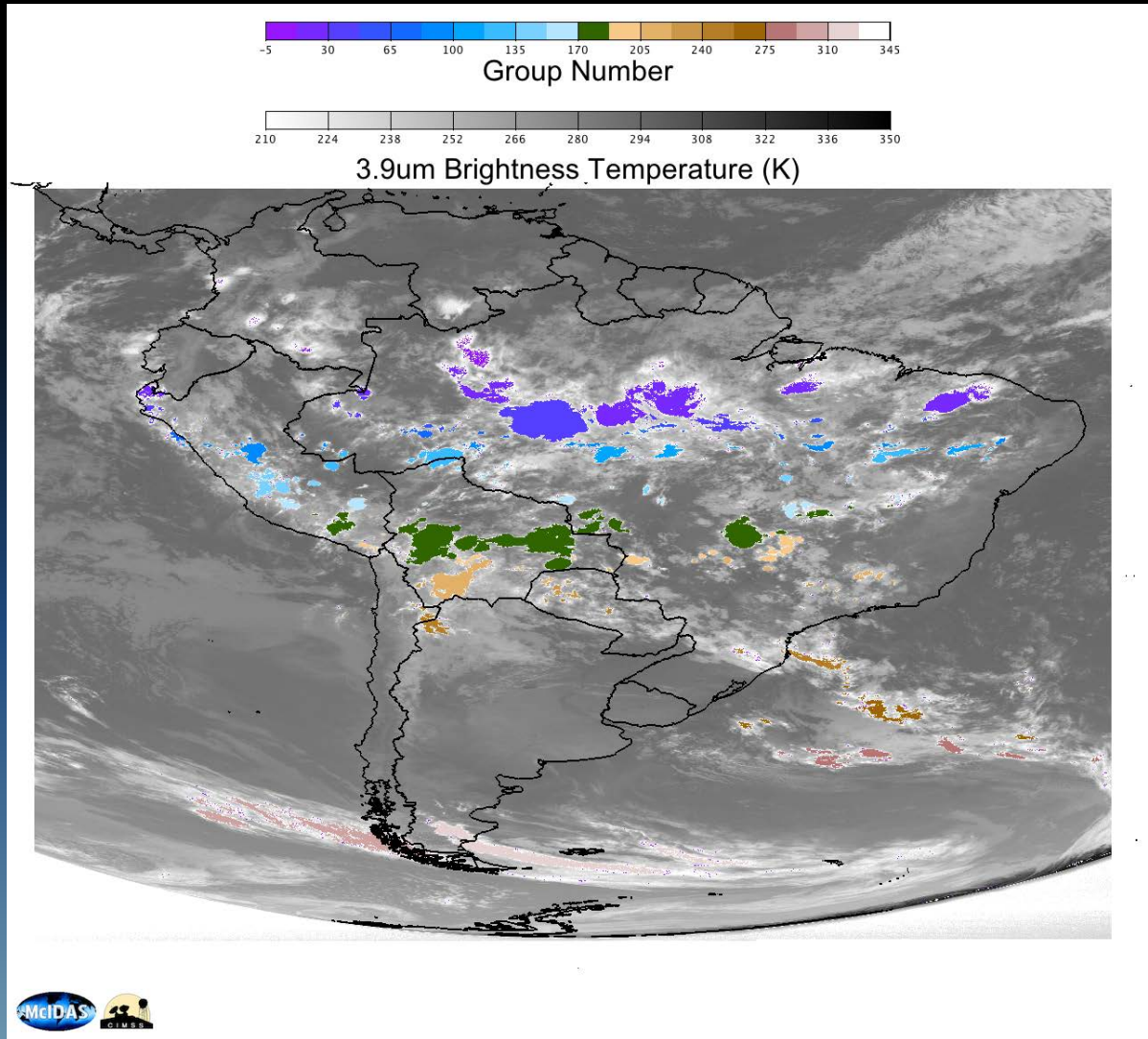
Memory: 1547/2494/6537 MB Latitude: 16.0 Longitude: -43.2 Altitude: 0.0 m

Automated Data Analysis

- CRON: sh <path>/runJython.sh <path>/reportAreaDiag.py --group=EASTL --desc=ALL --server=eastl.ssec.wisc.edu --pos
- runJython.sh: java -mx3000m -Dj3d.debug=true -Dpython.home=${home}/ McIDAS-V/jython2.5.2 -cp ${CLASSPATH} org.python.util.jython $1 $2 $3 $4 $5 $6 $7
- reportAreaDiag.py: uses recognizable ADDE commands at input. Run either with position number or date/time combinations. Uses area directory information.

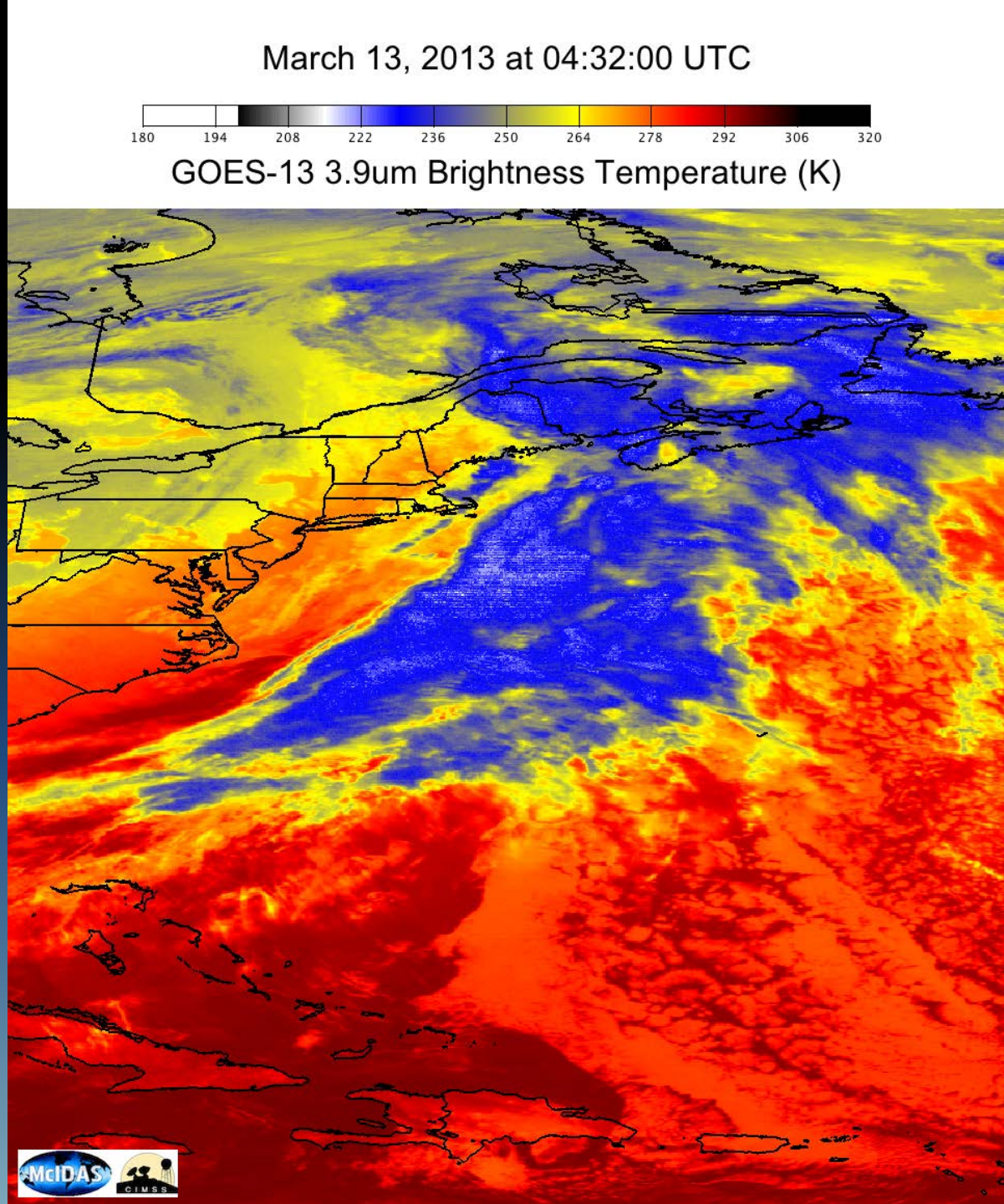
Variables Calculated

- 3.9 μm brightness temperature correlation between image times
- Percentage of image out of normal range (160 – 365 K)
- Area Coverage of pixels out of normal range (m^2)

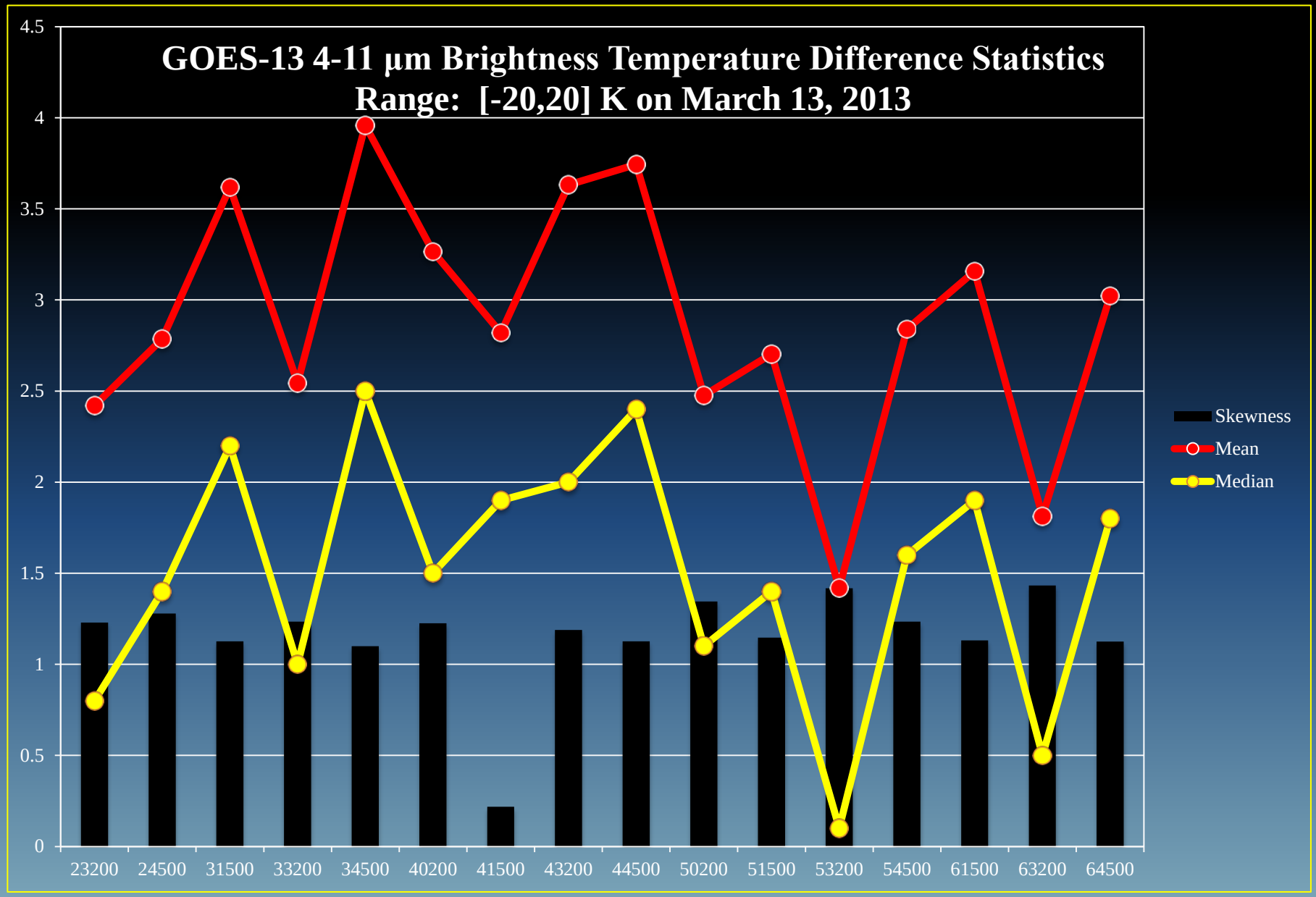


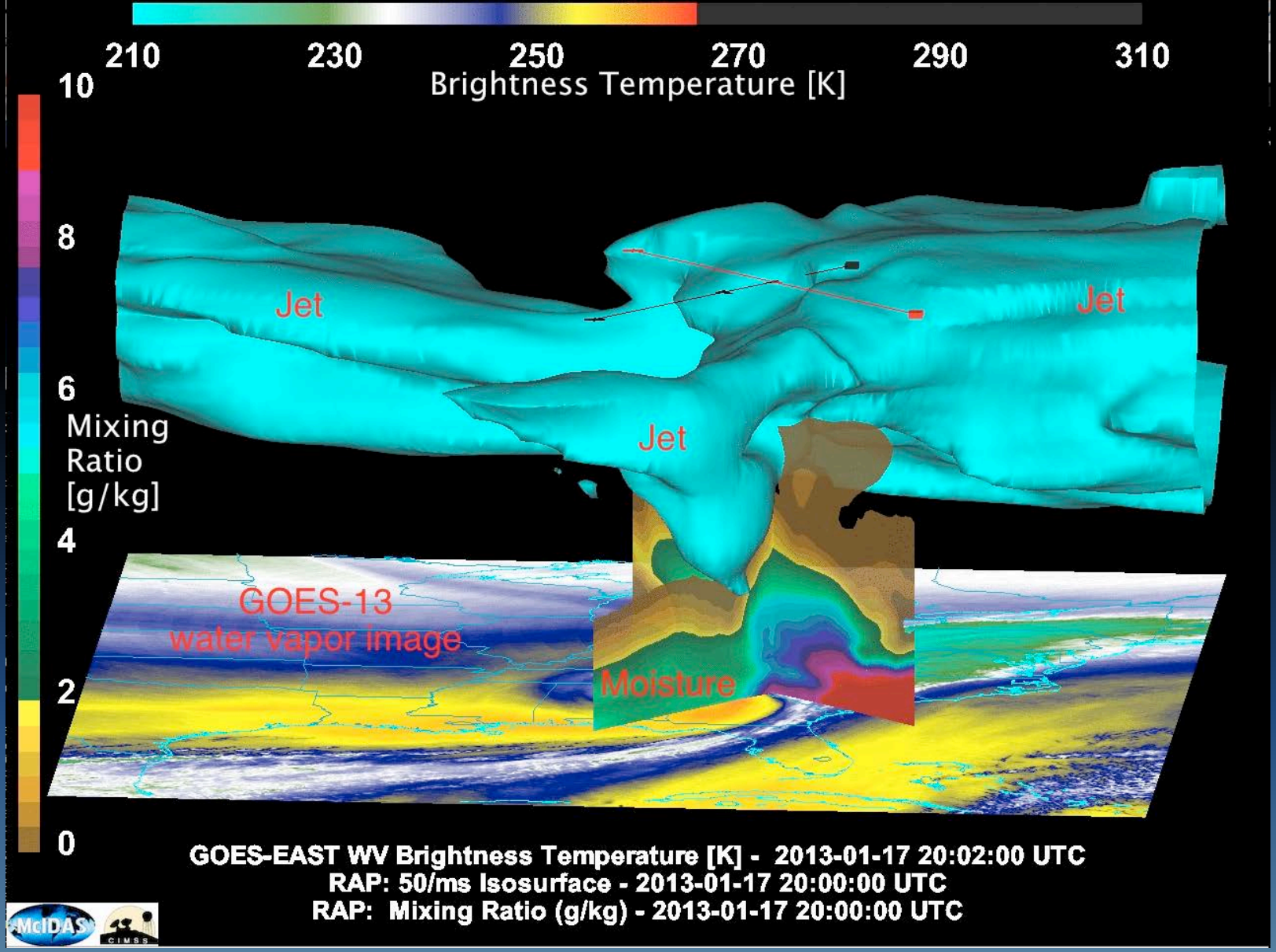
Variables Calculated

- Correlation between 3.9 and 10.7μm temperatures
- Band Difference (3.9-10.7 μm) Mean, Median, Variance, Kurtosis, Skewness, minimum, maximum, and more confined to range of -20 to 20 K

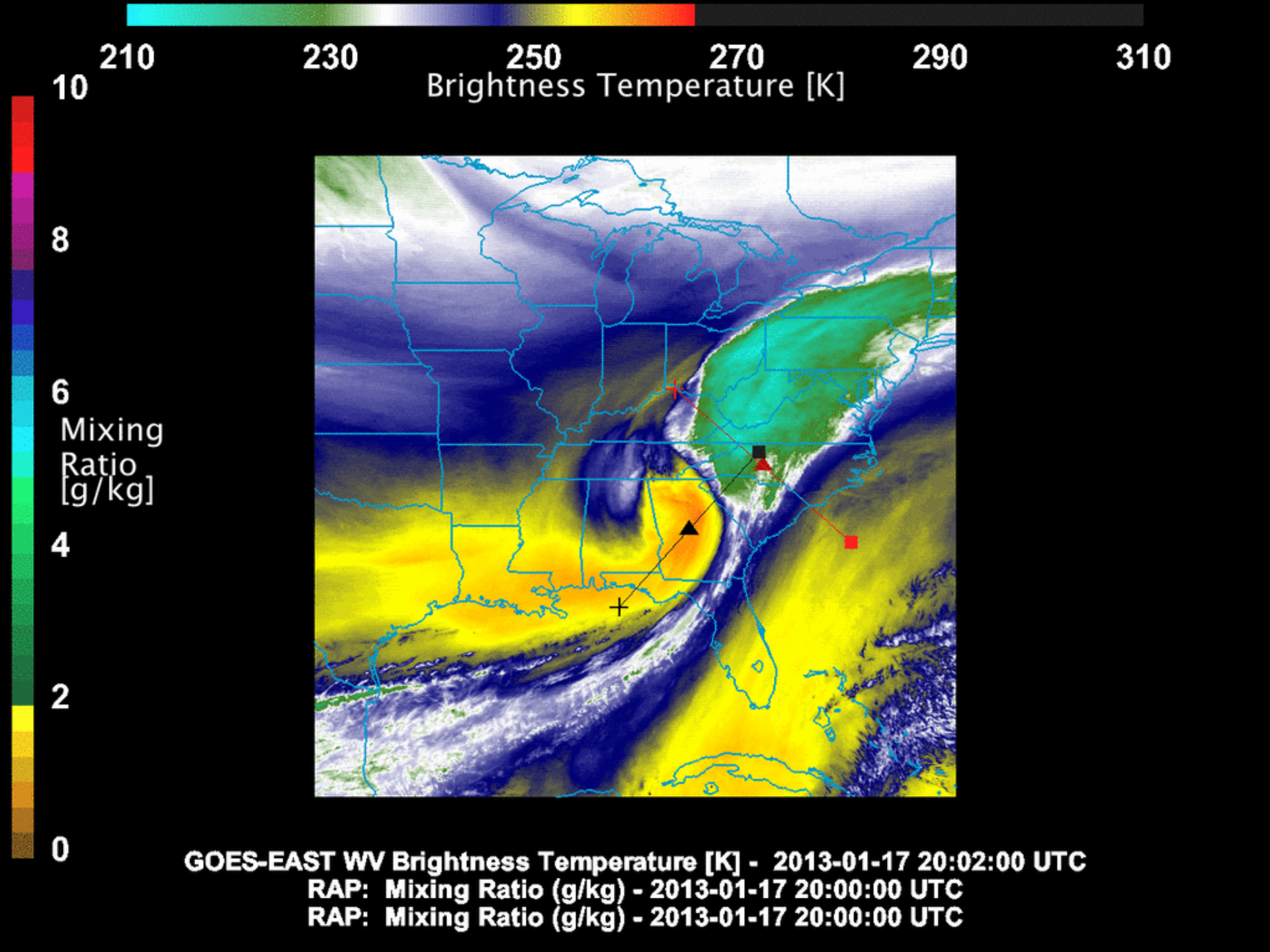


Possible Application

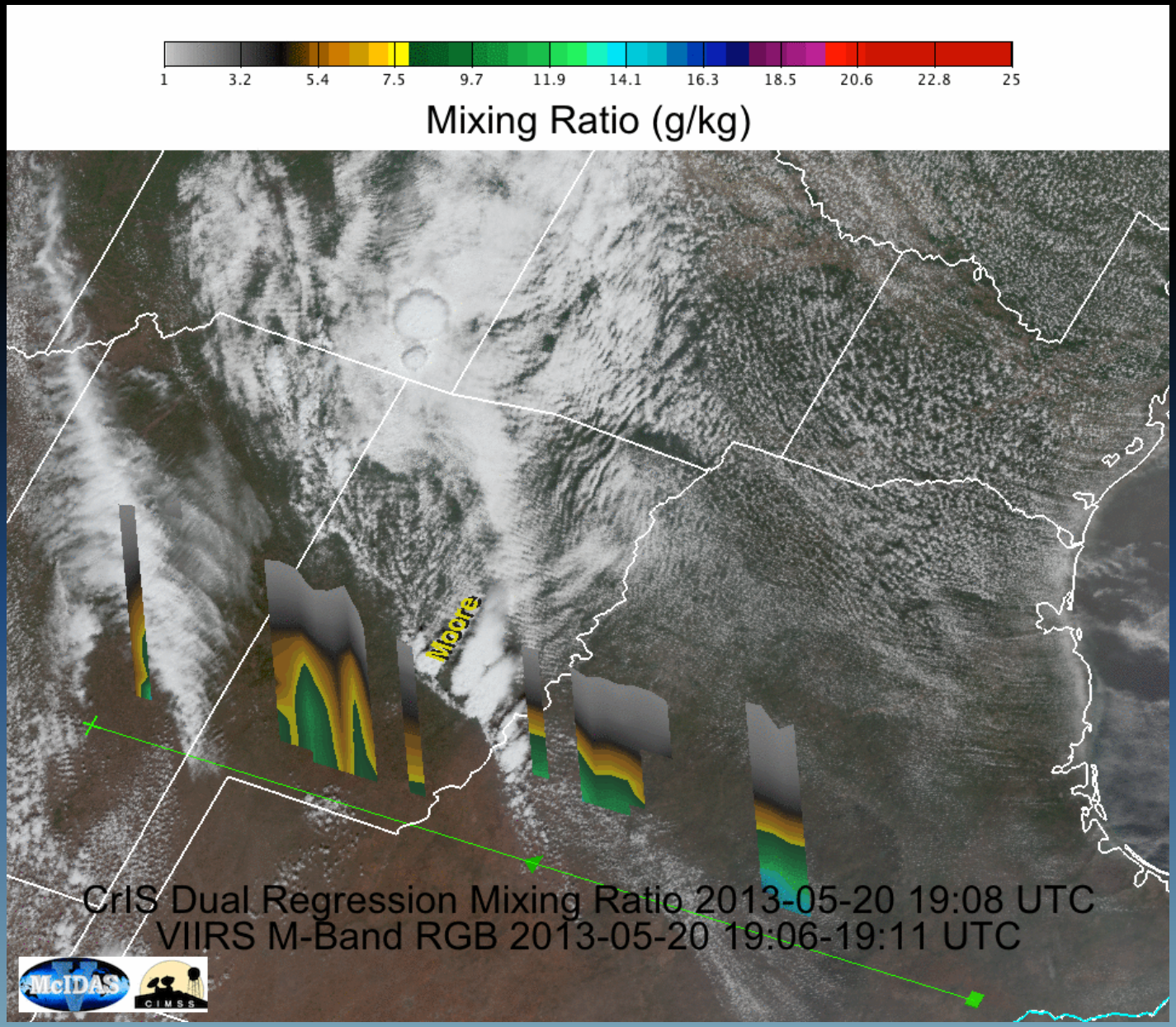




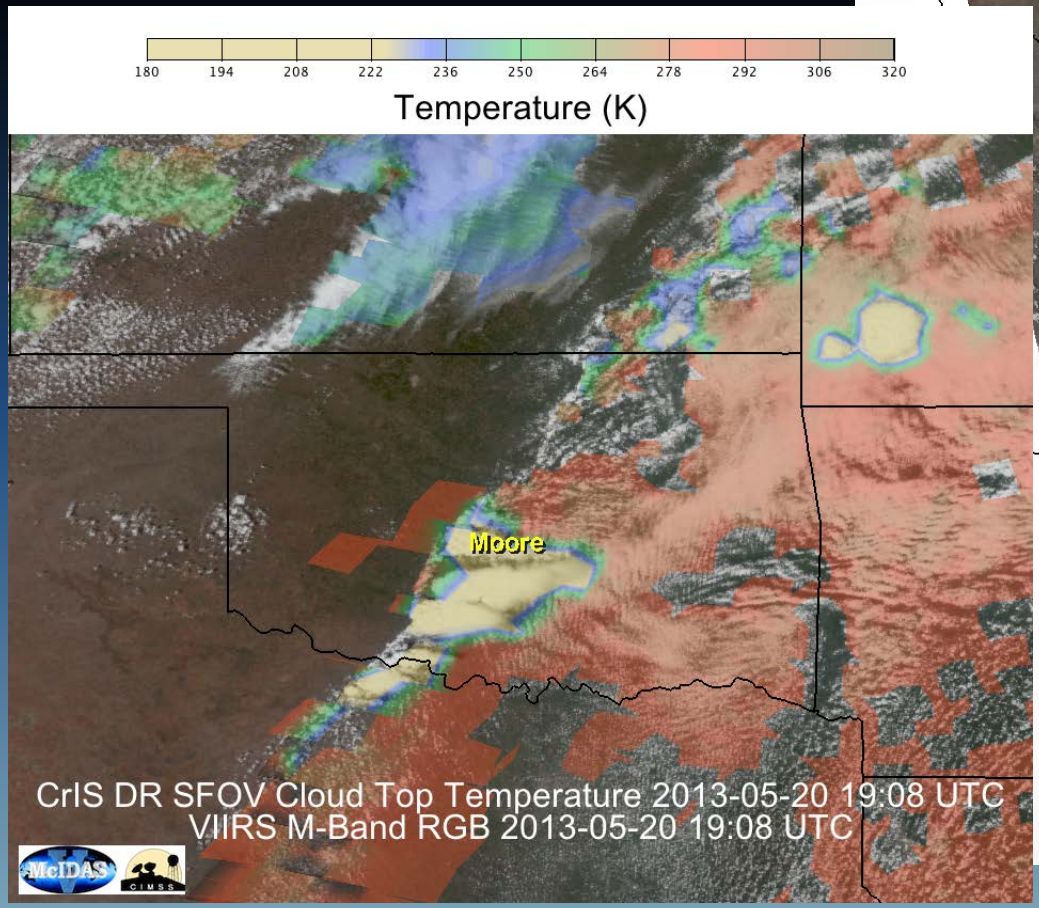
CIMSS Blog (Author: Scott Bachmeier)http://cimss.ssec.wisc.edu/goes/blog/archives/12

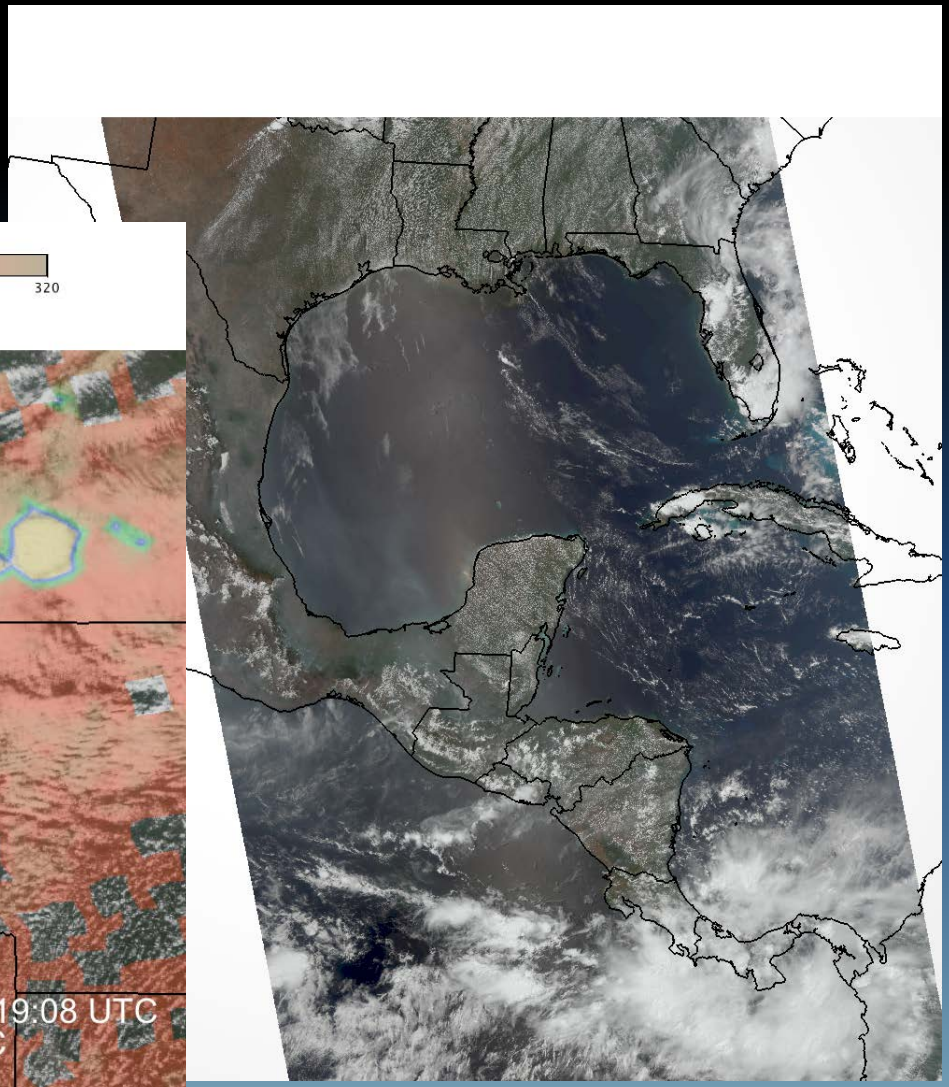

210
230
250
270
290
310
Brightness Temperature [K]
10
8
6
Mixing
Ratio
[g/kg]
4
2
0
GOES-EAST WV Brightness Temperature [K] - 2013-01-17 20:02:00 UTC
RAP: Mixing Ratio (g/kg) - 2013-01-17 20:00:00 UTC
RAP: Mixing Ratio (g/kg) - 2013-01-17 20:00:00 UTC

UW-CIMSS SFOV Specific Humidity



Suomi NPP CrIS RGB 18:59-1916 UTC and UW-SFOV Retrieval of Cloud Top Temperature





McIDAS-V Forums

http://dcdbs.ssec.wisc.edu/mcidasv/forums/

GOES Calibration

http://cimss.ssec.wisc.edu/goes/calibration/

Single Field of View Retrievals (International MODIS/AIRS Processing Package)

http://cimss.ssec.wisc.edu/imapp/