U.S. DEPARTMENT OF ENERGY | Office of Science

DOE/SC-ARM-TR-187

Multi-Angle Snowflake Camera Particle Analysis Value-Added Product

K Shkurko
A Talaei
T Garrett
K Gaustad

March 2018

ARM
CLIMATE RESEARCH FACILITY

DISCLAIMER

This report was prepared as an account of work sponsored by the U.S. Government. Neither the United States nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

DOE/SC-ARM-TR-187

Multi-Angle Snowflake Camera Particle Analysis Value-Added Product

Revision 1

K Shkurko, University of Utah
T Garrett, University of Utah
A Talaei, University of Utah
K Gaustad, Pacific Northwest National Laboratory

March 2018

Work supported by the U.S. Department of Energy,
Office of Science, Office of Biological and Environmental Research

Acronyms and Abbreviations

cm	centimeter
Hz	Hertz
IR	infrared
MASC	multi-angle snowflake camera
mm	millimeter
PC	personal computer
ROI	region of interest
USB	Universal Serial Bus
VAP	value-added product

Contents

Figures

Tables

1.0 Introduction

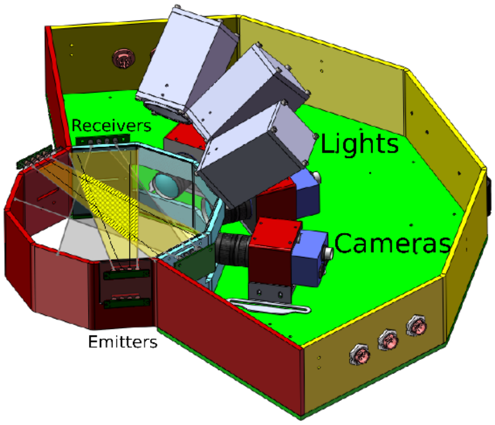

Figure 1. Schematic of the multi-angle snowflake camera (MASC). The hatched region represents the cross-section of the trigger area of the near-infrared motion detector system.

The Multi-Angle Snowflake Camera (MASC) addresses a need for high-resolution multi-angle imaging of hydrometeors in freefall with simultaneous measurement of fallspeed. As illustrated in **Figure 1**, the MASC consists of three cameras, separated by 36°, each pointing at an identical focal point approximately 10 cm away. Located immediately above each camera, a light aims directly at the center of depth of field for its corresponding camera. The focal point at which the cameras are aimed lies within a ring through which hydrometeors fall. The ring houses a system of near-infrared emitter-detector pairs, arranged in two arrays separated vertically by 32 mm.

When hydrometeors pass through the lower array, they simultaneously trigger all cameras and lights. Fallspeed is calculated from the time it takes to traverse the distance between the upper and lower triggering arrays. The trigger electronics filter out ambient light fluctuations associated with varying sunlight and shadows.

The microprocessor onboard the MASC controls the camera system and communicates with the personal computer (PC). The image data is sent via FireWire 800 line, and fallspeed (and camera control) is sent via a Universal Serial Bus (USB) line that relies on RS232-over-USB serial conversion.

See **Table 1** for specific details on the MASC located at the Oliktok Point Mobile Facility on the North Slope of Alaska.

The value-added product (VAP) detailed in this documentation analyzes the raw data (Section 2.0) using Python: images rely on OpenCV image processing library and derived aggregated statistics rely on some clever averaging. See Sections 4.1 and 4.2 for more details on what variables are computed.

Please see Garrett, 2012 for more details on the instrument and related data analysis.

Table 1. Specifics for the MASC located at Oliktok Point Mobile Facility. Note: This specification omits any upgrades or modifications that may have been applied to the camera system since manufacture.

Element	Description
Cameras	All three cameras are Unibrain Fire-i 980b grayscale cameras with maximum resolution of 2,448 x 2,048 pixels.
Lenses	All cameras use the same 12.5 mm Fujinon Megapixel C-mount lens with 75 mm horizontal field of view. This corresponds to 30.5 µm horizontal resolution per pixel.
Exposure	Exposure time is set to 40 µsec, or 1 / 25,000th of a second. Note: One can double-check this setting within the configuration XML file used for acquisition.
Lights	Each light is a 2,700 lumen light-emitting diode.
NIR detectors	Falling hydrometeors with maximum dimension > 0.1 mm can be detected by the two pairs of near-infrared detectors. Maximum detection frequency is set to 2 Hz.

2.0 Input Data

The MASC generates a collection of data for every detected particle. Each item listed below should be considered as a direct measurement. The VAP uses these values as input to derive variables it reports. See **Table 2** for a description of this data. The raw data consists of a few pieces, described in Sections 2.1-2.3.

Table 2. Types of data that MASC measures.

Element	Description
Timestamp	UTC timestamp when hydrometeor was captured.
Fallspeed	Measurement of the vertical fall speed of the hydrometeor. Calculated from the time taken for the hydrometeor to fall 32 mm between the top NIR trigger array and the bottom trigger array.
Images	Three images of the hydrometeor, one for each camera.
Configuration Options	Copy of the configuration file used for data acquisition.

2.1 Raw Data: Image Data Files

This ASCII file describes how to correlate image filenames to the camera ids and snowflake ids. The files are named `oliMASCM1.a0.YYYYMMDD.HH.raw_imgInfo.txt` See **Figure 2** for how the columns can be interpreted.

```
flake  camera  date          time                image name                               frame
ID     ID      (mm.dd.yyyy)  (hh:mm:ss.mmmmmm)                                            timestamp

1      0       10.25.2013    10:05:50.189202     2013.10.25_10.05.50_flake_1_cam_0.png   0

1      1       10.25.2013    10:05:50.219204     2013.10.25_10.05.50_flake_1_cam_1.png   0

1      2       10.25.2013    10:05:50.250206     2013.10.25_10.05.50_flake_1_cam_2.png   0

2      0       10.25.2013    10:05:50.802237     2013.10.25_10.05.50_flake_2_cam_0.png   0

2      1       10.25.2013    10:05:50.827239     2013.10.25_10.05.50_flake_2_cam_1.png   0
```

```
2     2      10.25.2013    10:05:50.856240    2013.10.25_10.05.50_flake_2_cam_2.png  0
```

Figure 2. Example image data info ASCII text file. Top row describes how to parse each column.

Please note that the perfect acquisition run will have each snowflake ID repeat three times, once for each camera. If one of these is missing, that means that particular image has been dropped. Also, all of the images listed under the image name must be present on disk. Finally, the timestamps of file acquisition may be slightly different—please be cautious of this when parsing timestamps, since images may span across the seconds (or minutes, or hours) boundary.

2.2 Raw Data: Fallspeed Data Files

This ASCII file describes how to correlate snowflake ids with the measured fallspeed. The files are named `oliMASCM1.a0.YYYYMMDD.HH.raw_dataInfo.txt` See **Figure 3** for how the columns can be interpreted.

```
snowflake id              date (mm.dd.yyyy)         time (hh:mm:ss.mmmmmm)   fall speed (m/s)

1                         10.25.2013                10:05:50.336211          0.314453

2                         10.25.2013                10:06:12.744492          0.673613

3                         10.25.2013                10:06:13.318525          1.02842

4                         10.25.2013                10:06:13.718548          0.863711
```

Figure 3. Example fallspeed data info ASCII text file. Top row describes how to parse each column.

2.3 Raw Data: Configuration XML File

This XML file is a copy of the configuration settings used for data acquisition, which are regenerated every time acquisition is restarted. These are necessary for the processing algorithm to convert number of pixels into physical measurements of distances. The files are named `oliMASCM1.a0.YYYYMMDD.HHMM.raw_config.xml` See MASC instrument documentation for the description of all elements for an example file shown in **Figure 15** in the Appendix. **Figure 4** describes elements that must be parsed for each camera.

XML Element	Description
`<fieldOfViewInmm val="0.0306372549"/>`	Specifies the horizontal field of view (in mm) of a single pixel in the image when looking at the focal plane. It has a single attribute that corresponds to the numeric value. This parameter is necessary for downstream data analysis and must be updated every time camera configuration changes. Computed by measuring the total field of view for the entire image and then dividing by the image width: 0.03063… = 75 / 2448

`<top val="0"/>`
`<left val="0"/>`
`<bottom val="0"/>`
`<right val="0"/>`
	Within `<startUpInfo> / <format7Info>` attribute. Identifies the amount of cropping from the top, left, bottom, or right of the image in val attribute.

Figure 4. Elements within configuration XML file relevant for parsing. These reside within `<acquisitionConfiguration> / <camDeviceConfiguration> / <camerasInfo> / <camera>` attributes. There is a `<camera>` for each camera in the MASC.

2.4 Analysis Configuration File

The configuration file is in the JSON format and stores all the parameters that the analysis script uses. Some of the parameters may need to be updated every time the instrument is moved or changed. Although it also contains values used to filter out particles, they should be considered as defaults because they are overwritten by values stored in `valid_min`, `valid_max`, `warn_min` and `warn_max` attributes for the appropriate diagnostic variables within each datastream. For more information on how the VAP filters data, see Section 4.2.

The file is named `defImgAnlParams.json` and resides in the VAP's configuration directory. An example file is shown in **Figure 16** in the Appendix. Comments next to each variable indicate usage. The final values used by the VAP are written into `anal_config_json` global attribute within `mascparticlesM1.c1` datastream.

3.0 Algorithm and Methodology

This section describes the basic workflow for the VAP. For a detailed description of pseudocode, see Appendix C. Every variable that determines quality is described in Section 4.2 including how the determination is made.

The VAP operates on one day of data. First, the VAP processes particle data: analyzing three images and then aggregating the result per particle. This is stored in mascparticlesM1.c1 datastream. Then, all particles are binned into 5-min. intervals and their analysis data aggregated per bin. This is stored in mascparticlesavgM1.c1 datastream. Data is tested and filtered at various stages of processing.

Essentially, the VAP processing has two modes: image processing using OpenCV and averages of results at various granularity.

3.1 Per-Particle Analysis

At the very first stage, the VAP checks whether input is appropriate. If the fallspeed is determined to be too fast (valid_max), the appropriate quality bit is set for all data for this particle. If either camera index is set to MISSING_VALUE or the image cannot be found, the analysis data for this image is set to MISSING_VALUE (and appropriate quality bits are set). If for the particle, all camera indexes or images are missing or no images were used to compute per-particle average, then the particle averages are set to MISSING_VALUE (and appropriate quality bits are set).

The VAP relies on OpenCV for image processing. See the following section for an algorithmic flowchart. First, it crops the image to remove infrared (IR) emitters and reduce computation workload. We determine which pixels are in the foreground by simple thresholding of their brightness (value). Edges of each particle within the image are detected using Otsu's binarization (See: `http://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0`). We rely on

OpenCV's contour algorithm to extract each particle captured within the image. According to documentation, it relies on the algorithm detailed in Suzuki, 1985.

Analysis for each contour (particle within image) relies on the following OpenCV functions:

- The bounding box is determined using the boundingRect function on pixels deemed as foreground. The documentation does not provide an algorithm, but we assume it is a simple min/max of locations of pixels that are foreground.
- Particle area is computed by simply counting how many pixels within the contour are foreground. This uses OpenCV's function countNonZero.
- Perimeter of the particle relies on OpenCV's function arcLength. The documentation does not provide a link to the algorithm it uses.
- Best fit ellipse is fit using OpenCV's fitEllipse function. It relies on the first algorithm described by Fitzgibbon, 1995.
- The rest of the measures are simple arithmetic averages that use OpenCV's mean function.

We filter each contour (particle within image) using six parameters: size (area), maximum pixel intensity, pixel intensity variability, length of particle touching the image edge, measure of focus, and location of the bottom of the particle. See Section 4.2 for more details.

If the contour passes filters, the VAP will stores the analysis results for the contour most in focus. If the contour does not pass filters, the VAP still stores the data, but sets appropriate quality bits. The VAP also stores how many contours (particles) were found within the image.

Besides storing analysis results for each image, each particle also stores the averages of per-image results. These are simple averages, which consider only images that pass the filters. Additionally, each particle stores an approximation to its flatness, which is derived from at least two valid images. If there are 0 or 1 valid images, then flatness is set to MISSING_VALUE, and appropriate quality bits are set.

3.2 Generating Time Bins

Once again, this is a different method of averaging. Once all particles were analyzed, the VAP bins them into 5-min. intervals based on capture time. Then it averages per-particle data (averages derived from images), but only for valid particles.

If there are not enough total or valid particles within the bin, all data but particle count is set to MISSING_VALUE. To be considered valid, a particle must have an appropriate fallspeed, and all of its averages must be good (no quality bits set).

If there are enough particles for the average to be considered statistically significant, then the time bin stores data.

3.3 Algorithmic Flowchart

Once again, see Appendix C for more detailed pseudocode of all steps the VAP implements.

4.0 Output Data

The VAP produces two datastream files per day. The datastream that stores data per particle is saved in files named olimascparticlesM1.c1.YYYYMMDD.hhmmss and time-binned data is saved to olimascparticlesavgM1.c1.YYYYMMDD.hhmmss. The YYYYMMDD.hhmmss is the timestamp of the first measurement in the file

First we will describe what data the VAP computes, and then describe all of the datastreams (to aid the long field descriptions within PCM).

4.1 Scientifically Relevant Variables

See **Table 3** for a detailed description of all variables deemed scientifically relevant.

Table 3. Description of scientifically relevant variables.

Symbol	Variable	Units	Description / Compute Method
	Hydrometeor timestamp	UTC timestamp	**Description:** Timestamp when either hydrometeor fallspeed was captured (particle data set) or the center for time bin that averaged particle data (time series data set) **Compute Method:** Captured
	Hydrometeor image path		**Description:** File path to each captured image file for the hydrometeor **Compute Method:** Captured
v	Fallspeed	m/s	**Description:** Measurement of the vertical fall speed for the captured hydrometeor **Compute Method:** Captured
D	Maximum dimension	mm	**Description:** The longest axis of the hydrometeor as derived from the image **Compute Method:** First, identify a region of interest in the image and fit an ellipse to it. Then, maximum dimension is the longest axis of said ellipse
A	Geometric cross-section	mm^2	**Description:** Cross-sectional area of the hydrometeor, excluding interior holes **Compute Method:** First, identify a region of interest in the image and label pixels not part of the background (relies on image brightness threshold). Then, geometric cross-section is the total number of pixels remaining within that is converted into mm^2
r	Area-equivalent radius	mm	**Description:** The area-equivalent radius of a circle with the same area as computed geometric cross-section *A* **Compute Method:** First, compute geometric cross-section *A*, then $r = \sqrt{A/\pi}$
P	Perimeter	mm	**Description**: Length of hydrometeor's outer contour **Compute Method:** Based on the region of interest, measures which pixels are on the outside edge
θ	Orientation	degrees	**Description:** Angle of the longest axis away from the horizontal. Measured as absolute value in counter-clockwise direction with 0 on the right.

Symbol	Variable	Units	Description / Compute Method
			Compute Method: Fits an ellipse to the region of interest, and then measures the angle towards the longest axis
α	Aspect ratio	unitless	**Description:** Ratio of shortest to longest axis of the hydrometeor. Represents how elongated a particle is. A cylindrical aggregate falling at an angle will tend to have a low value of α at all camera views. A crystalline snowflake will have a high value of α in one view and a low value in another **Compute Method:** Fit an ellipse to the region of interest, and then measure lengths of the longest *LA* and shortest *SA* axes. Then $\alpha = SA / LA$
i	Intensity	unitless	**Description:** Fractional intensity (in [0, 1]) of the region of interest, averaged over the image excluding the background **Compute Method:** For every pixel within the region of interest, excluding background, first convert its intensity into range [0, 1]. Then *i* is the mean of all these intensities
<*σ*>	Intensity variability	unitless	**Description:** Fractional intensity (in [0, 1]) variability of the region of interest, averaged over the image excluding background **Compute Method:** For every pixel within the region of interest excluding the background, compute intensity variability between that pixel and all immediately adjacent pixels (via function similar to MATLAB's `rangefilt`). Then convert into range [0, 1]. The variability <*σ*> is the mean of all these per-pixel variabilities
χ	Complexity / habit	unitless	**Description:** Estimate of hydrometeor complexity and riming extent based on perimeter and internal complexity. The complexity metric offers an objective measure for hydrometeor type. Riming tends to “round” and “smooth” hydrometeors leading to relatively low values of χ. Prior results from Utah suggest that values of $\chi < 1.35$ consistently correspond with lump and conical graupel. More aggregated forms have $\chi > 1.75$. Heavily rimed crystals and aggregates tend to lie in between. **Compute Method:** $\chi = \frac{P}{2\pi r}\left(1 + \langle\sigma\rangle\right)$ A value of $\chi = 1$ translates to a perfect, homogenous circle.
ϕ	Flatness	unitless	**Description:** Estimate of how flat a hydrometeor is based on multiple angle views, unlike aspect ratio α, which represents how elongated a particle is. **Compute Method:** $\phi = \frac{\alpha_{max} - \alpha_{min}}{\alpha}$ where α_{min} and α_{max} are the minimum and maximum values of α from all images respectively. A value of $\phi = 0$ corresponds to a sphere.
	Rain	unitless	**Description:** Estimate if a hydrometeor is a rain drop. **Compute Method:** The rain drop is estimated based on hydrometeor intensity *i*, intensity variability <*σ*> or the number of particles. If a hydrometeor meets any of the followings conditions, it is estimated as a rain drop:

Symbol	Variable	Units	Description / Compute Method
			1. Mean pixel intensity *i* of hydrometeor is > 0.3. 2. Mean pixel intensity variability <*σ*> of hydrometeor is > 0.3. 3. Number of the particles is equal or higher than 3. Rain value becomes 1 if hydrometeor meets any of the above conditions. It becomes 0 if it does not meet any of the above conditions. It becomes NA when not enough information is available to identify the hydrometer type.

4.2 Diagnostic Variables and Accepted Values

See **Table 4** for a detailed description of all variables deemed relevant for diagnostic purpose. ROI here means region of interest, which is computed as described in Section C.3. The table also describes how each variable is computed and what data values are considered of good quality.

Table 4. Description of diagnostic variables.

Symbol	Variable	Units	Description / Compute Method / Quality Check
v	Fallspeed	m/s	**Description:** Fall speed of captured hydrometeor, which is also useful to identify bad data **Compute Method:** Captured **Quality Check:** $0 < v < 10$ checks for bad data implemented through `valid_min` and `valid_max`, $v < 5$ used before particle is averaged into a time bin implemented through `warn_max`
A	Geometric cross-section	mm^2	**Description:** Cross-sectional area of the hydrometeor, excluding interior holes **Compute Method:** First, find the region of interest within an image. Then count how many non-background pixels are within the ROI, converted into mm **Quality Check:** 0.04 via `warn_min`
	Particle edge touch	mm	**Description:** For each image, stores the estimate for how much the particle overlaps with the image edge. If the VAP specifies additional cropping, the image edge is moved appropriately. First counts the number of non-background pixels in the region of interest crossing the edge, and then converts it into physical units **Compute Method:** First, find the region of interest within an image. Then count how many pixels over all with image border, converted into mm **Quality Check:** 0.5 via `warn_max`
i	Intensity	unitless	**Description:** The value *i* from **Table 3** **Quality Check:** 0.2 via `warn_min`
<*σ*>	Intensity variability	unitless	**Description:** The value <*σ*> *from* **Table 3** **Quality Check:** 0.019 via `warn_min`
Δy	ROI bottom, vertical position	mm	**Description:** Distance of the bottom of the hydrometeor from the top of the image frame (excluding any cropping). Hydrometeors typically trigger the lower near infrared sensor array within a narrow range of distances from the top of the image.

Symbol	Variable	Units	Description / Compute Method / Quality Check
			Δy is a very good first order indicator whether a hydrometeor image is obtained in a manner consistent with the instrument design The VAP filters particle images using this measure, so that no fallspeeds (and thus particles) are accepted if Δy is outside of a given acceptable range. The range of Δy values are best obtained by computing a histogram of all Δy to see where most values lie **Compute Method:** First, find the region of interest within an image. Then fit axis-aligned box tightly around this ROI. Then Δy is distance from the top of the image to the bottom bound, converted from pixels into mm **Quality Check:** 32 `warn_min`, 36 `warn_max`
f	ROI focus	unitless	**Description**: Estimate of the focus of the region of interest; a bit like an entropy measure **Compute Method:** Using values shown in **Table 3**, $f = i \langle \sigma \rangle$ **Quality Check:** A value $f > 0.01$ is a reasonable minimum acceptable value for scientifically valuable images
N	Num objects in frame		**Description:** Number of regions of interest in the frame, all ignoring background pixels. Values of *N* greater than a threshold (perhaps 1) might be associated with blowing snow or coincident hydrometeors that confuse the fall speed measurement **Compute Method:** Simple count
	Num images for average (per particle)		**Description:** Number of camera images used to derive per-particle analysis averages **Compute Method:** Simple count **Quality Check:** `valid_min` 1 and `warn_min` 2 for average, 2 for flatness
	Num images for average (per time bin)		**Description:** Number of camera images used to derive analysis averages over all particles that fall within a time bin **Compute Method:** Simple count **Quality Check:** `valid_min` 10, which depends on time bin width (5 min. in our case). For 1 hour, should be 60
	Rain	unitless	**Description:** a condition that estimates if a hydrometeor is a rain drop **Compute Method:** If a hydrometeor meets any of the conditions mentioned in **Table 3**, it is estimated as a rain drop. It gets 1, 0, or NA. 1 means the hydrometeor is most likely a rain drop, 0 means not a rain drop, and NA when image does not pass all the quality checks mentioned above. MASC data provides three images for the each particle id. Therefore, all three cameras should give the same value for the type of hydrometeor. The following logic is set for the rain from the rain value of each individual camera: I. Set rain value to 1 for all three images if a) rain is identified by at least two of the images, or b) rain is identified by one but two others have 0 values for the rain, c) rain is identified by one but two others are 0 and NA. II. Set rain value to 0 for all three images if a) all three images get 0 for the rain value, or b) two get 0 and the third one gets NA.

Symbol	Variable	Units	Description / Compute Method / Quality Check
			III. Set rain value to NA for all three images if at least two of the images get NA value for the rain.

4.3 Output Datastream: masc.b1

This datastream is created as part of the raw data ingest and contains all of the raw data MASC produces: Section 2.0. See **Table 5** for descriptions of every variable and **Table 6** for associated quality variables. The dimension described in the table specifies data type that can be used to index into the array of data for each variable. When there are multiple dimensions, they are listed in order.

All images are stored as files with unique names: `olimascM1.a1.20160428.030610.png.id_00000036_cam_0.png`. These names can be decrypted in the following way: `<location>masc<data type>.<data level>.<date>.<time>.png.id_<flake index>_cam_<camera id>.png`. The `<date>` is formatted `yyyymmdd` and `<time>` is formatted `hhmmss`. These names are not stored explicitly, but are regenerated by the VAP during runtime based on stored per-particle data, like timestamp.

Some of the attributes included within the datastream by default are left out for brevity, because they are not essential to the VAP.

Table 5. Details on variables stored within the raw datastream, masc.b1.

Variable	Units	Description / Dimensions
`time`	UTC timestamp	Timestamp when the particle was captured by the MASC **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`snowflake_id`	-	Index of the captured particle. Index may be restarted at 0 when acquisition was restarted **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`snowflake_fall_speed`	m/s	Measured speed of the captured hydrometeor, *v*. **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`camera_id`	-	An array of camera ids, one for every image that was captured. If the image is found, then the corresponding array entry is set to the camera id (like 1, etc.). Otherwise, the entry is set to missing value **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`field_of_view`	mm	Horizontal field of view per pixel within the camera. This is on the camera resolution and lens. Can be computed information from the calibration: `(total horizontal field of view) / (horizontal image resolution)` Value is necessary for the VAP to convert pixel counts into units, like mm. It is extracted from saved copy configuration files **Dimension:** *camera* = 3, one for each camera in MASC
`crop_from_top`	pixels	Number of pixels that were cropped from the top of the image (per camera) before being saved Value is necessary for the VAP to convert pixel counts into physical

Variable	Units	Description / Dimensions
		units, like mm. It is extracted from the saved copy of configuration files. **Dimension:** *camera* = 3, one for each camera in MASC
`crop_from_bottom`	pixels	Number of pixels that were cropped from the bottom of the image (per camera) before being saved Value is necessary for the VAP to convert pixel counts into physical units, like mm. It is extracted from the saved copy of configuration files **Dimension:** *camera* = 3, one for each camera in MASC
`crop_from_left`	pixels	Number of pixels that were cropped from the left of the image (per camera) before being saved Value is necessary for the VAP to convert pixel counts into physical units, like mm. It is extracted from the saved copy of configuration files **Dimension:** *camera* = 3, one for each camera in MASC
`crop_from_right`	pixels	Number of pixels that were cropped from the right of the image (per camera) before being saved Value is necessary for the VAP to convert pixel counts into physical units, like mm. It is extracted from the saved copy of configuration files **Dimension:** *camera* = 3, one for each camera in MASC

Table 6. Quality variables for raw datastream, masc.b1. See Section 4.2 for more details on diagnostic variables.

Variable	Bits / Description
`qc_snowflake_fall_speed`	Checks `snowflake_fall_speed` value bit 1 (bad): `value = MISSING_VALUE` bit 2 (bad): `value < valid_min` bit 3 (bad): `value > valid_max`

4.4 Derived Datastream: mascparticles.c1

This datastream is created by the VAP as the result of analysis. It holds analysis data per particle and per each image associated with the particle. The datastream houses both types of variables described in Sections 4.1 and 4.2.

Table 7 describes all variables bits within the datastream that are stored for every image captured per particle. **Table 8** describes all other variables in the datastream that are per-particle aggregates of the image data. **Table 9** describes the quality of all variables within this datastream.

Some of the attributes included within the datastream by default are left out for brevity, because they are not essential to the VAP.

Table 7. Details on variables stored within the derived datastream, mascparticles.c1. These variables store data per image for every particle.

Variable	Units	Description / Dimensions
`time`	UTC timestamp	Copy of `time` variable from masc.b1 datastream in **Table 5**
`snowflake_id`	-	Copy of `snowflake_id` variable from masc.b1 datastream in **Table 5**
`snowflake_fall_speed`	m/s	Copy of `snowflake_fall_speed` variable from masc.b1 datastream in **Table 5**
`camera_id`	-	Copy of `camera_id` variable from masc.b1 datastream in **Table 5**
`maximum_dimension`	mm	Stores the value *D* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`particle_area`	mm^2	For each image, stores the estimate for particle area including background pixels. First counts the number of pixels in the region of interest and then converts it into physical units **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`particle_edge_touch`	mm	For each image, stores the estimate for how much the particle overlaps with the image edge. If the VAP specifies additional cropping, the image edge is moved appropriately. First counts the number of non-background pixels in the region of interest crossing the edge, and then converts it into physical units **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`area_eq_radius`	mm	Stores the value *r* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`perimeter`	mm	Stores the value *P* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`orientation`	degrees	Stores the value θ per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`aspect_ratio`	unitless	Stores the value α per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`complexity`	unitless	Stores the value χ per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`geometric_cross_ section`	mm^2	Stores the value *A* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`mean_pixel_intensity`	unitless	Stores the value *i* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`mean_pixel_intensity_ variability`	unitless	Stores the value $<\sigma>$ per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`roi_focus`	unitless	Stores the value *f* per image as computed by the VAP, see **Table 3**

Variable	Units	Description / Dimensions
		Dimension: *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC.
`num_objects`	-	Stores the value *N* per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`roi_position`	mm	Stores the location of the center of the region of interest per image as computed by the VAP **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC. *xy* = 2, one for each *x* and *y* dimensions
`roi_bot_position`	mm	Stores the value Δy per image as computed by the VAP, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC
`roi_half_width_height`	mm	Stores halves of width and height from the center of the region of interest per image as computed by the VAP. For example, left bound of ROI is: `(center x) + (half width)` **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC. *xy* = 2, one for each *x* and *y* dimensions
`rain`	unitless	For each image, stores the estimate for rain value, see **Table 3** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor). *camera* = 3, one for each camera in MASC

Table 8. Details on variables stored within the raw datastream, mascparticles.c1. These variables store data per particle that were computed based on per-image data in Table 7: Details on variables are stored within the derived datastream, mascparticles.c1 These variables store data per image for every particle.

Variable	Units	Description / Dimensions
`num_imgs_used_avg`	-	Stores number of images that were used to compute averages **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`maximum_dimension_avg`	mm	Stores the average of *D* over per-particle images. **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`particle_area_avg`	mm^2	Stores the average of `particle_area` over per-particle images, from **Table 7** **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`area_eq_radius_avg`	mm	Stores the average *r* over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`perimeter_avg`	mm	Stores the average *P* over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`orientation_avg`	degrees	Stores the average *θ* over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)

Variable	Units	Description / Dimensions
`aspect_ratio_avg`	unitless	Stores the average α over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`complexity_avg`	unitless	Stores the average χ over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`geometric_cross_ section_avg`	mm^2	Stores the average A over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`mean_pixel_ intensity_avg`	unitless	Stores the average i over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`mean_pixel_intensity_ variability_avg`	unitless	Stores the average $<\sigma>$ over per-particle images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`flatness`	unitless	Stores the value of computed from images stored within the particle. Requires at least two images **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)
`rain`	unitless	Stores the average of rain value over per-particle images, see **Table 4**. **Dimension:** *time*, single entry per timestamp (thus captured hydrometeor)

Table 9: Quality variables for derived datastream, mascparticles.c1. See Section 4.2 for more details on diagnostic variables.

Variable	Bits / Description
`qc_snowflake_fall_speed`	bit 1 (bad): `value = MISSING_VALUE` bit 2 (bad): `value < valid_min` bit 3 (bad): `value > valid_max` bit 4 (ind): `value > warn_max`. Filters particles before time binning
`qc_particle_edge_touch`	bit 1 (bad): `snowflake_fall_speed = MISSING_VALUE` bit 2 (bad): `camera_id` for this camera = `MISSING_VALUE` bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is `< warn_min` or `> warn_max` for that property bit 6 (ind): `value > warn_max`
`qc_geometric_cross_section`	bit 1 (bad): `snowflake_fall_speed = MISSING_VALUE` bit 2 (bad): `camera_id` for this camera = `MISSING_VALUE` bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is `< warn_min` or `> warn_max` for that property bit 6 (ind): `value < warn_min`

Variable	Bits / Description
qc_mean_pixel_intensity	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): camera_id for this camera = MISSING_VALUE bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < warn_min or > warn_max for that property bit 6 (ind): value < warn_min
qc_mean_pixel_intensity_variability	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): camera_id for this camera = MISSING_VALUE bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < warn_min or > warn_max for that property bit 6 (ind): value < warn_min
qc_roi_focus	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): camera_id for this camera = MISSING_VALUE bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < warn_min or > warn_max for that property bit 6 (ind): round(value * 100)/100 < warn_min
qc_roi_bot_position	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): camera_id for this camera = MISSING_VALUE bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < warn_min or > warn_max for that property bit 6 (ind): value < warn_min bit 7 (ind): value > warn_max
qc_num_imgs_used_avg	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): all camera_id for this particle = MISSING_VALUE bit 3 (bad): num_objects == MISSING_VALUE for all camera_ids due to a property of the particle most in focus is < warn_min or > warn_max for that property bit 4 (bad): value < valid_min bit 5 (ind): value < warn_min. Could indicate not enough data for an average
qc_num_objects	bit 1 (bad): snowflake_fall_speed = MISSING_VALUE bit 2 (bad): camera_id for this camera = MISSING_VALUE bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < warn_min or > warn_max for that property

Variable	Bits / Description
qc_maximum_dimension **qc_particle_area** **qc_area_eq_radius** **qc_perimeter** **qc_orientation** **qc_aspect_ratio** **qc_complexity** **qc_roi_position** **qc_roi_half_width_height**	bit 1 (bad): `snowflake_fall_speed = MISSING_VALUE` bit 2 (bad): `camera_id` for this camera = `MISSING_VALUE` bit 3 (bad): image for this camera missing bit 4 (bad): No particle detected in image bit 5 (ind): A property of the particle most in focus is < `warn_min` or > `warn_max` for that property
qc_maximum_dimension_avg **qc_particle_area_avg** **qc_area_eq_radius_avg** **qc_perimeter_avg** **qc_orientation_avg** **qc_complexity_avg** **qc_geometric_cross_section_avg** **qc_mean_pixel_intensity_avg** **qc_mean_pixel_intensity_variability_avg** **qc_flatnes**	bit 1 (bad): `snowflake_fall_speed = MISSING_VALUE` bit 2 (bad): all `camera_id` for this particle = `MISSING_VALUE` bit 3 (bad): `num_imgs_used = MISSING_VALUE` bit 4 (bad): `num_imgs_used < valid_min` bit 5 (ind): `num_imgs_used < warn_min`. Could indicate not enough data for an average

4.5 Derived Datastream: mascparticlesavg.c1

This datastream is created by the VAP as the result of analysis. It holds analysis data over all particles that fall within a time bin. Effectively these are averages over variables listed in Section 4.1.

Before being considered appropriate for averaging, all particles that fall into a specific time bin are filtered according to the quality parameters outlined in Section 4.2. If a particle fails a particular test, it is not considered appropriate to be averaged and is skipped.

Table 10 describes all variables within the datastream that are stored per time bin. **Table 11** describes the quality variables for all data stored within.

Some of the attributes included within the datastream by default are left out for brevity, because they are not essential to the VAP.

Table 10. Details on variables stored within the raw datastream mascparticlesavg.c1. These variables store data per time bin that were computed based on per-image data in Table 7.

Variable	Units	Description / Dimensions
time	UTC timestamp	Indicates the center of each time bin. The `bounds` attribute is set to -150, 150 sec. (5 min. bin width)
num_particles_total	unitless	Total number of particles that fell into this bin. Only a subset (counted in `num_particles_for_avg`) will be used to compute averages **Dimension:** *time*, single entry per timestamp (thus time bin)
num_particles_for_avg	unitless	Number of particles (subset of `num_particles_total`) that were used to compute the average stored in this time bin. If none, bin is empty and not created. The value of num_objects for at least 2 of 3 images must be the same and equal to 1. **Dimension:** *time*, single entry per timestamp (thus time bin)

Variable	Units	Description / Dimensions
fall_speed_avg	m/s	Average of fallspeed versus for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
maximum_dimension_avg	mm	Average of maximum dimension *D* for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
particle_area_avg	mm^2	Average of particle area including background for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
area_eq_radius_avg	mm	Average of area equivalent radius *r* for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
perimeter_avg	mm	Average of perimeter *P* for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
orientation_avg	degrees	Average of orientation θ for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
aspect_ratio_avg	unitless	Average of aspect ratio α for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
complexity_avg	unitless	Average of complexity χ for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
geometric_cross_ section_avg	mm^2	Average of geometric cross-section *A* for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
mean_pixel_ intensity_avg	unitless	Average of mean pixel intensity *i* for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
mean_pixel_intensity_ variability_avg	unitless	Average of mean pixel intensity variability <σ> for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)
flatness	unitless	Average of flatness ϕ for all valid particles within **Dimension:** *time*, single entry per timestamp (thus time bin)

Table 11. Quality variables for derived datastream, mascparticlesavg.c1. See Section 4.2 for more details on diagnostic variables.

Variable	Bits / Description
qc_num_particles_total	bit 1 (bad): `value = MISSING_VALUE` bit 2 (ind): `value = 0`
qc_num_particles_for_avg	bit 1 (bad): `value = MISSING_VALUE` bit 2 (ind): `value < warn_min`. Could indicate lack of samples for statistically significant results bit 3 (ind): `value = 0`
qc_fall_speed_avg **qc_maximum_dimension_avg** **qc_particle_area_avg** **qc_area_eq_radius_avg** **qc_perimeter_avg** **qc_orientation_avg** **qc_aspect_ratio_avg** **qc_complexity_avg** **qc_geometric_cross_section_avg**	Value here refers to appropriate variable bit 1 (bad): `value = MISSING_VALUE` bit 2 (bad): `num_particles_for_avg = MISSING_VALUE` bit 3 (ind): `num_particles_for_avg < warn_min`. Could indicate lack of samples for statistically significant results bit 4 (bad): `num_particles_for_avg = 0`

Variable	Bits / Description
qc_mean_pixel_intensity_avg **qc_mean_pixel_intensity_variability_avg** **qc_flatness_avg**	

5.0 Summary

The MASC captures both the fall speed and three camera views for individual hydrometeors. These measurements can be analyzed to provide information on hydrometeor properties. This document summarizes how these properties are calculated and their associated quality-control flags within the ARM MASC VAP.

6.0 Example Plots

All of the plots below were generated using ARM's dq_inspector tool applied to the VAP output. This section shows a selection of scientifically relevant variables (and their quality assessment) for particles (Section 6.1) and their averages over time (Section 6.2).

One can read these plots in the following manner. The plots show particular variables of interest as a time series (x-axis). The top portion plots the values as dots, which are colored by quality: G means good and I means indeterminate. The bottom portion shows which quality bits were set for each particular data point. If data do not pass the quality test (is set to MISSING_VALUE), then the data point is not shown on the top portion.

The colors in these plots can be deciphered as follows. Gray corresponds to lack of data. Green corresponds to data that is considered good, as in the bit associated with badness was not set. Yellow corresponds to data quality as indeterminate.

For example, if a fall speed is deemed too fast (`value > valid_max`), then the data is considered bad. If a number of particles used for average is low (`value < warn_min`), then the data is considered of indeterminate quality because it may not be statistically significant. It is up to the user to decide whether it should be ignored. One can check the actual value that quality bit checks.

6.1 Per-Particle Data, mascparticlesM1.c1 Datastream

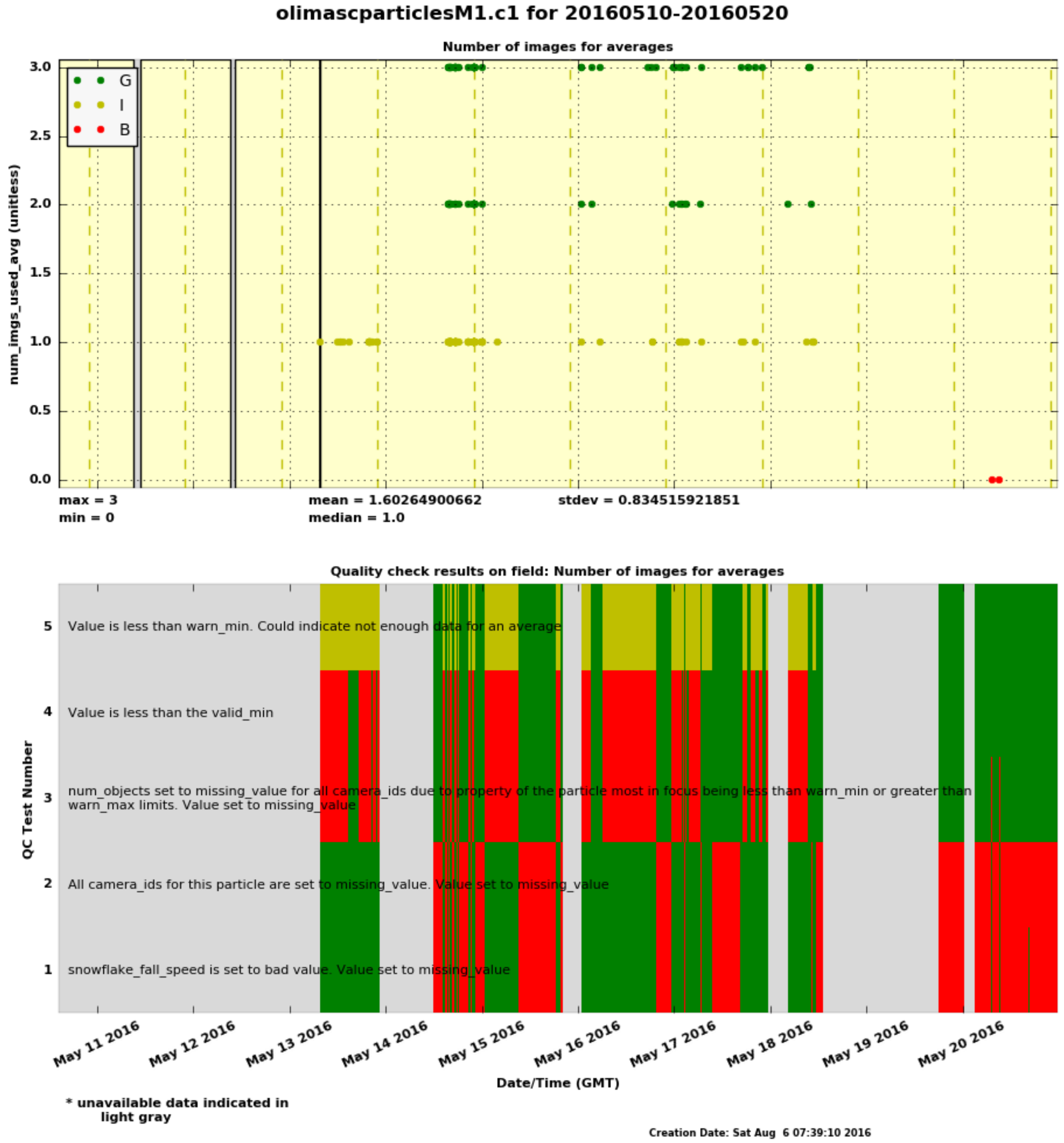

Figure 5. Retrieved number of images (of 3) used to average computed features for every captured particle (top) and quality bits (bottom) for data captured between May 10 and May 20, 2016. See Section 6.0 for description of colors.

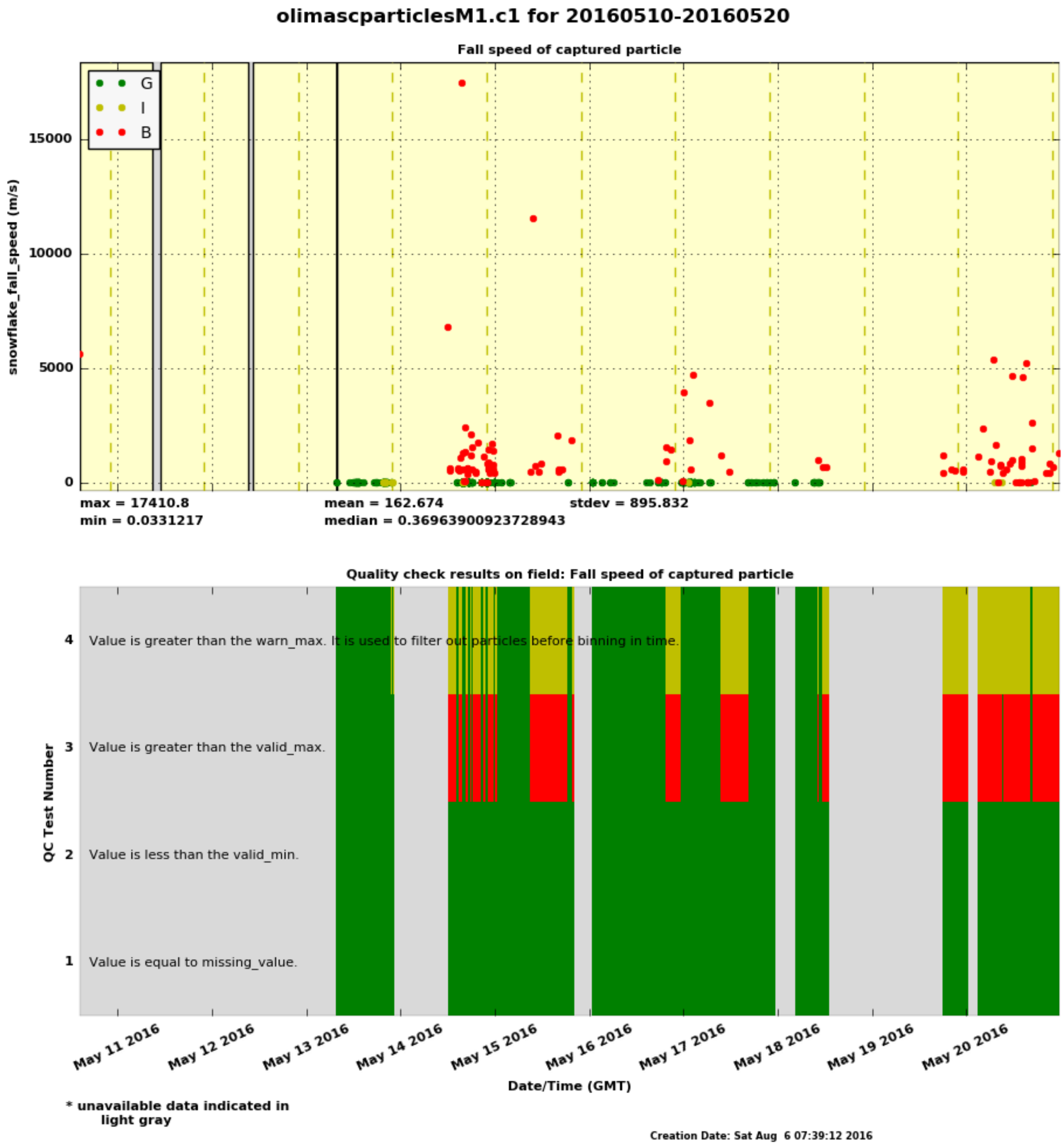

Figure 6. Retrieved fall speed for every captured particle (top) and quality bits (bottom) for data captured between May 10 and May 20, 2016. See Section 6.0 for description of colors.

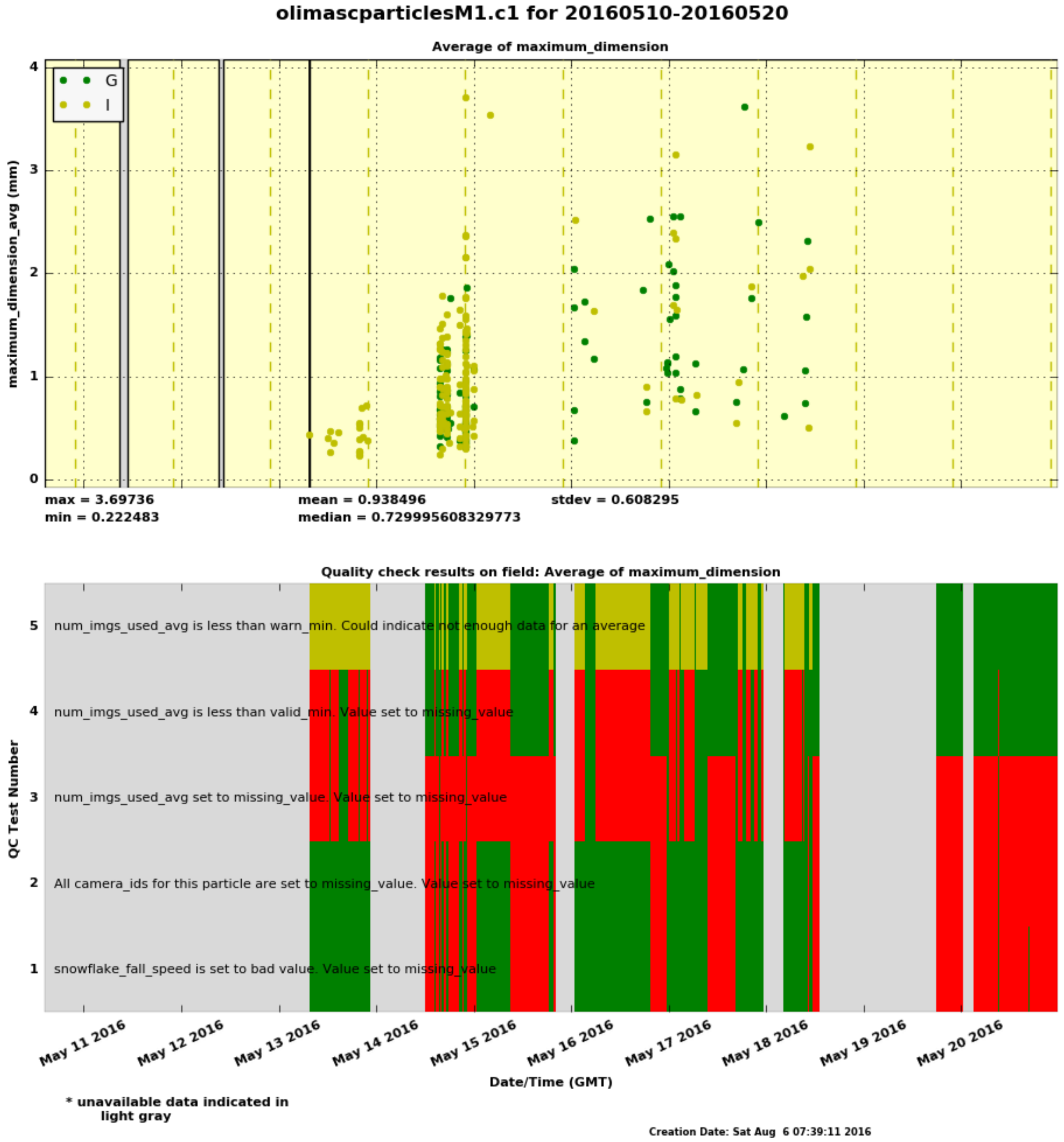

Figure 7. Retrieved maximum dimension averaged over three images for every captured particle (top) and quality bits (bottom) for data captured between May 10 and May 20, 2016. See Section 6.0 for description of colors.

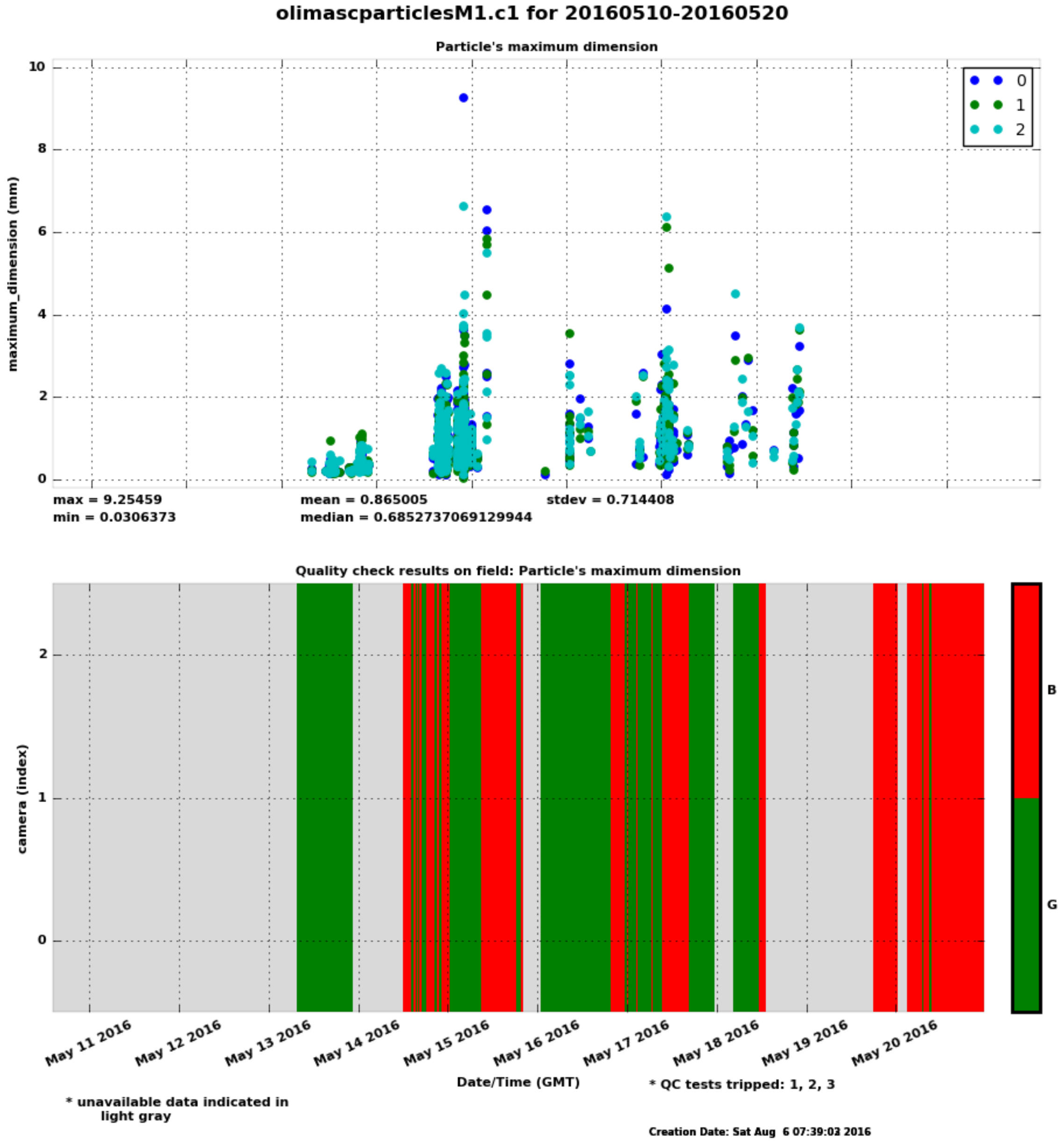

Figure 8. Retrieved maximum dimension for each of the three images for every captured particle (top). Numbers denote camera id. All average quantities are derived from per-image data like this. Quality bits (bottom) for data captured between May 10 and May 20, 2016. See Section 6.0 for description of colors.

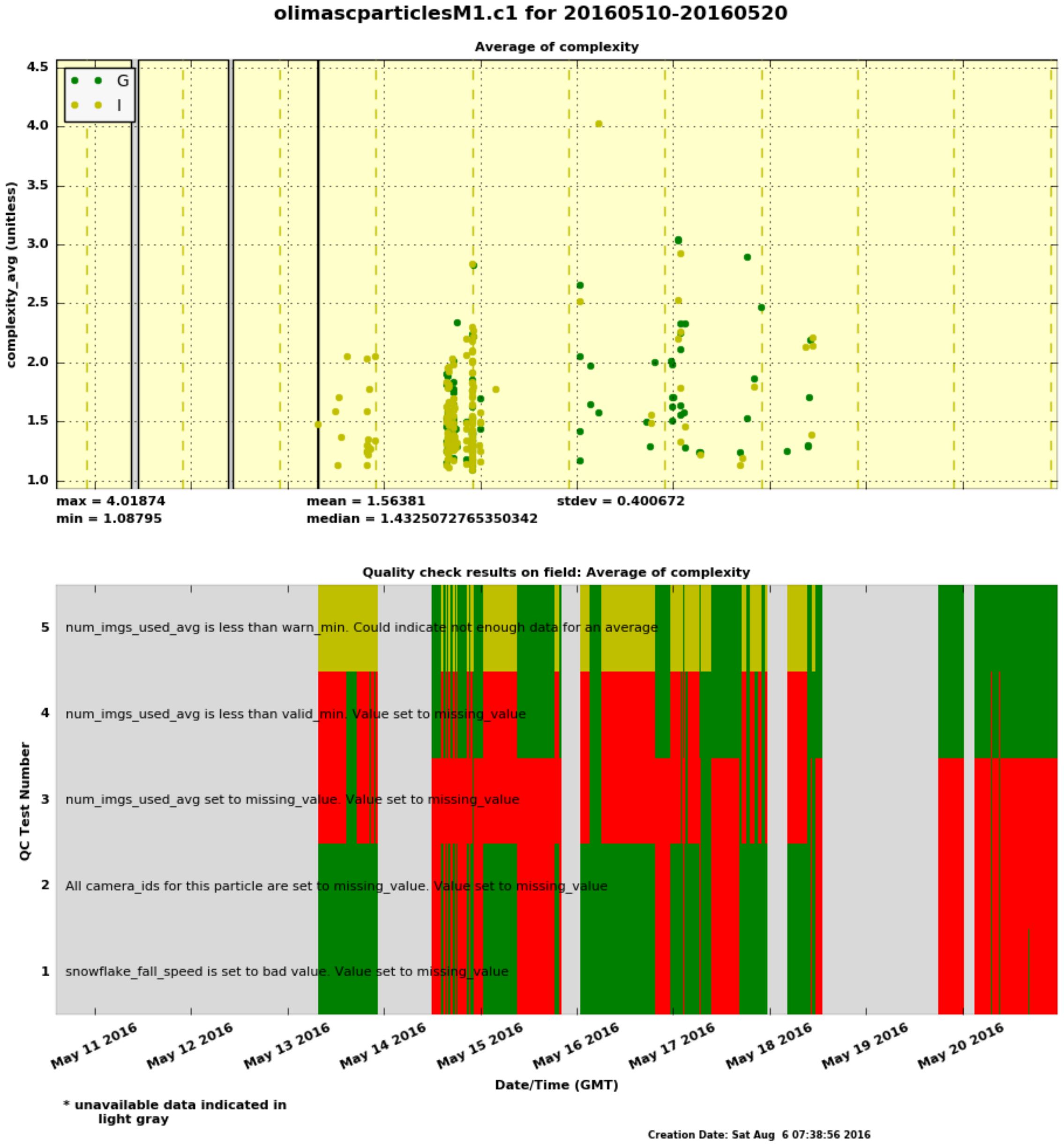

Figure 9. Retrieved complexity averaged over three images for every captured particle (top) and quality bits (bottom) for data captured between May 10 and May 20, 2016. See Section 6.0 for description of colors.

6.2 Bins in Time, mascparticlesavgM1.c1 Datastream

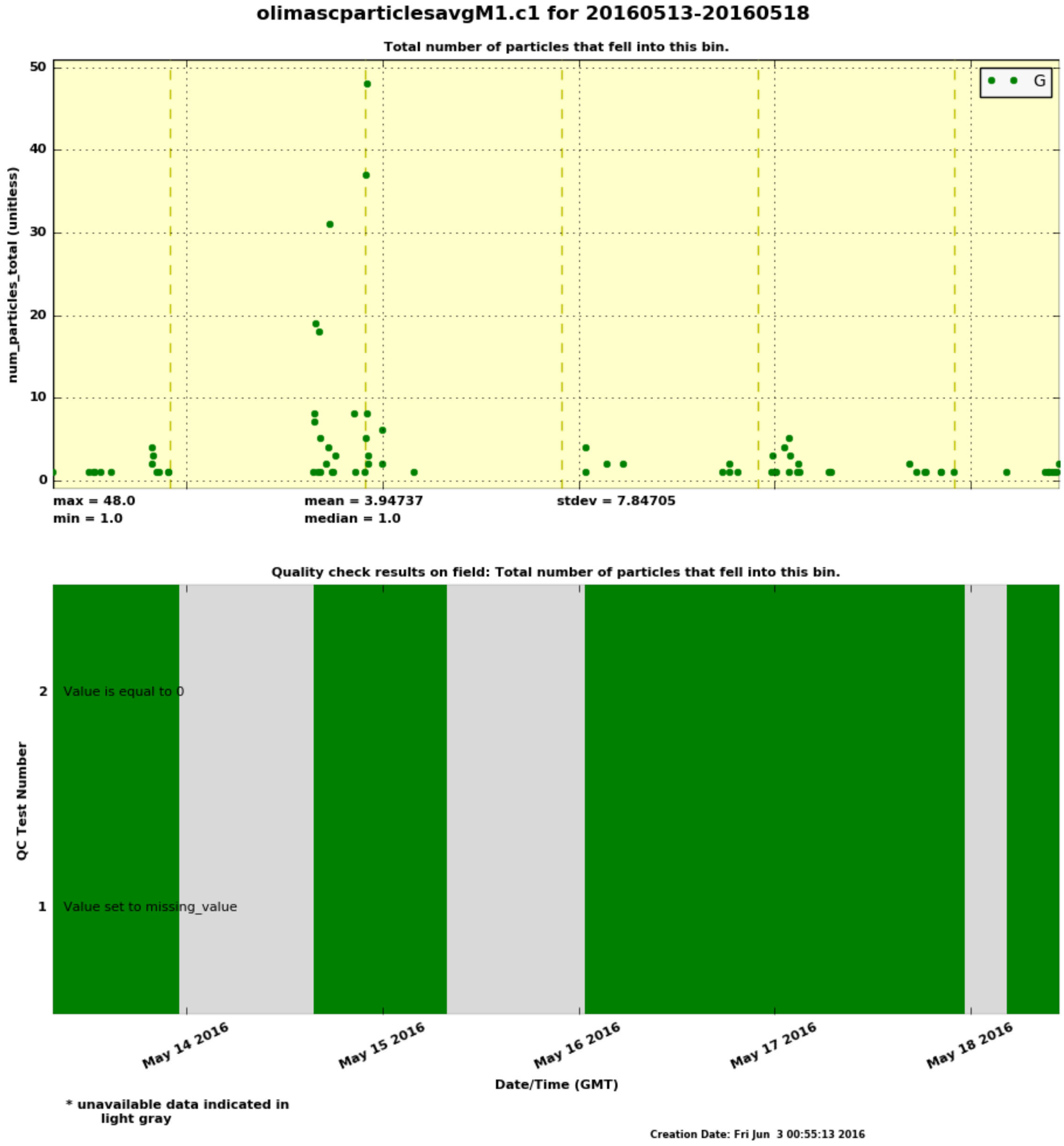

Figure 10. Retrieved total number of particles that fell into each 5-min.-wide time bin (top) and quality bits (bottom) for data captured between May 13 and May 19, 2016. See Section 6.0 for description of colors.

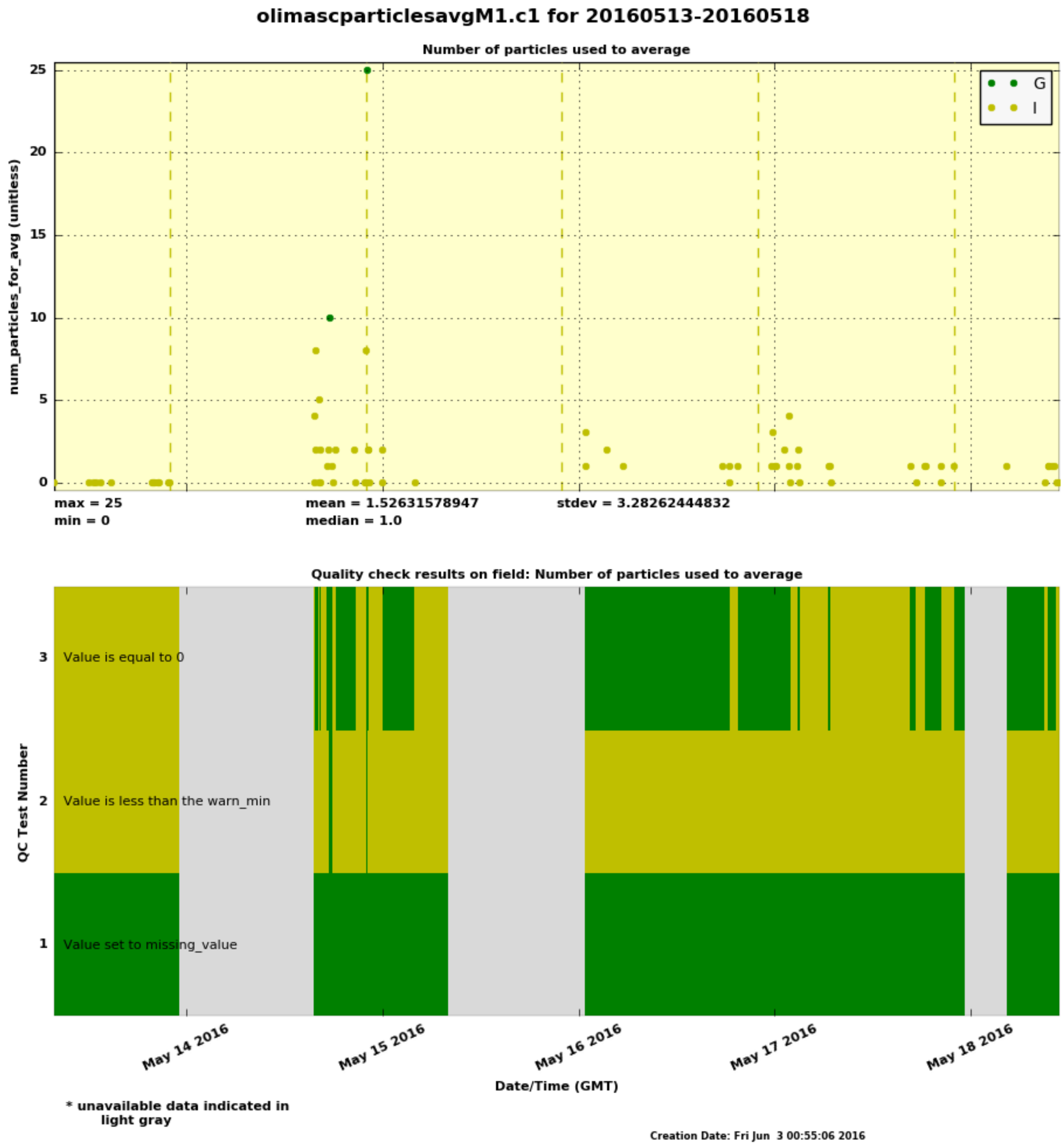

Figure 11. Retrieved number of particles used to average data for each 5-min.-wide time bin (top) and quality bits (bottom) for data captured between May 13 and May 19, 2016. See Section 6.0 for description of colors.

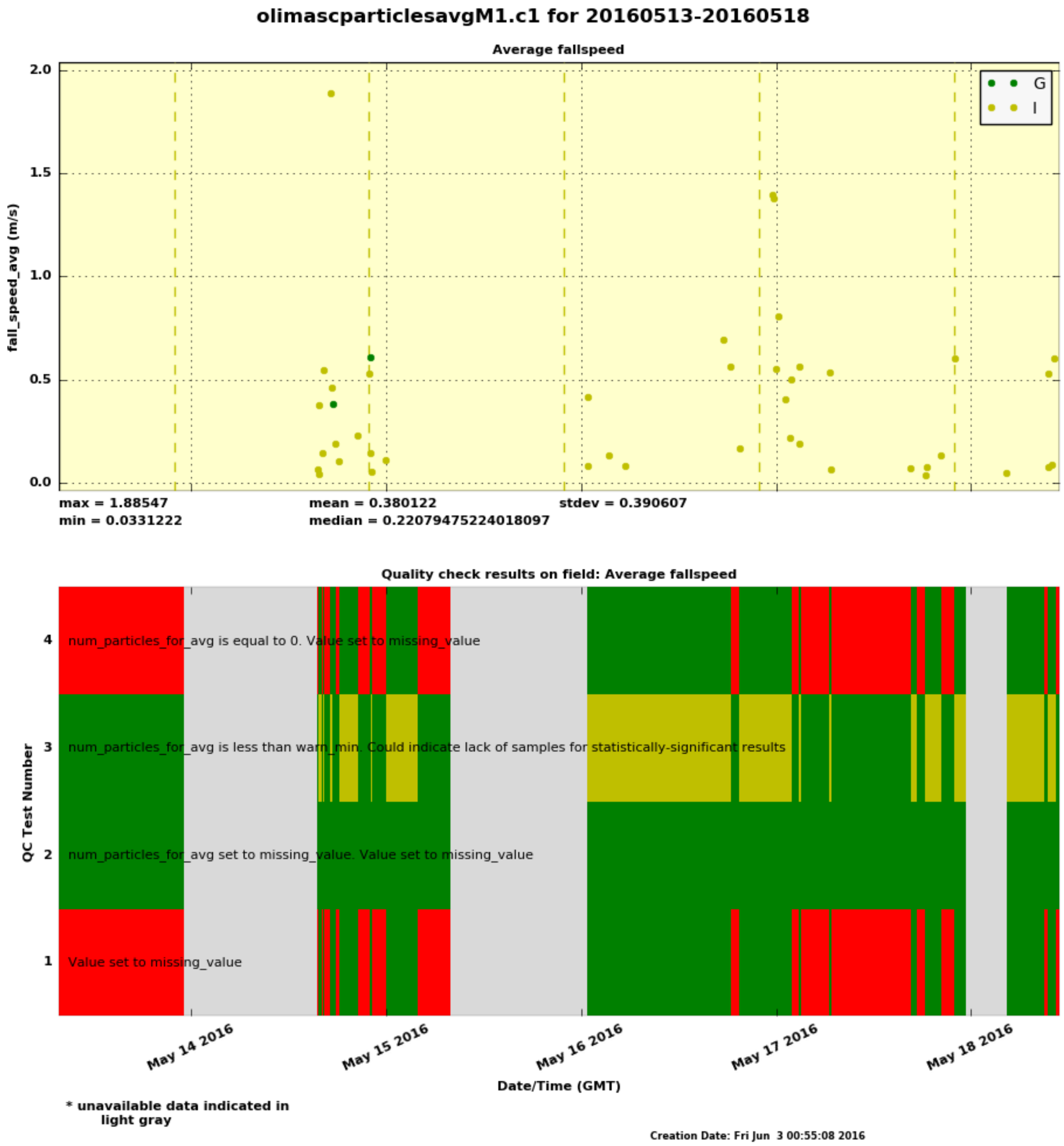

Figure 12. Retrieved average fall speed for a subset of particles that fell into each 5-min.-wide time bin (top) and quality bits (bottom) for data captured between May 13 and May 19, 2016. See Section 6.0 for description of colors.

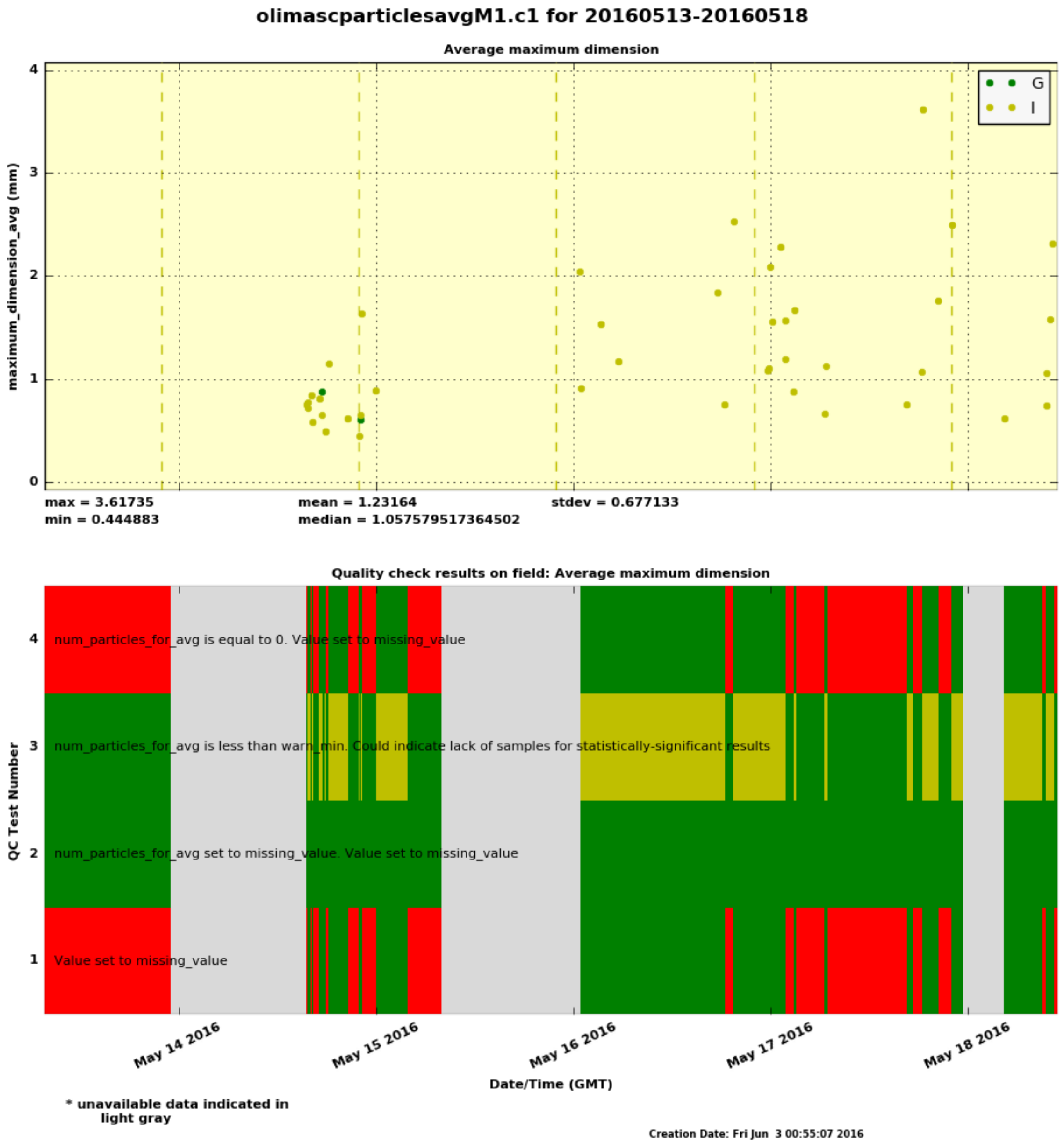

Figure 13. Retrieved average of maximum dimension for a subset of particles that fell into each 5-min.-wide time bin (top) and quality bits (bottom) for data captured between May 13 and May 19, 2016. See Section 6.0 for description of colors.

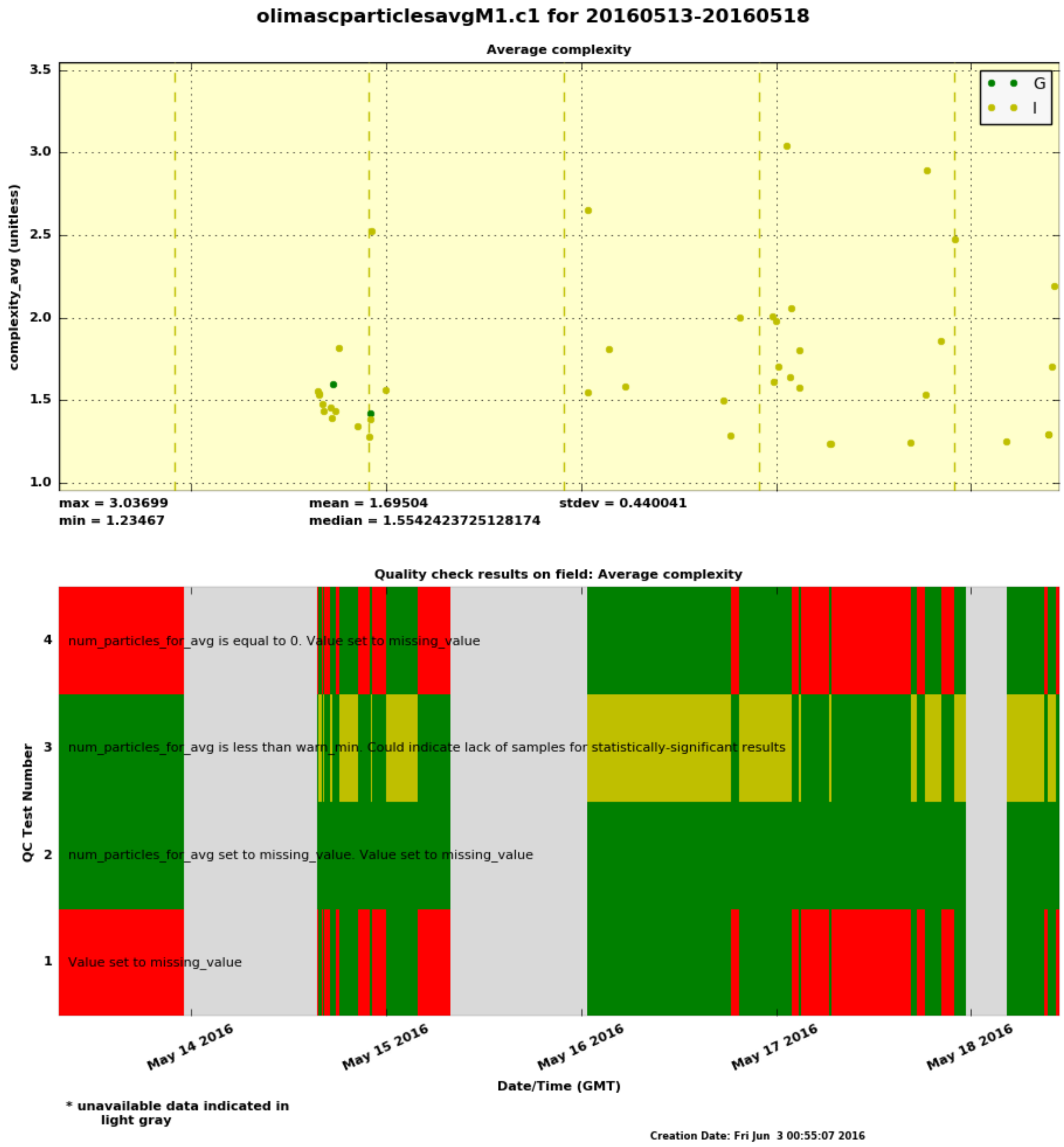

Figure 14. Retrieved average complexity for a subset of particles that fell into each 5-min.-wide time bin (top) and quality bits (bottom) for data captured between May 13 and May 19, 2016. See Section 6.0 for description of colors.

7.0 References

Garrett, TJ, SE Yuter, C Fallgatter, K Shkurko, SR Rhodes, and JL Endries. 2015. “Orientations and aspect ratios of falling snow.” *Geophysical Research Letters* 42(11): 4617-4622, doi:10.1002/2015GL064040.

Garrett, TJ, and SE Yuter. 2014. “Observed influence of riming, temperature, and turbulence on the fallspeed of solid precipitation.” *Geophysical Research Letters* 41(18): 6515-6522, doi:10.1002/2014GL061016.

Garrett, TJ, C Fallgatter, K Shkurko, and D Howlett. 2012. “Fall speed measurement and high-resolution multi-angle photography of hydrometeors in free fall.” *Atmospheric Measurement Techniques* 5(11): 2625-2633, doi:10.5194/amt-5-2625-2012.

OpenCV Documentation

Fitzgibbon, A, and A Fisher. 1995. “A Buyer’s Guide to Conic Fitting.” *Proceedings of the 5th British Machine Vision Conference*, Birmingham, United Kingdom, pp. 513-522.

Suzuki, S, and K Abe. 1985. “Topological Structural Analysis of Digitized Binary Images by Border Following.” *CVGIP* 30(1): pp 32-46.

Appendix A

Configuration XML File

```
<?xml version="1.0" standalone="yes"?>
<acquisitionConfiguration>
    <version val="0.156"/>
    <options>
        <maxNumCameras val="10"/>
        <maxNumCamFrames val="100"/>
        <numProcessorThreads val="0"/>
        <adapterId val="1"/>
        <COMPort val="com3"/>
        <verbose val="1"/>
    </options>
    <PNGHeader>
        <author>ARM</author>
        <copyright>null</copyright>
        <description>null</description>
    </PNGHeader>
    <statusLogOptions>
        <outStatusLogFilename val="oliMASCM1.a0.%Y%M%D.%h.raw_statusLog.txt"/>
        <intervalBetChecks val="60"/>
    </statusLogOptions>
    <fileOptions>
        <outDataInfoFilename val="oliMASCM1.a0.%Y%M%D.%h.raw_dataInfo.txt"/>
        <outImageInfoFilename val="oliMASCM1.a0.%Y%M%D.%h.raw_imgInfo.txt"/>
        <outConfigFilename val="oliMASCM1.a0.%Y%M%D.%h.raw_config.xml"/>
        <outDirectories num="1">
            <dir val=".\"/>
        </outDirectories>
        <outDirectorySplitNumFlakes val="0"/>
        <imageNameScheme val="oliMASCM1.a0.%Y%M%D.%h%m%s.id_%f_cam_%c.png"/>
        <configFilename val="acquisitionconfig.xml"/>
    </fileOptions>
    <camDeviceConfiguration>
        <camerasInfo num="3">
            <camera id="0">
                <GUID val="08:14:43:65:00:01:00:11"/>
                <autoExposure on="0" value="40" autoMode="0"/>
                <shutter on="1" value="40" autoMode="0"/>
                <gain on="1" value="0" autoMode="0"/>
                <trigger on="1" mode="15" polarity="0" source="0"/>
                <fieldOfViewInmm val="0.0306372549"/>
                <startUpInfo>
                    <videoFormat val="VIDF_7"/>
                    <videoMode val="VIDM_0"/>
                    <camFPS val="FPS_30"/>
                    <isoChannel val="4"/>
                    <dataTransferRate val="S800"/>
                    <format7Info>
                        <width val="2448"/>
                        <height val="2048"/>
                        <top val="0"/>
                        <left val="0"/>
                        <bottom val="0"/>
                        <right val="0"/>
                        <pixelFormat val="Y_MONO"/>
                        <packetSize val="2900"/>
                        <pixelsPerFrame val="5013504"/>
                        <bytesPerFrame val="5013504"/>
                        <numPackets val="1729"/>
                        <rawMode val="RM_GBRG"/>
                        <rawModeConversion val="RMC_BILINEAR_INTERPOLATION"/>
                    </format7Info>
                </startUpInfo>
            </camera>
            <camera id="1">
                <GUID val="08:14:43:65:00:01:00:0f"/>
                <autoExposure on="0" value="40" autoMode="0"/>
                <shutter on="1" value="40" autoMode="0"/>
                <gain on="1" value="0" autoMode="0"/>
                <trigger on="1" mode="15" polarity="0" source="0"/>
                <fieldOfViewInmm val="0.0306372549"/>
                <startUpInfo>
                    <videoFormat val="VIDF_7"/>
                    <videoMode val="VIDM_0"/>
                    <camFPS val="FPS_30"/>
                    <isoChannel val="39"/>
                    <dataTransferRate val="S800"/>
                    <format7Info>
                        <width val="2448"/>
                        <height val="2048"/>
                        <top val="0"/>
                        <left val="0"/>
                        <bottom val="0"/>
                        <right val="0"/>
                        <pixelFormat val="Y_MONO"/>
                        <packetSize val="2900"/>
                        <pixelsPerFrame val="5013504"/>
                        <bytesPerFrame val="5013504"/>
                        <numPackets val="1729"/>
                        <rawMode val="RM_GBRG"/>
                        <rawModeConversion val="RMC_BILINEAR_INTERPOLATION"/>
                    </format7Info>
                </startUpInfo>
            </camera>
            <camera id="2">
                <GUID val="08:14:43:65:00:01:00:10"/>
                <autoExposure on="0" value="40" autoMode="0"/>
                <shutter on="1" value="40" autoMode="0"/>
                <gain on="1" value="0" autoMode="0"/>
                <trigger on="1" mode="15" polarity="0" source="0"/>
                <fieldOfViewInmm val="0.0306372549"/>
                <startUpInfo>
                    <videoFormat val="VIDF_7"/>
                    <videoMode val="VIDM_0"/>
                    <camFPS val="FPS_30"/>
                    <isoChannel val="47"/>
                    <dataTransferRate val="S800"/>
                    <format7Info>
                        <width val="2448"/>
                        <height val="2048"/>
                        <top val="0"/>
                        <left val="0"/>
                        <bottom val="0"/>
                        <right val="0"/>
                        <pixelFormat val="Y_MONO"/>
                        <packetSize val="2900"/>
                        <pixelsPerFrame val="5013504"/>
                        <bytesPerFrame val="5013504"/>
                        <numPackets val="1729"/>
                        <rawMode val="RM_GBRG"/>
                        <rawModeConversion val="RMC_BILINEAR_INTERPOLATION"/>
                    </format7Info>
                </startUpInfo>
            </camera>
        </camerasInfo>
    </camDeviceConfiguration>
</acquisitionConfiguration>
```

Figure 15. Example of configuration XML file.

Appendix B

Default Analysis Parameters

```
{
    // Parameters for image analysis
    "imageAnalysisParameters":
    {
        // Additional amount to crop from each image before processing. Intended to
        // remove clutter sometimes found on the edges (like infra red sensors)
        "additionalImageCrop":
        {
            "top":    460,
            "bottom": 360,
            "left":   600,
            "right":  600
        },

        // Threshold used to determine whether pixel should be considered as
        // background based on its intensity. Represented in range [0, 1]
        "backgroundThreshold01":       0.03,    // = 7.6 / 255

        // To assess flake area, internal complexities are blurred with this parameter
        // which determines the amount of dilation and erosion during image processing
        // This avoids small local discontinuities to make a single flake.
        "lineFillInMicrons":           200,

        // Minimum acceptable average width for a flake (in microns)
        "minFlakeSizeInMicrons":       200,

        // Maximum acceptable length for a flake to touch the image frame edge
        // (in microns)
        "maxEdgeTouchLengthInMicrons": 500,

        // Minimum acceptable maximum pixel brightness in range [0, 1]. Darker flakes
        // tend to be out of focus
        "minMaxPixelIntensity01":      0.2,

        // Irregularities in the background or out-of-focus images have very low
        // internal variability. This threshold specifies the minimum variability
        // images must have
        "rangeIntensityThreshold01":   0.01961,      // = 5 / 255

        // Ignored for VAP
        "flagSaveCroppedImages":       0,
        // Flag whether to filter out of focus images. Should always be set to 1
        "flagRejectOutOfFocus":        1,

        // Threshold is a guess for the focus reject, not a hard and fast rule.
        // Lower values correspond to fewer 'rejects'
        "focusThreshold01":            0.01,

        // Identify a 'sweet' spot range where 'good' triggers happen, expressible as
        // a distance (in mm) from the top of the (uncropped) image. This will be
        // specific to each camera and its alignment. An easy way to find values
        // relies on a histogram of roi_bot_position for all particles.
        "boundingBoxThresholdInMM":
        {
            "bottomMin": 33,
            "bottomMax": 39
        },

        // Configuration options per camera, as set during capture. Implicitly ordered
        // by camera_id like 0, 1, 2
        "perCamera":
        [
            {
                // Horizontal FOV per pixel in microns per camera. These values are
                // for 5MP cameras with 16mm lens -> 75mm field of view for the entire
                // image
                "horizFOVPerPixelInMM": 0.030637255,   // = 75mm / 2448 pixels

                // Amount the image from this camera was cropped before being saved to
                // disk.
                "cropAtCapture":
                {
                    "top":    0,
                    "bottom": 0,
                    "left":   0,
                    "right":  0
                }
            },
            {
                "horizFOVPerPixelInMM": 0.030637255,
                "cropAtCapture":
                {
                    "top":    0,
                    "bottom": 0,
                    "left":   0,
                    "right":  0
                }
            },
            {
                "horizFOVPerPixelInMM": 0.030637255,
                "cropAtCapture":
                {
                    "top":    0,
                    "bottom": 0,
                    "left":   0,
                    "right":  0
                }
            }
```

```
        ],
    },

    // Parameters for averaging particle data into time bins
    "timeBinningParameters":
    {
        // How wide is each time bin in seconds
        "binWidthInSec":          300,

        // Maximum particle fallspeed required for the particle to be averaged into a
        // bin (m/s). If particle travels faster, it is ignored during average
        // computation
        "maxFallSpeedInMetersPS": 5,

        // Minimum number of particles used for average required for stats to be
        // 'good'. Technically this value depends on how wide time bins are (for an
        // hour, 60 is suggested)
        "minNumParticlesPerBin":  10
    }
}
```

Figure 16. Example of the JSON configuration file used by the VAP to identify default values for parameters needed for processing. Note that `valid_min`, `valid_max`, `warn_min`, and `warn_max` values within appropriate datastreams will replace these before actual computation. Comments next to each variable indicate usage. The final values used by the VAP are written into `anal_config_json` global attribute within `mascparticlesM1.c1` datastream.

Appendix C

Algorithm Pseudocode

The pseudocode for the entire VAP, starting with the entry point and running through image analysis and aggregations into per-particle and per-time-bin outputs.

C.1 Processing Entry Point

```
Get input datastream, masc.b1
Get output datastreams, mascparticles.c1, mascparticlesavg.c1
Load filter values for diagnostic parameters
Load default analysis configuration parameters from defImgAnlParams.json configuration, and replace appropriate parameters with filter values
all_particles = []
Loop over all read entries (particles) in masc.b1:
|      Clear appropriate output particle data in mascparticles.c1
|      Create particle structure
|      Loop over all cameras (3):
|      |      Build image filename given particle information
|      |      If the file exists, add filename into particle structure and count as good image
|      If number of good images == 0, continue onto next particle
|      Analyze particle
|      Loop over all images in particle:
|      |      Set per-image output data in mascparticles.c1 datastream for this particle
|      Set per-particle aggregate data in mascparticles.c1 datastream for this particle
|      Add particle into all_particles list
Bin particles within all_particles list
If have some bins:
       Build array of bin centers
       Allocate data for all bins
Loop over all bins:
|      If bin is not empty:
|             Set output bin data in mascparticlesavg.c1 datastream for this bin
```

C.2 Analyzing a Particle

This describes a function call (`Analyze particle` above) that is applied to individual particle structures, which holds particle data gathered from masc.b1 (image structures, fallspeed, etc.) and aggregated image analysis. Each image structure holds image filename and image analysis parameters.

The averaging only averages images that have passed the quality checks. This may result in no images averaged successfully, even though each image may have some analysis parameters saved.

```
Input:  particle structure
Output: per-particle and per-image analysis parameters
------------------------------
Loop over particle images:
|   per_image_data = Analyze image
particle_avg_data = Average(per_image_data)
```

C.3 Analyzing a Single Image

This function (`Analyze image` above) is the heart of the processing.

```
Input:  image structure
Output: analysis parameters
------------------------------
Load image based on filename
Mask out background:
    Crop the image an additional amount (based on analysis configuration)
    Set all background pixels to back threshold
Otsu thresholding:
       Apply gaussian blur
       Threshold (binary + otsu) - see Otsu's binarization here:
http://docs.opencv.org/trunk/d7/d4d/tutorial_py_thresholding.html#gsc.tab=0
       Find edges using Canny. Returns image with edges in it
       Dilate the edges
       Find contours in dilated image and fill them
       Erode result and return
Mark non-background pixels (save as per-pixel mask)
Compute intensity range per pixel (max-min within 3x3 neighborhood of pixels)
Find contours in eroded image
Loop over each contour:          // contour is ROI
|      Create image mask with filled in contour
|      Analyze image within contour
|      Compute features used for quality checks (via thresholding)
|      If features pass thresholds, set appropriate quality flag
Return analysis features
```

C.4 Image Feature Computation

This describes the function (`Analyze image within contour` above) that is applied to each contour within the image. The image analysis, which relies on OpenCV, results in all features, both scientific and diagnostic, as reported by the VAP. See Python code below for more details.

```
Input:  image, feature contour to analyze, mask identifying non-background pixels
Output: image features
------------------------------
# Figure out offsets due to image cropping at capture to offset things
leftOffset, topOffset

# get bounding box (x, y, w, h)
# (x, y) is top left corner of the box
aabb     = list(cv2.boundingRect(contour))
aabb[0] += leftOffset
aabb[1] += topOffset

# area (number of pixels within the contour)
contourArea = cv2.countNonZero(contourMask)

# perimeter
contourPerimeter = cv2.arcLength(contour, True)

# fit ellipse ((x, y), (major, minor), angle)
ellipse        = list(cv2.fitEllipse(contour))
ellipse[0]     = list(ellipse[0])
ellipse[0][0] += leftOffset
ellipse[0][1] += topOffset

# mask of non-background pixels within our contour
contourNonBackMask = cv2.bitwise_and(contourMask, nonBackMask)

# number of pixels brighter than background within our mask
flakeArea = cv2.countNonZero(contourNonBackMask)

# average intensity of the flake
flakeIntensity = cv2.mean(self._image, mask = contourNonBackMask)
flakeIntensity = flakeIntensity[0] / 255.

# maximum intensity of the flake
m, maxIntensity, ml, Ml = cv2.minMaxLoc(self._image, mask = contourNonBackMask)
maxIntensity = maxIntensity / 255.

# get the range of this flake's intensity
rangeIntensity = cv2.mean(imgIntensityRange, mask = contourNonBackMask)
rangeIntensity = rangeIntensity[0] / 255.

# fraction of enclosed area that is brighter than the background
partialArea = cv2.countNonZero(contourNonBackMask) / float(contourArea)

# estimate for a degree of focus, on the basis that in focus flakes are both
# bright and variable
focus     = flakeIntensity * rangeIntensity
areaFocus = flakeArea * focus

# length of flake that touches edge of image frame
borderContour = cv2.bitwise_and(contourMask, contourMask, mask = borderPixels)
edgeTouch     = cv2.countNonZero(borderContour)
```

Figure 17. Python code that uses OpenCV to compute features returned by the VAP. This function analyzes a particular contour within the image.

C.5 Aggregating Particles into Bins

This describes the function (**`Bin particles`** above) that is applied to a list of particles that has been successfully analyzed. Particles that have no average values associated with them, will be ignored from being averaged. This will be marked appropriately in the particle counters per bin. If no particles overlap with a bin, it is not generated.

```
Input: list of particles
Output: list of bins
Find min and max time for input particles
bin_list = Initialize bins that span the input range (all bins here are empty)
particles_to_avg = []
current_bin_id = -1
Loop over all particles:
|      particle_bin = Project particle into bins
|      if particle_bin == current_bin_id:
|             add particle to particles_to_avg
|      else:
|             bin_list[current_bin_id] = Average particles_to_avg
|             current_bin_id = particle_bin
|             particles_to_avg = particle
if current_bin_id:
       bin_list[current_bin_id] = Average particles_to_avg
return bin_list
```

In this case, the Average function called several times above, actually performs the average of particle analysis data.

```
Input: list of particles
Output: average
num_good = 0
Loop over all particles:
|      If particle passes filter, num_good += 1
If num_good == 0, return NULL
Reset ave_values
Loop over all particles:
|      If particle passes filter, accumulate into ave_values
Compute flatness within ave_values
Return ave_values / num_good
```

Appendix D

NetCDF Header for mascparticlesM1.c1

An example header from the MASC_FLAKE_ANAL VAP is given below:

```
netcdf olimascparticlesM1.c1.20160301.065053 {
dimensions:
        time = UNLIMITED ; // (51 currently)
        num_elems_roi_position = 2 ;
        camera = 3 ;
variables:
        int base_time ;
                base_time:string = "2016-03-01 00:00:00 0:00" ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-1-1 0:00:00 0:00" ;
                base_time:ancillary_variables = "time_offset" ;
        double time_offset(time) ;
                time_offset:long_name = "Time offset from base_time" ;
                time_offset:units = "seconds since 2016-03-01 00:00:00 0:00" ;
                time_offset:ancillary_variables = "base_time" ;
        double time(time) ;
                time:long_name = "Time offset from midnight" ;
                time:units = "seconds since 2016-03-01 00:00:00 0:00" ;
                time:calendar = "gregorian" ;
                time:standard_name = "time" ;
        int snowflake_id(time) ;
                snowflake_id:long_name = "Snowflake ID number" ;
                snowflake_id:units = "unitless" ;
                snowflake_id:source = "olimascM1.b1:snowflake_id" ;
        float snowflake_fall_speed(time) ;
                snowflake_fall_speed:long_name = "Fall speed of captured particle" ;
                snowflake_fall_speed:units = "m/s" ;
                snowflake_fall_speed:source = "olimascM1.b1:snowflake_fall_speed" ;
                snowflake_fall_speed:valid_min = 0.f ;
                snowflake_fall_speed:valid_max = 10.f ;
                snowflake_fall_speed:warn_max = 5.f ;
                snowflake_fall_speed:missing_value = -9999.f ;
                snowflake_fall_speed:ancillary_variables = "qc_snowflake_fall_speed" ;
                snowflake_fall_speed:comment = "Fall speed of captured particle. Note
the warn_max check is an additional check, which is necessary for averaging particle
analysis into time bins, so if it fails the data is still ok" ;
        int qc_snowflake_fall_speed(time) ;
                qc_snowflake_fall_speed:long_name = "Quality check results on field:
Fall speed of captured particle" ;
                qc_snowflake_fall_speed:units = "unitless" ;
                qc_snowflake_fall_speed:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_snowflake_fall_speed:flag_method = "bit" ;
                qc_snowflake_fall_speed:bit_1_description = "Value is equal to
missing_value." ;
                qc_snowflake_fall_speed:bit_1_assessment = "Bad" ;
                qc_snowflake_fall_speed:bit_2_description = "Value is less than the
valid_min." ;
                qc_snowflake_fall_speed:bit_2_assessment = "Bad" ;
                qc_snowflake_fall_speed:bit_3_description = "Value is greater than the
valid_max." ;
                qc_snowflake_fall_speed:bit_3_assessment = "Bad" ;
                qc_snowflake_fall_speed:bit_4_description = "Value is greater than the
warn_max. It is used to filter out particles before binning in time." ;
                qc_snowflake_fall_speed:bit_4_assessment = "Indeterminate" ;
        int camera_id(time, camera) ;
                camera_id:long_name = "Camera ID of image taken" ;
                camera_id:units = "unitless" ;
                camera_id:source = "olimascM1.b1:camera_id" ;
                camera_id:missing_value = -9999 ;
        float maximum_dimension(time, camera) ;
                maximum_dimension:long_name = "Particle\'s maximum dimension" ;
                maximum_dimension:units = "mm" ;
                maximum_dimension:ancillary_variables = "qc_maximum_dimension" ;
                maximum_dimension:missing_value = -9999.f ;
                maximum_dimension:cell_methods = "time: point" ;
                maximum_dimension:comment = "Maximum dimension of the particle in the
image. Is major axis of best-fit ellipse." ;
        int qc_maximum_dimension(time, camera) ;
                qc_maximum_dimension:long_name = "Quality check results on field:
Particle\'s maximum dimension" ;
                qc_maximum_dimension:units = "unitless" ;
                qc_maximum_dimension:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_maximum_dimension:flag_method = "bit" ;
                qc_maximum_dimension:bit_1_description = "snowflake_fall_speed is set
to bad value. Value set to missing_value" ;
                qc_maximum_dimension:bit_1_assessment = "Bad" ;
                qc_maximum_dimension:bit_2_description = "camera_id for this camera
set to missing_value. Value set to missing_value" ;
                qc_maximum_dimension:bit_2_assessment = "Bad" ;
                qc_maximum_dimension:bit_3_description = "Image file for this camera
is missing (can\'t be opened). Value set to missing_value" ;
                qc_maximum_dimension:bit_3_assessment = "Bad" ;
                qc_maximum_dimension:bit_4_description = "No particle detected in
image. Value set to missing_value" ;
                qc_maximum_dimension:bit_4_assessment = "Bad" ;
                qc_maximum_dimension:bit_5_description = "A property of the particle
most in focus is less than warn_min or greater than warn_max limit." ;
                qc_maximum_dimension:bit_5_assessment = "Indeterminate" ;
        float particle_area(time, camera) ;
                particle_area:long_name = "Particle area" ;
                particle_area:units = "mm^2" ;
                particle_area:ancillary_variables = "qc_particle_area" ;
                particle_area:comment = "Area of a particle including interior holes,
unlike geometric_cross_section" ;
                particle_area:missing_value = -9999.f ;
                particle_area:cell_methods = "time: point" ;
        int qc_particle_area(time, camera) ;
                qc_particle_area:long_name = "Quality check results on field: Particle
area" ;
                qc_particle_area:units = "unitless" ;
                qc_particle_area:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
```

```
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_particle_area:flag_method = "bit" ;
                qc_particle_area:bit_1_description = "snowflake_fall_speed is set to
bad value. Value set to missing_value" ;
                qc_particle_area:bit_1_assessment = "Bad" ;
                qc_particle_area:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_particle_area:bit_2_assessment = "Bad" ;
                qc_particle_area:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_particle_area:bit_3_assessment = "Bad" ;
                qc_particle_area:bit_4_description = "No particle detected in image.
Value set to missing_value" ;
                qc_particle_area:bit_4_assessment = "Bad" ;
                qc_particle_area:bit_5_description = "A property of the particle most
in focus is less than warn_min or greater than warn_max limit." ;
                qc_particle_area:bit_5_assessment = "Indeterminate" ;
        float particle_edge_touch(time, camera) ;
                particle_edge_touch:long_name = "Particle overlap with image edge" ;
                particle_edge_touch:units = "mm" ;
                particle_edge_touch:ancillary_variables = "qc_particle_edge_touch" ;
                particle_edge_touch:warn_max = 0.5f ;
                particle_edge_touch:missing_value = -9999.f ;
                particle_edge_touch:cell_methods = "time: point" ;
                particle_edge_touch:comment = "How much of the particle overlaps with
the image edge. Measured as the length along the perimeter accounting for all sides."
;
        int qc_particle_edge_touch(time, camera) ;
                qc_particle_edge_touch:long_name = "Quality check results on field:
Particle overlap with image edge" ;
                qc_particle_edge_touch:units = "unitless" ;
                qc_particle_edge_touch:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_particle_edge_touch:flag_method = "bit" ;
                qc_particle_edge_touch:bit_1_description = "snowflake_fall_speed is
set to bad value. Value set to missing_value" ;
                qc_particle_edge_touch:bit_1_assessment = "Bad" ;
                qc_particle_edge_touch:bit_2_description = "camera_id for this camera
set to missing_value. Value set to missing_value" ;
                qc_particle_edge_touch:bit_2_assessment = "Bad" ;
                qc_particle_edge_touch:bit_3_description = "Image file for this camera
is missing (can\'t be opened). Value set to missing_value" ;
                qc_particle_edge_touch:bit_3_assessment = "Bad" ;
                qc_particle_edge_touch:bit_4_description = "No particle detected in
image. Value set to missing_value" ;
                qc_particle_edge_touch:bit_4_assessment = "Bad" ;
                qc_particle_edge_touch:bit_5_description = "A property of the particle
most in focus is less than warn_min or greater than warn_max limit." ;
                qc_particle_edge_touch:bit_5_assessment = "Indeterminate" ;
                qc_particle_edge_touch:bit_6_description = "Value is greater than
warn_max" ;
                qc_particle_edge_touch:bit_6_assessment = "Indeterminate" ;
        float area_eq_radius(time, camera) ;
                area_eq_radius:long_name = "Particle\'s area equivalent radius" ;
                area_eq_radius:units = "mm" ;
                area_eq_radius:ancillary_variables = "qc_area_eq_radius" ;
                area_eq_radius:missing_value = -9999.f ;
                area_eq_radius:cell_methods = "time: point" ;
                area_eq_radius:comment = "Area equivalent radius of the flake. Radius
of a circle that has the same area as the flake (excluding background pixels)" ;
```

```
        int qc_area_eq_radius(time, camera) ;
                qc_area_eq_radius:long_name = "Quality check results on field: 
Particle\'s area equivalent radius" ;
                qc_area_eq_radius:units = "unitless" ;
                qc_area_eq_radius:description = "This field contains bit packed 
integer values, where each bit represents a QC test on the data. Non-zero bits 
indicate the QC condition given in the description for those bits; a value of 0 (no 
bits set) indicates the data has not failed any QC tests." ;
                qc_area_eq_radius:flag_method = "bit" ;
                qc_area_eq_radius:bit_1_description = "snowflake_fall_speed is set to 
bad value. Value set to missing_value" ;
                qc_area_eq_radius:bit_1_assessment = "Bad" ;
                qc_area_eq_radius:bit_2_description = "camera_id for this camera set 
to missing_value. Value set to missing_value" ;
                qc_area_eq_radius:bit_2_assessment = "Bad" ;
                qc_area_eq_radius:bit_3_description = "Image file for this camera is 
missing (can\'t be opened). Value set to missing_value" ;
                qc_area_eq_radius:bit_3_assessment = "Bad" ;
                qc_area_eq_radius:bit_4_description = "No particle detected in image. 
Value set to missing_value" ;
                qc_area_eq_radius:bit_4_assessment = "Bad" ;
                qc_area_eq_radius:bit_5_description = "A property of the particle most 
in focus is less than warn_min or greater than warn_max limit." ;
                qc_area_eq_radius:bit_5_assessment = "Indeterminate" ;
        float perimeter(time, camera) ;
                perimeter:long_name = "Particle perimeter" ;
                perimeter:units = "mm" ;
                perimeter:ancillary_variables = "qc_perimeter" ;
                perimeter:missing_value = -9999.f ;
                perimeter:cell_methods = "time: point" ;
        int qc_perimeter(time, camera) ;
                qc_perimeter:long_name = "Quality check results on field: Particle 
perimeter" ;
                qc_perimeter:units = "unitless" ;
                qc_perimeter:description = "This field contains bit packed integer 
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC 
condition given in the description for those bits; a value of 0 (no bits set) 
indicates the data has not failed any QC tests." ;
                qc_perimeter:flag_method = "bit" ;
                qc_perimeter:bit_1_description = "snowflake_fall_speed is set to bad 
value. Value set to missing_value" ;
                qc_perimeter:bit_1_assessment = "Bad" ;
                qc_perimeter:bit_2_description = "camera_id for this camera set to 
missing_value. Value set to missing_value" ;
                qc_perimeter:bit_2_assessment = "Bad" ;
                qc_perimeter:bit_3_description = "image file for this camera is 
missing (can\'t be opened). Value set to missing_value" ;
                qc_perimeter:bit_3_assessment = "Bad" ;
                qc_perimeter:bit_4_description = "No particle detected in image. Value 
set to missing_value" ;
                qc_perimeter:bit_4_assessment = "Bad" ;
                qc_perimeter:bit_5_description = "A property of the particle most in 
focus is less than warn_min or greater than warn_max limit." ;
                qc_perimeter:bit_5_assessment = "Indeterminate" ;
        float orientation(time, camera) ;
                orientation:long_name = "Particle orientation" ;
                orientation:units = "degree" ;
                orientation:ancillary_variables = "qc_orientation" ;
                orientation:missing_value = -9999.f ;
                orientation:cell_methods = "time: point" ;
                orientation:comment = "Particle orientation. Measures absolute value 
of angle from horizontal to the major axis of best-fit ellipse." ;
        int qc_orientation(time, camera) ;
```

```
                qc_orientation:long_name = "Quality check results on field: Particle
orientation" ;
                qc_orientation:units = "unitless" ;
                qc_orientation:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_orientation:flag_method = "bit" ;
                qc_orientation:bit_1_description = "snowflake_fall_speed is set to bad
value. Value set to missing_value" ;
                qc_orientation:bit_1_assessment = "Bad" ;
                qc_orientation:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_orientation:bit_2_assessment = "Bad" ;
                qc_orientation:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_orientation:bit_3_assessment = "Bad" ;
                qc_orientation:bit_4_description = "No particle detected in image.
Value set to missing_value" ;
                qc_orientation:bit_4_assessment = "Bad" ;
                qc_orientation:bit_5_description = "A property of the particle most in
focus is less than warn_min or greater than warn_max limit." ;
                qc_orientation:bit_5_assessment = "Indeterminate" ;
        float aspect_ratio(time, camera) ;
                aspect_ratio:long_name = "Particle aspect ratio" ;
                aspect_ratio:units = "unitless" ;
                aspect_ratio:ancillary_variables = "qc_aspect_ratio" ;
                aspect_ratio:missing_value = -9999.f ;
                aspect_ratio:cell_methods = "time: point" ;
                aspect_ratio:comment = "Particle aspect ratio = minor axis / major
axis, where each comes from best-fit ellipse" ;
        int qc_aspect_ratio(time, camera) ;
                qc_aspect_ratio:long_name = "Quality check results on field: Particle
aspect ratio" ;
                qc_aspect_ratio:units = "unitless" ;
                qc_aspect_ratio:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_aspect_ratio:flag_method = "bit" ;
                qc_aspect_ratio:bit_1_description = "snowflake_fall_speed is set to
bad value. Value set to missing_value" ;
                qc_aspect_ratio:bit_1_assessment = "Bad" ;
                qc_aspect_ratio:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_aspect_ratio:bit_2_assessment = "Bad" ;
                qc_aspect_ratio:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_aspect_ratio:bit_3_assessment = "Bad" ;
                qc_aspect_ratio:bit_4_description = "No particle detected in image.
Value set to missing_value" ;
                qc_aspect_ratio:bit_4_assessment = "Bad" ;
                qc_aspect_ratio:bit_5_description = "A property of the particle most
in focus is less than warn_min or greater than warn_max limit." ;
                qc_aspect_ratio:bit_5_assessment = "Indeterminate" ;
        float complexity(time, camera) ;
                complexity:long_name = "Particle complexity" ;
                complexity:units = "unitless" ;
                complexity:ancillary_variables = "qc_complexity" ;
                complexity:missing_value = -9999.f ;
                complexity:cell_methods = "time: point" ;
                complexity:comment = "Particle complexity = perimeter / (2pi *
area_eq_radius) * (1 + mean_pixel_intensity_variability)" ;
```

```
        int qc_complexity(time, camera) ;
                qc_complexity:long_name = "Quality check results on field: Particle
complexity" ;
                qc_complexity:units = "unitless" ;
                qc_complexity:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_complexity:flag_method = "bit" ;
                qc_complexity:bit_1_description = "snowflake_fall_speed is set to bad
value. Value set to missing_value" ;
                qc_complexity:bit_1_assessment = "Bad" ;
                qc_complexity:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_complexity:bit_2_assessment = "Bad" ;
                qc_complexity:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_complexity:bit_3_assessment = "Bad" ;
        float geometric_cross_section(time, camera) ;
                geometric_cross_section:long_name = "Particle geometric cross section"
;
                geometric_cross_section:units = "mm^2" ;
                geometric_cross_section:ancillary_variables =
"qc_geometric_cross_section" ;
                geometric_cross_section:comment = "Area of a particle excluding any
holes, unlike particle_area" ;
                geometric_cross_section:warn_min = 0.04f ;
                geometric_cross_section:missing_value = -9999.f ;
                geometric_cross_section:cell_methods = "time: point" ;
        int qc_geometric_cross_section(time, camera) ;
                qc_geometric_cross_section:long_name = "Quality check results on
field: Particle geometric cross section" ;
                qc_geometric_cross_section:units = "unitless" ;
                qc_geometric_cross_section:description = "This field contains bit
packed integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_geometric_cross_section:flag_method = "bit" ;
                qc_geometric_cross_section:bit_1_description = "snowflake_fall_speed
is set to bad value. Value set to missing_value" ;
                qc_geometric_cross_section:bit_1_assessment = "Bad" ;
                qc_geometric_cross_section:bit_2_description = "camera_id for this
camera set to missing_value. Value set to missing_value" ;
                qc_geometric_cross_section:bit_2_assessment = "Bad" ;
                qc_geometric_cross_section:bit_3_description = "Image file for this
camera is missing (can\'t be opened). Value set to missing_value" ;
                qc_geometric_cross_section:bit_3_assessment = "Bad" ;
                qc_geometric_cross_section:bit_4_description = "No particle detected
in image. Value set to missing_value" ;
                qc_geometric_cross_section:bit_4_assessment = "Bad" ;
                qc_geometric_cross_section:bit_5_description = "A property of the
particle most in focus is less than warn_min or greater than warn_max limit." ;
                qc_geometric_cross_section:bit_5_assessment = "Indeterminate" ;
                qc_geometric_cross_section:bit_6_description = "Value is less than
warn_min" ;
                qc_geometric_cross_section:bit_6_assessment = "Indeterminate" ;
        float mean_pixel_intensity(time, camera) ;
                mean_pixel_intensity:long_name = "Particle mean pixel intensity" ;
                mean_pixel_intensity:units = "unitless" ;
                mean_pixel_intensity:ancillary_variables = "qc_mean_pixel_intensity" ;
                mean_pixel_intensity:warn_min = 0.2f ;
                mean_pixel_intensity:missing_value = -9999.f ;
                mean_pixel_intensity:cell_methods = "time: point" ;
                mean_pixel_intensity:comment = "Mean pixel intensity of the particle
most in focus within the image. Ignores background pixels" ;
        int qc_mean_pixel_intensity(time, camera) ;
                qc_mean_pixel_intensity:long_name = "Quality check results on field:
Particle mean pixel intensity" ;
                qc_mean_pixel_intensity:units = "unitless" ;
                qc_mean_pixel_intensity:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity:flag_method = "bit" ;
                qc_mean_pixel_intensity:bit_1_description = "snowflake_fall_speed is
set to bad value. Value set to missing_value" ;
                qc_mean_pixel_intensity:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity:bit_2_description = "camera_id for this camera
set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity:bit_3_description = "Image file for this
camera is missing (can\'t be opened). Value set to missing_value" ;
                qc_mean_pixel_intensity:bit_3_assessment = "Bad" ;
                qc_mean_pixel_intensity:bit_4_description = "No particle detected in
image. Value set to missing_value" ;
                qc_mean_pixel_intensity:bit_4_assessment = "Bad" ;
                qc_mean_pixel_intensity:bit_5_description = "A property of the
particle most in focus is less than warn_min or greater than warn_max limit." ;
                qc_mean_pixel_intensity:bit_5_assessment = "Indeterminate" ;
                qc_mean_pixel_intensity:bit_6_description = "Value is less than the
warn_min" ;
                qc_mean_pixel_intensity:bit_6_assessment = "Indeterminate" ;
        float mean_pixel_intensity_variability(time, camera) ;
                mean_pixel_intensity_variability:long_name = "Variability of
particle\'s mean pixel intensity" ;
                mean_pixel_intensity_variability:units = "unitless" ;
                mean_pixel_intensity_variability:ancillary_variables =
"qc_mean_pixel_intensity_variability" ;
                mean_pixel_intensity_variability:warn_min = 0.019f ;
                mean_pixel_intensity_variability:missing_value = -9999.f ;
                mean_pixel_intensity_variability:cell_methods = "time: point" ;
                mean_pixel_intensity_variability:comment = "Variability of the mean
pixel intensity of the particle most in focus within the image. Ignores background
pixels" ;
        int qc_mean_pixel_intensity_variability(time, camera) ;
                qc_mean_pixel_intensity_variability:long_name = "Quality check results
on field: Variability of particle\'s mean pixel intensity" ;
                qc_mean_pixel_intensity_variability:units = "unitless" ;
                qc_mean_pixel_intensity_variability:description = "This field contains
bit packed integer values, where each bit represents a QC test on the data. Non-zero
bits indicate the QC condition given in the description for those bits; a value of 0
(no bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity_variability:flag_method = "bit" ;
                qc_mean_pixel_intensity_variability:bit_1_description =
"snowflake_fall_speed is set to bad value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability:bit_2_description = "camera_id for
this camera set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability:bit_3_description = "Image file
for this camera is missing (can\'t be opened). Value set to missing_value" ;
                qc_mean_pixel_intensity_variability:bit_3_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability:bit_4_description = "No particle
detected in image. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability:bit_4_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability:bit_5_description = "A property of
the particle most in focus is less than warn_min or greater than warn_max limit." ;
                qc_mean_pixel_intensity_variability:bit_5_assessment = "Indeterminate"
;
                qc_mean_pixel_intensity_variability:bit_6_description = "Value is less
than the warn_min" ;
                qc_mean_pixel_intensity_variability:bit_6_assessment = "Indeterminate"
;
        float roi_focus(time, camera) ;
                roi_focus:long_name = "Focus estimate for region of interest" ;
                roi_focus:units = "unitless" ;
                roi_focus:ancillary_variables = "qc_roi_focus" ;
                roi_focus:warn_min = 0.01f ;
                roi_focus:missing_value = -9999.f ;
                roi_focus:cell_methods = "time: point" ;
                roi_focus:comment = "Focus estimate for region of interest =
mean_pixel_intensity * mean_pixel_intensity_variability. Quality bit set to bad when
focus below warn_min (data still kept)" ;
        int qc_roi_focus(time, camera) ;
                qc_roi_focus:long_name = "Quality check results on field: Focus
estimate for region of interest" ;
                qc_roi_focus:units = "unitless" ;
                qc_roi_focus:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_roi_focus:flag_method = "bit" ;
                qc_roi_focus:bit_1_description = "snowflake_fall_speed is set to bad
value. Value set to missing_value" ;
                qc_roi_focus:bit_1_assessment = "Bad" ;
                qc_roi_focus:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_roi_focus:bit_2_assessment = "Bad" ;
                qc_roi_focus:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_roi_focus:bit_3_assessment = "Bad" ;
                qc_roi_focus:bit_4_description = "No particle detected in image. Value
set to missing_value" ;
                qc_roi_focus:bit_4_assessment = "Bad" ;
                qc_roi_focus:bit_5_description = "A property of the particle most in
focus is less than warn_min or greater than warn_max limit." ;
                qc_roi_focus:bit_5_assessment = "Indeterminate" ;
                qc_roi_focus:bit_6_description = "Value is less than the warn_min.
Check uses formula: round(roi_focus * 100)/100 < warn_min" ;
                qc_roi_focus:bit_6_assessment = "Indeterminate" ;
        int num_objects(time, camera) ;
                num_objects:long_name = "Number of objects within image" ;
                num_objects:units = "unitless" ;
                num_objects:ancillary_variables = "qc_num_objects" ;
                num_objects:missing_value = -9999 ;
                num_objects:cell_methods = "time: point" ;
                num_objects:comment = "Number of objects found within this image. The
stored analysis values are for the particle most in focus" ;
        int qc_num_objects(time, camera) ;
                qc_num_objects:long_name = "Quality check results on field: Number of
objects within image" ;
                qc_num_objects:units = "unitless" ;
                qc_num_objects:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_num_objects:flag_method = "bit" ;
                qc_num_objects:bit_1_description = "snowflake_fall_speed is set to bad
value. Value set to missing_value" ;
                qc_num_objects:bit_1_assessment = "Bad" ;
                qc_num_objects:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_num_objects:bit_2_assessment = "Bad" ;
                qc_num_objects:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_num_objects:bit_3_assessment = "Bad" ;
                qc_num_objects:bit_4_description = "No particle detected in image.
Value set to missing_value" ;
                qc_num_objects:bit_4_assessment = "Bad" ;
                qc_num_objects:bit_5_description = "A property of the particle most in
focus is less than warn_min or greater than warn_max limit. Value set to
missing_value" ;
                qc_num_objects:bit_5_assessment = "Bad" ;
        float roi_position(time, camera, num_elems_roi_position) ;
                roi_position:long_name = "Position of center of region of interest" ;
                roi_position:units = "mm" ;
                roi_position:ancillary_variables = "qc_roi_position" ;
                roi_position:missing_value = -9999.f ;
                roi_position:cell_methods = "time: point" ;
                roi_position:comment = "Region of interest, position (x,y) to the
center of ROI measured from top-left corner of the image." ;
        int qc_roi_position(time, camera, num_elems_roi_position) ;
                qc_roi_position:long_name = "Quality check results on field: Position
of center of region of interest" ;
                qc_roi_position:units = "unitless" ;
                qc_roi_position:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_roi_position:flag_method = "bit" ;
                qc_roi_position:bit_1_description = "snowflake_fall_speed is set to
bad value. Value set to missing_value" ;
                qc_roi_position:bit_1_assessment = "Bad" ;
                qc_roi_position:bit_2_description = "camera_id for this camera set to
missing_value. Value set to missing_value" ;
                qc_roi_position:bit_2_assessment = "Bad" ;
                qc_roi_position:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_roi_position:bit_3_assessment = "Bad" ;
                qc_roi_position:bit_4_description = "No particle detected in image.
Value set to missing_value" ;
                qc_roi_position:bit_4_assessment = "Bad" ;
                qc_roi_position:bit_5_description = "A property of the particle most
in focus is less than warn_min or greater than warn_max limit." ;
                qc_roi_position:bit_5_assessment = "Indeterminate" ;
        float roi_bot_position(time, camera) ;
                roi_bot_position:long_name = "Location of region of interest\'s bottom
border" ;
                roi_bot_position:units = "mm" ;
                roi_bot_position:ancillary_variables = "qc_roi_bot_position" ;
                roi_bot_position:warn_min = 32.f ;
                roi_bot_position:warn_max = 36.f ;
                roi_bot_position:missing_value = -9999.f ;
                roi_bot_position:cell_methods = "time: point" ;
                roi_bot_position:comment = "Region of interest, location of the bottom
border. Measured from top of the image." ;
        int qc_roi_bot_position(time, camera) ;
                qc_roi_bot_position:long_name = "Quality check results on field:
Location of region of interest\'s bottom border" ;
                qc_roi_bot_position:units = "unitless" ;
                qc_roi_bot_position:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_roi_bot_position:flag_method = "bit" ;
                qc_roi_bot_position:bit_1_description = "snowflake_fall_speed is set
to bad value. Value set to missing_value" ;
                qc_roi_bot_position:bit_1_assessment = "Bad" ;
                qc_roi_bot_position:bit_2_description = "camera_id for this camera set
to missing_value. Value set to missing_value" ;
                qc_roi_bot_position:bit_2_assessment = "Bad" ;
                qc_roi_bot_position:bit_3_description = "Image file for this camera is
missing (can\'t be opened). Value set to missing_value" ;
                qc_roi_bot_position:bit_3_assessment = "Bad" ;
                qc_roi_bot_position:bit_4_description = "No particle detected in
image. Value set to missing_value" ;
                qc_roi_bot_position:bit_4_assessment = "Bad" ;
                qc_roi_bot_position:bit_5_description = "A property of the particle
most in focus is less than warn_min or greater than warn_max limit." ;
                qc_roi_bot_position:bit_5_assessment = "Indeterminate" ;
                qc_roi_bot_position:bit_6_description = "Value is less than the
warn_min" ;
                qc_roi_bot_position:bit_6_assessment = "Indeterminate" ;
                qc_roi_bot_position:bit_7_description = "Value is greater than the
warn_max" ;
                qc_roi_bot_position:bit_7_assessment = "Indeterminate" ;
        float roi_half_width_height(time, camera, num_elems_roi_position) ;
                roi_half_width_height:long_name = "Half width and height of region of
interest" ;
                roi_half_width_height:units = "mm" ;
                roi_half_width_height:ancillary_variables = "qc_roi_half_width_height"
;
                roi_half_width_height:missing_value = -9999.f ;
                roi_half_width_height:cell_methods = "time: point" ;
        int qc_roi_half_width_height(time, camera, num_elems_roi_position) ;
                qc_roi_half_width_height:long_name = "Quality check results on field:
Half width and height of region of interest" ;
                qc_roi_half_width_height:units = "unitless" ;
                qc_roi_half_width_height:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_roi_half_width_height:flag_method = "bit" ;
                qc_roi_half_width_height:bit_1_description = "snowflake_fall_speed is
set to bad value. Value set to missing_value" ;
                qc_roi_half_width_height:bit_1_assessment = "Bad" ;
                qc_roi_half_width_height:bit_2_description = "camera_id for this
camera set to missing_value. Value set to missing_value" ;
                qc_roi_half_width_height:bit_2_assessment = "Bad" ;
                qc_roi_half_width_height:bit_3_description = "Image file for this
camera is missing (can\'t be opened). Value set to missing_value" ;
                qc_roi_half_width_height:bit_3_assessment = "Bad" ;
                qc_roi_half_width_height:bit_4_description = "No particle detected in
image. Value set to missing_value" ;
                qc_roi_half_width_height:bit_4_assessment = "Bad" ;
                qc_roi_half_width_height:bit_5_description = "A property of the
particle most in focus is less than warn_min or greater than warn_max limit." ;
                qc_roi_half_width_height:bit_5_assessment = "Indeterminate" ;
        int rain(time, camera) ;
                rain:long_name = "Hydrometeor is a rain drop" ;
                rain:units = "unitless" ;
                rain:description = "This field estimates the rain value.  becomes 1 if
hydrometeor meets any of the above conditions. Rain value of 1 means the hydrometeor
```

```
is most likely a rain drop, 0 means not a rain drop, and NA mans not enough info are
available to identify the hydrometer type." ;
        int num_imgs_used_avg(time) ;
                num_imgs_used_avg:long_name = "Number of images for averages" ;
                num_imgs_used_avg:units = "unitless" ;
                num_imgs_used_avg:ancillary_variables = "qc_num_imgs_used_avg" ;
                num_imgs_used_avg:valid_min = 1 ;
                num_imgs_used_avg:warn_min = 2 ;
                num_imgs_used_avg:missing_value = -9999 ;
                num_imgs_used_avg:cell_methods = "time: point" ;
                num_imgs_used_avg:comment = "Number of images used to compute per-
snowflake averages (all variables named *_avg). Only images that pass quality checks
are considered" ;
        int qc_num_imgs_used_avg(time) ;
                qc_num_imgs_used_avg:long_name = "Quality check results on field:
Number of images for averages" ;
                qc_num_imgs_used_avg:units = "unitless" ;
                qc_num_imgs_used_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_num_imgs_used_avg:flag_method = "bit" ;
                qc_num_imgs_used_avg:bit_1_description = "snowflake_fall_speed is set
to bad value. Value set to missing_value" ;
                qc_num_imgs_used_avg:bit_1_assessment = "Bad" ;
                qc_num_imgs_used_avg:bit_2_description = "All camera_ids for this
particle are set to missing_value. Value set to missing_value" ;
                qc_num_imgs_used_avg:bit_2_assessment = "Bad" ;
                qc_num_imgs_used_avg:bit_3_description = "num_objects set to
missing_value for all camera_ids due to property of the particle most in focus being
less than warn_min or greater than warn_max limits. Value set to missing_value" ;
                qc_num_imgs_used_avg:bit_3_assessment = "Bad" ;
                qc_num_imgs_used_avg:bit_4_description = "Value is less than the
valid_min" ;
                qc_num_imgs_used_avg:bit_4_assessment = "Bad" ;
                qc_num_imgs_used_avg:bit_5_description = "Value is less than warn_min.
Could indicate not enough data for an average" ;
                qc_num_imgs_used_avg:bit_5_assessment = "Indeterminate" ;
        float maximum_dimension_avg(time) ;
                maximum_dimension_avg:long_name = "Average of maximum_dimension" ;
                maximum_dimension_avg:units = "mm" ;
                maximum_dimension_avg:ancillary_variables = "qc_maximum_dimension_avg"
;
                maximum_dimension_avg:missing_value = -9999.f ;
                maximum_dimension_avg:cell_methods = "camera: mean" ;
                maximum_dimension_avg:comment = "Average of maximum_dimension value
for all images which passed quality check." ;
        int qc_maximum_dimension_avg(time) ;
                qc_maximum_dimension_avg:long_name = "Quality check results on field:
Average of maximum_dimension" ;
                qc_maximum_dimension_avg:units = "unitless" ;
                qc_maximum_dimension_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_maximum_dimension_avg:flag_method = "bit" ;
                qc_maximum_dimension_avg:bit_1_description = "snowflake_fall_speed is
set to bad value. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_1_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_2_description = "All camera_ids for this
particle are set to missing_value. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_2_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_3_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_4_description = "num_imgs_used_avg is
less than valid_min. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_4_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_5_description = "num_imgs_used_avg is
less than warn_min. Could indicate not enough data for an average" ;
                qc_maximum_dimension_avg:bit_5_assessment = "Indeterminate" ;
        float particle_area_avg(time) ;
                particle_area_avg:long_name = "Average of particle_area" ;
                particle_area_avg:units = "mm^2" ;
                particle_area_avg:ancillary_variables = "qc_particle_area_avg" ;
                particle_area_avg:missing_value = -9999.f ;
                particle_area_avg:cell_methods = "camera: mean" ;
                particle_area_avg:comment = "Average of particle_area for all images
which passed quality check" ;
        int qc_particle_area_avg(time) ;
                qc_particle_area_avg:long_name = "Quality check results on field:
Average of particle_area" ;
                qc_particle_area_avg:units = "unitless" ;
                qc_particle_area_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_particle_area_avg:flag_method = "bit" ;
                qc_particle_area_avg:bit_1_description = "snowflake_fall_speed is set
to bad value. Value set to missing_value" ;
                qc_particle_area_avg:bit_1_assessment = "Bad" ;
                qc_particle_area_avg:bit_2_description = "All camera_ids for this
particle are set to missing_value. Value set to missing_value" ;
                qc_particle_area_avg:bit_2_assessment = "Bad" ;
                qc_particle_area_avg:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_particle_area_avg:bit_3_assessment = "Bad" ;
                qc_particle_area_avg:bit_4_description = "num_imgs_used_avg is less
than valid_min. Value set to missing_value" ;
                qc_particle_area_avg:bit_4_assessment = "Bad" ;
                qc_particle_area_avg:bit_5_description = "num_imgs_used_avg is less
than warn_min. Could indicate not enough data for an average" ;
                qc_particle_area_avg:bit_5_assessment = "Indeterminate" ;
        float area_eq_radius_avg(time) ;
                area_eq_radius_avg:long_name = "Average of area_eq_radius" ;
                area_eq_radius_avg:units = "mm" ;
                area_eq_radius_avg:ancillary_variables = "qc_area_eq_radius_avg" ;
                area_eq_radius_avg:missing_value = -9999.f ;
                area_eq_radius_avg:cell_methods = "camera: mean" ;
                area_eq_radius_avg:comment = "Average of area_eq_radius value for all
images which passed quality check" ;
        int qc_area_eq_radius_avg(time) ;
                qc_area_eq_radius_avg:long_name = "Quality check results on field:
Average of area_eq_radius" ;
                qc_area_eq_radius_avg:units = "unitless" ;
                qc_area_eq_radius_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_area_eq_radius_avg:flag_method = "bit" ;
                qc_area_eq_radius_avg:bit_1_description = "snowflake_fall_speed is set
to bad value. Value set to missing_value" ;
                qc_area_eq_radius_avg:bit_1_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_2_description = "All camera_ids for this
particle are set to missing_value. Value set to missing_value" ;
```

```
                qc_area_eq_radius_avg:bit_2_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_area_eq_radius_avg:bit_3_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_4_description = "num_imgs_used_avg is less
than valid_min. Value set to missing_value" ;
                qc_area_eq_radius_avg:bit_4_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_5_description = "num_imgs_used_avg is less
than warn_min. Could indicate not enough data for an average" ;
                qc_area_eq_radius_avg:bit_5_assessment = "Indeterminate" ;
        float perimeter_avg(time) ;
                perimeter_avg:long_name = "Average of perimeter" ;
                perimeter_avg:units = "mm" ;
                perimeter_avg:ancillary_variables = "qc_perimeter_avg" ;
                perimeter_avg:missing_value = -9999.f ;
                perimeter_avg:cell_methods = "camera: mean" ;
                perimeter_avg:comment = "Average of perimeter value for all images
which passed quality check" ;
        int qc_perimeter_avg(time) ;
                qc_perimeter_avg:long_name = "Quality check results on field: Average
of perimeter" ;
                qc_perimeter_avg:units = "unitless" ;
                qc_perimeter_avg:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_perimeter_avg:flag_method = "bit" ;
                qc_perimeter_avg:bit_1_description = "snowflake_fall_speed is set to
bad value. Value set to missing_value" ;
                qc_perimeter_avg:bit_1_assessment = "Bad" ;
                qc_perimeter_avg:bit_2_description = "All camera_ids for this particle
are set to missing_value. Value set to missing_value" ;
                qc_perimeter_avg:bit_2_assessment = "Bad" ;
                qc_perimeter_avg:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_perimeter_avg:bit_3_assessment = "Bad" ;
                qc_perimeter_avg:bit_4_description = "num_imgs_used_avg is less than
valid_min. Value set to missing_value" ;
                qc_perimeter_avg:bit_4_assessment = "Bad" ;
                qc_perimeter_avg:bit_5_description = "num_imgs_used_avg is less than
warn_min. Could indicate not enough data for an average" ;
                qc_perimeter_avg:bit_5_assessment = "Indeterminate" ;
        float orientation_avg(time) ;
                orientation_avg:long_name = "Average of orientation" ;
                orientation_avg:units = "degree" ;
                orientation_avg:ancillary_variables = "qc_orientation_avg" ;
                orientation_avg:missing_value = -9999.f ;
                orientation_avg:cell_methods = "camera: mean" ;
                orientation_avg:comment = "Average of orientation value for all images
which passed quality check" ;
        int qc_orientation_avg(time) ;
                qc_orientation_avg:long_name = "Quality check results on field:
Average of orientation" ;
                qc_orientation_avg:units = "unitless" ;
                qc_orientation_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_orientation_avg:flag_method = "bit" ;
                qc_orientation_avg:bit_1_description = "snowflake_fall_speed is set to
bad value. Value set to missing_value" ;
                qc_orientation_avg:bit_1_assessment = "Bad" ;
                qc_orientation_avg:bit_2_description = "All camera_ids for this 
particle are set to missing_value. Value set to missing_value" ;
                qc_orientation_avg:bit_2_assessment = "Bad" ;
                qc_orientation_avg:bit_3_description = "num_imgs_used_avg set to 
missing_value. Value set to missing_value" ;
                qc_orientation_avg:bit_3_assessment = "Bad" ;
                qc_orientation_avg:bit_4_description = "num_imgs_used_avg is less than 
valid_min. Value set to missing_value" ;
                qc_orientation_avg:bit_4_assessment = "Bad" ;
                qc_orientation_avg:bit_5_description = "num_imgs_used_avg is less than 
warn_min. Could indicate not enough data for an average" ;
                qc_orientation_avg:bit_5_assessment = "Indeterminate" ;
        float aspect_ratio_avg(time) ;
                aspect_ratio_avg:long_name = "Average of aspect_ratio" ;
                aspect_ratio_avg:units = "unitless" ;
                aspect_ratio_avg:ancillary_variables = "qc_aspect_ratio_avg" ;
                aspect_ratio_avg:missing_value = -9999.f ;
                aspect_ratio_avg:cell_methods = "camera: mean" ;
                aspect_ratio_avg:comment = "Average of aspect_ratio value for all 
images which passed quality check" ;
        int qc_aspect_ratio_avg(time) ;
                qc_aspect_ratio_avg:long_name = "Quality check results on field: 
Average of aspect_ratio" ;
                qc_aspect_ratio_avg:units = "unitless" ;
                qc_aspect_ratio_avg:description = "This field contains bit packed 
integer values, where each bit represents a QC test on the data. Non-zero bits 
indicate the QC condition given in the description for those bits; a value of 0 (no 
bits set) indicates the data has not failed any QC tests." ;
                qc_aspect_ratio_avg:flag_method = "bit" ;
                qc_aspect_ratio_avg:bit_1_description = "snowflake_fall_speed is set 
to bad value. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_1_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_2_description = "All camera_ids for this 
particle are set to missing_value. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_2_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_3_description = "num_imgs_used_avg set to 
missing_value. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_3_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_4_description = "num_imgs_used_avg is less 
than valid_min. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_4_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_5_description = "num_imgs_used_avg is less 
than warn_min. Could indicate not enough data for an average" ;
                qc_aspect_ratio_avg:bit_5_assessment = "Indeterminate" ;
        float complexity_avg(time) ;
                complexity_avg:long_name = "Average of complexity" ;
                complexity_avg:units = "unitless" ;
                complexity_avg:ancillary_variables = "qc_complexity_avg" ;
                complexity_avg:missing_value = -9999.f ;
                complexity_avg:cell_methods = "camera: mean" ;
                complexity_avg:comment = "Average of complexity value for all images 
which passed quality check" ;
        int qc_complexity_avg(time) ;
                qc_complexity_avg:long_name = "Quality check results on field: Average 
of complexity" ;
                qc_complexity_avg:units = "unitless" ;
                qc_complexity_avg:description = "This field contains bit packed 
integer values, where each bit represents a QC test on the data. Non-zero bits 
indicate the QC condition given in the description for those bits; a value of 0 (no 
bits set) indicates the data has not failed any QC tests." ;
                qc_complexity_avg:flag_method = "bit" ;
                qc_complexity_avg:bit_1_description = "snowflake_fall_speed is set to 
bad value. Value set to missing_value" ;
```

```
                qc_complexity_avg:bit_1_assessment = "Bad" ;
                qc_complexity_avg:bit_2_description = "All camera_ids for this
particle are set to missing_value. Value set to missing_value" ;
                qc_complexity_avg:bit_2_assessment = "Bad" ;
                qc_complexity_avg:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_complexity_avg:bit_3_assessment = "Bad" ;
                qc_complexity_avg:bit_4_description = "num_imgs_used_avg is less than
valid_min. Value set to missing_value" ;
                qc_complexity_avg:bit_4_assessment = "Bad" ;
                qc_complexity_avg:bit_5_description = "num_imgs_used_avg is less than
warn_min. Could indicate not enough data for an average" ;
                qc_complexity_avg:bit_5_assessment = "Indeterminate" ;
        float geometric_cross_section_avg(time) ;
                geometric_cross_section_avg:long_name = "Average of
geometric_cross_section" ;
                geometric_cross_section_avg:units = "mm^2" ;
                geometric_cross_section_avg:ancillary_variables =
"qc_geometric_cross_section_avg" ;
                geometric_cross_section_avg:missing_value = -9999.f ;
                geometric_cross_section_avg:cell_methods = "camera: mean" ;
                geometric_cross_section_avg:comment = "Average of
geometric_cross_section for all images which passed quality check" ;
        int qc_geometric_cross_section_avg(time) ;
                qc_geometric_cross_section_avg:long_name = "Quality check results on
field: Average of geometric_cross_section" ;
                qc_geometric_cross_section_avg:units = "unitless" ;
                qc_geometric_cross_section_avg:description = "This field contains bit
packed integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_geometric_cross_section_avg:flag_method = "bit" ;
                qc_geometric_cross_section_avg:bit_1_description =
"snowflake_fall_speed is set to bad value. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_1_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_2_description = "All camera_ids for
this particle are set to missing_value. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_2_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_3_description = "num_imgs_used_avg
set to missing_value. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_3_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_4_description = "num_imgs_used_avg
is less than valid_min. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_4_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_5_description = "num_imgs_used_avg
is less than warn_min. Could indicate not enough data for an average" ;
                qc_geometric_cross_section_avg:bit_5_assessment = "Indeterminate" ;
        float mean_pixel_intensity_avg(time) ;
                mean_pixel_intensity_avg:long_name = "Average of mean_pixel_intensity"
;
                mean_pixel_intensity_avg:units = "unitless" ;
                mean_pixel_intensity_avg:ancillary_variables =
"qc_mean_pixel_intensity_avg" ;
                mean_pixel_intensity_avg:missing_value = -9999.f ;
                mean_pixel_intensity_avg:cell_methods = "camera: mean" ;
                mean_pixel_intensity_avg:comment = "Average of mean_pixel_intensity
value for all images which passed quality check" ;
        int qc_mean_pixel_intensity_avg(time) ;
                qc_mean_pixel_intensity_avg:long_name = "Quality check results on
field: Average of mean_pixel_intensity" ;
                qc_mean_pixel_intensity_avg:units = "unitless" ;
                qc_mean_pixel_intensity_avg:description = "This field contains bit
packed integer values, where each bit represents a QC test on the data. Non-zero bits
```

```
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity_avg:flag_method = "bit" ;
                qc_mean_pixel_intensity_avg:bit_1_description = "snowflake_fall_speed
is set to bad value. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_2_description = "All camera_ids for
this particle are set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_3_description = "num_imgs_used_avg set
to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_3_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_4_description = "num_imgs_used_avg is
less than valid_min. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_4_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_5_description = "num_imgs_used_avg is
less than warn_min. Could indicate not enough data for an average" ;
                qc_mean_pixel_intensity_avg:bit_5_assessment = "Indeterminate" ;
        float mean_pixel_intensity_variability_avg(time) ;
                mean_pixel_intensity_variability_avg:long_name = "Average of
mean_pixel_intensity_variability" ;
                mean_pixel_intensity_variability_avg:units = "unitless" ;
                mean_pixel_intensity_variability_avg:ancillary_variables =
"qc_mean_pixel_intensity_variability_avg" ;
                mean_pixel_intensity_variability_avg:missing_value = -9999.f ;
                mean_pixel_intensity_variability_avg:cell_methods = "camera: mean" ;
                mean_pixel_intensity_variability_avg:comment = "Average of
mean_pixel_intensity_variability value for all images which passed quality check" ;
        int qc_mean_pixel_intensity_variability_avg(time) ;
                qc_mean_pixel_intensity_variability_avg:long_name = "Quality check
results on field: Average of mean_pixel_intensity_variability" ;
                qc_mean_pixel_intensity_variability_avg:units = "unitless" ;
                qc_mean_pixel_intensity_variability_avg:description = "This field
contains bit packed integer values, where each bit represents a QC test on the data.
Non-zero bits indicate the QC condition given in the description for those bits; a
value of 0 (no bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity_variability_avg:flag_method = "bit" ;
                qc_mean_pixel_intensity_variability_avg:bit_1_description =
"snowflake_fall_speed is set to bad value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_2_description = "All
camera_ids for this particle are set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_3_description =
"num_imgs_used_avg set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_3_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_4_description =
"num_imgs_used_avg is less than valid_min. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_4_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_5_description =
"num_imgs_used_avg is less than warn_min. Could indicate not enough data for an
average" ;
                qc_mean_pixel_intensity_variability_avg:bit_5_assessment =
"Indeterminate" ;
        float flatness(time) ;
                flatness:long_name = "Estimate of particle flatness" ;
                flatness:units = "unitless" ;
                flatness:ancillary_variables = "qc_flatness" ;
                flatness:missing_value = -9999.f ;
                flatness:cell_methods = "time: point" ;
                flatness:comment = "Measures particle flatness based on aspect ratios
of individual images. Computed as: abs( (max(aspect_ratio) - min(aspect_ratio)) /
average(aspect_ratio) ). Set to missing value when ave_num_imgs_used <= 1" ;
```

```
        int qc_flatness(time) ;
                qc_flatness:long_name = "Quality check results on field: Estimate of
particle flatness" ;
                qc_flatness:units = "unitless" ;
                qc_flatness:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_flatness:flag_method = "bit" ;
                qc_flatness:bit_1_description = "snowflake_fall_speed is set to bad
value. Value set to missing_value" ;
                qc_flatness:bit_1_assessment = "Bad" ;
                qc_flatness:bit_2_description = "All camera_ids for this particle are
set to missing_value. Value set to missing_value" ;
                qc_flatness:bit_2_assessment = "Bad" ;
                qc_flatness:bit_3_description = "num_imgs_used_avg set to
missing_value. Value set to missing_value" ;
                qc_flatness:bit_3_assessment = "Bad" ;
                qc_flatness:bit_4_description = "num_imgs_used_avg is less than
valid_min. Value set to missing_value" ;
                qc_flatness:bit_4_assessment = "Bad" ;
                qc_flatness:bit_5_description = "num_imgs_used_avg is less than
warn_min. Could indicate not enough data for an average. Value set to missing_value" ;
                qc_flatness:bit_5_assessment = "Bad" ;
        float lat ;
                lat:long_name = "North latitude" ;
                lat:units = "degree_N" ;
                lat:valid_min = -90.f ;
                lat:valid_max = 90.f ;
                lat:standard_name = "latitude" ;
        float lon ;
                lon:long_name = "East longitude" ;
                lon:units = "degree_E" ;
                lon:valid_min = -180.f ;
                lon:valid_max = 180.f ;
                lon:standard_name = "longitude" ;
        float alt ;
                alt:long_name = "Altitude above mean sea level" ;
                alt:units = "m" ;
                alt:standard_name = "altitude" ;

// global attributes:
                :command_line = "masc_flake_anal_vap.py -s oli -f M1 -b 20151025 -e
20160623 -D 2 -R" ;
                :Conventions = "ARM-1.1" ;
                :process_version = "$" ;
                :anal_config_json = "{\"timeBinningParameters\":
{\"maxFallSpeedInMetersPS\": 5.0, \"binWidthInSec\": 300.0, \"minNumParticlesPerBin\":
10.0}, \"imageAnalysisParameters\": {\"focusThreshold01\": 0.009999999776482582,
\"backgroundThreshold01\": 0.03, \"minFlakeSizeInMicrons\": 199.9999977648258,
\"flagSaveCroppedImages\": false, \"maxEdgeTouchLengthInMicrons\": 500.0,
\"flagRejectOutOfFocus\": 1, \"additionalImageCrop\": {\"top\": 460, \"right\": 600,
\"bottom\": 360, \"left\": 600}, \"rangeIntensityThreshold01\": 0.01899999938905239,
\"lineFillInMicrons\": 200, \"minMaxPixelIntensity01\": 0.2000000298023224,
\"boundingBoxThresholdInMM\": {\"bottomMax\": 36.0, \"bottomMin\": 32.0}}}" ;
                :dod_version = "mascparticles-c1-1.2" ;
                :input_datastreams = "olimascM1.b1 : 1.1 : 20160301.065053" ;
                :site_id = "oli" ;
                :platform_id = "mascparticles" ;
                :facility_id = "M1" ;
                :data_level = "c1" ;
                :location_description = "North Slope of Alaska (NSA), Oliktok Point,
Alaska" ;
```

```
                :datastream = "olimascparticlesM1.c1" ;
                :doi = "10.5439/1239672" ;
                :masc_version = "v3.0: 3x 5MP Unibrain cameras, 12mm lenses" ;
                :history = "created by user shkurko on machine research at 2016-08-19
02:40:18, using $" ;
}
```

Appendix E

NetCDF Header for mascparticlesavgM1.c1

An example header from the MASC_FLAKE_ANAL VAP is given below:

```
netcdf olimascparticlesavgM1.c1.20151025.052730 {
dimensions:
        time = UNLIMITED ; // (8 currently)
        bound = 2 ;
variables:
        int base_time ;
                base_time:string = "2015-10-25 00:00:00 0:00" ;
                base_time:long_name = "Base time in Epoch" ;
                base_time:units = "seconds since 1970-1-1 0:00:00 0:00" ;
                base_time:ancillary_variables = "time_offset" ;
        double time_offset(time) ;
                time_offset:long_name = "Time offset from base_time" ;
                time_offset:units = "seconds since 2015-10-25 00:00:00 0:00" ;
                time_offset:ancillary_variables = "base_time" ;
        double time(time) ;
                time:long_name = "Time offset from midnight" ;
                time:units = "seconds since 2015-10-25 00:00:00 0:00" ;
                time:bounds = "time_bounds" ;
                time:calendar = "gregorian" ;
                time:standard_name = "time" ;
        double time_bounds(time, bound) ;
                time_bounds:long_name = "Time cell bounds" ;
                time_bounds:bound_offsets = -150., 150. ;
        float num_particles_total(time) ;
                num_particles_total:long_name = "Total number of particles that fell
into this bin." ;
                num_particles_total:units = "unitless" ;
                num_particles_total:ancillary_variables = "qc_num_particles_total" ;
                num_particles_total:missing_value = -9999.f ;
                num_particles_total:cell_methods = "time: sum" ;
                num_particles_total:comment = "Only a subset (counted in
num_particles_for_avg) will be used to compute averages." ;
        int qc_num_particles_total(time) ;
                qc_num_particles_total:long_name = "Quality check results on field:
Total number of particles that fell into this bin." ;
                qc_num_particles_total:units = "unitless" ;
                qc_num_particles_total:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
```

```
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_num_particles_total:flag_method = "bit" ;
                qc_num_particles_total:bit_1_description = "Value set to
missing_value" ;
                qc_num_particles_total:bit_1_assessment = "Bad" ;
                qc_num_particles_total:bit_2_description = "Value is equal to 0" ;
                qc_num_particles_total:bit_2_assessment = "Indeterminate" ;
        int num_particles_for_avg(time) ;
                num_particles_for_avg:long_name = "Number of particles used to
average" ;
                num_particles_for_avg:units = "unitless" ;
                num_particles_for_avg:ancillary_variables = "qc_num_particles_for_avg"
;
                num_particles_for_avg:warn_min = 10 ;
                num_particles_for_avg:missing_value = -9999 ;
                num_particles_for_avg:cell_methods = "time: sum" ;
                num_particles_for_avg:comment = "Number of particles (that fall within
the time bin) used to average. Only particles passing quality control contribute and
the value of num_objects for at least 2 of 3 images must be same and equal to 1.
Depends on time_bin_width. Limit here set based on 5min bin width" ;
        int qc_num_particles_for_avg(time) ;
                qc_num_particles_for_avg:long_name = "Quality check results on field:
Number of particles used to average" ;
                qc_num_particles_for_avg:units = "unitless" ;
                qc_num_particles_for_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_num_particles_for_avg:flag_method = "bit" ;
                qc_num_particles_for_avg:bit_1_description = "Value set to
missing_value" ;
                qc_num_particles_for_avg:bit_1_assessment = "Bad" ;
                qc_num_particles_for_avg:bit_2_description = "Value is less than the
warn_min" ;
                qc_num_particles_for_avg:bit_2_assessment = "Indeterminate" ;
                qc_num_particles_for_avg:bit_3_description = "Value is equal to 0" ;
                qc_num_particles_for_avg:bit_3_assessment = "Indeterminate" ;
        float fall_speed_avg(time) ;
                fall_speed_avg:long_name = "Average fallspeed" ;
                fall_speed_avg:units = "m/s" ;
                fall_speed_avg:ancillary_variables = "qc_fall_speed_avg" ;
                fall_speed_avg:missing_value = -9999.f ;
                fall_speed_avg:cell_methods = "time: mean" ;
                fall_speed_avg:comment = "Average of fallspeeds for all valid
particles within time bin" ;
        int qc_fall_speed_avg(time) ;
                qc_fall_speed_avg:long_name = "Quality check results on field: Average
fallspeed" ;
                qc_fall_speed_avg:units = "unitless" ;
                qc_fall_speed_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_fall_speed_avg:flag_method = "bit" ;
                qc_fall_speed_avg:bit_1_description = "Value set to missing_value" ;
                qc_fall_speed_avg:bit_1_assessment = "Bad" ;
                qc_fall_speed_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_fall_speed_avg:bit_2_assessment = "Bad" ;
                qc_fall_speed_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_fall_speed_avg:bit_3_assessment = "Indeterminate" ;
                qc_fall_speed_avg:bit_4_description = "num_particles_for_avg is equal
to 0. Value set to missing_value" ;
                qc_fall_speed_avg:bit_4_assessment = "Bad" ;
        float maximum_dimension_avg(time) ;
                maximum_dimension_avg:long_name = "Average maximum dimension" ;
                maximum_dimension_avg:units = "mm" ;
                maximum_dimension_avg:ancillary_variables = "qc_maximum_dimension_avg"
;
                maximum_dimension_avg:missing_value = -9999.f ;
                maximum_dimension_avg:cell_methods = "time: mean" ;
                maximum_dimension_avg:comment = "Average maximum dimension for all
particles within this time bin" ;
        int qc_maximum_dimension_avg(time) ;
                qc_maximum_dimension_avg:long_name = "Quality check results on field:
Average maximum dimension" ;
                qc_maximum_dimension_avg:units = "unitless" ;
                qc_maximum_dimension_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_maximum_dimension_avg:flag_method = "bit" ;
                qc_maximum_dimension_avg:bit_1_description = "Value set to
missing_value" ;
                qc_maximum_dimension_avg:bit_1_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_2_description = "num_particles_for_avg
set to missing_value. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_2_assessment = "Bad" ;
                qc_maximum_dimension_avg:bit_3_description = "num_particles_for_avg is
less than warn_min. Could indicate lack of samples for statistically-significant
results" ;
                qc_maximum_dimension_avg:bit_3_assessment = "Indeterminate" ;
                qc_maximum_dimension_avg:bit_4_description = "num_particles_for_avg is
equal to 0. Value set to missing_value" ;
                qc_maximum_dimension_avg:bit_4_assessment = "Bad" ;
        float particle_area_avg(time) ;
                particle_area_avg:long_name = "Average particle area" ;
                particle_area_avg:units = "mm^2" ;
                particle_area_avg:ancillary_variables = "qc_particle_area_avg" ;
                particle_area_avg:missing_value = -9999.f ;
                particle_area_avg:cell_methods = "time: mean" ;
                particle_area_avg:comment = "Average particle area for all particles
within this time bin" ;
        int qc_particle_area_avg(time) ;
                qc_particle_area_avg:long_name = "Quality check results on field:
Average particle area" ;
                qc_particle_area_avg:units = "unitless" ;
                qc_particle_area_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_particle_area_avg:flag_method = "bit" ;
                qc_particle_area_avg:bit_1_description = "Value set to missing_value"
;
                qc_particle_area_avg:bit_1_assessment = "Bad" ;
                qc_particle_area_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_particle_area_avg:bit_2_assessment = "Bad" ;
                qc_particle_area_avg:bit_3_description = "num_particles_for_avg is
less than warn_min. Could indicate lack of samples for statistically-significant
results" ;
                qc_particle_area_avg:bit_3_assessment = "Indeterminate" ;
                qc_particle_area_avg:bit_4_description = "num_particles_for_avg is
equal to 0. Value set to missing_value" ;
                qc_particle_area_avg:bit_4_assessment = "Bad" ;
        float area_eq_radius_avg(time) ;
                area_eq_radius_avg:long_name = "Average area equivalent radius" ;
                area_eq_radius_avg:units = "mm" ;
                area_eq_radius_avg:ancillary_variables = "qc_area_eq_radius_avg" ;
                area_eq_radius_avg:missing_value = -9999.f ;
                area_eq_radius_avg:cell_methods = "time: mean" ;
                area_eq_radius_avg:comment = "Average area equivalent radius for all
particles within this time bin" ;
        int qc_area_eq_radius_avg(time) ;
                qc_area_eq_radius_avg:long_name = "Quality check results on field:
Average area equivalent radius" ;
                qc_area_eq_radius_avg:units = "unitless" ;
                qc_area_eq_radius_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_area_eq_radius_avg:flag_method = "bit" ;
                qc_area_eq_radius_avg:bit_1_description = "Value set to missing_value"
;
                qc_area_eq_radius_avg:bit_1_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_2_description = "num_particles_for_avg set
to missing_value. Value set to missing_value" ;
                qc_area_eq_radius_avg:bit_2_assessment = "Bad" ;
                qc_area_eq_radius_avg:bit_3_description = "num_particles_for_avg is
less than warn_min. Could indicate lack of samples for statistically-significant
results" ;
                qc_area_eq_radius_avg:bit_3_assessment = "Indeterminate" ;
                qc_area_eq_radius_avg:bit_4_description = "num_particles_for_avg is
equal to 0. Value set to missing_value" ;
                qc_area_eq_radius_avg:bit_4_assessment = "Bad" ;
        float perimeter_avg(time) ;
                perimeter_avg:long_name = "Average perimeter" ;
                perimeter_avg:units = "mm" ;
                perimeter_avg:ancillary_variables = "qc_perimeter_avg" ;
                perimeter_avg:missing_value = -9999.f ;
                perimeter_avg:cell_methods = "time: mean" ;
                perimeter_avg:comment = "Average perimeter for all particles within
this time bin" ;
        int qc_perimeter_avg(time) ;
                qc_perimeter_avg:long_name = "Quality check results on field: Average
perimeter" ;
                qc_perimeter_avg:units = "unitless" ;
                qc_perimeter_avg:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
```

```
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_perimeter_avg:flag_method = "bit" ;
                qc_perimeter_avg:bit_1_description = "Value set to missing_value" ;
                qc_perimeter_avg:bit_1_assessment = "Bad" ;
                qc_perimeter_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_perimeter_avg:bit_2_assessment = "Bad" ;
                qc_perimeter_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_perimeter_avg:bit_3_assessment = "Indeterminate" ;
                qc_perimeter_avg:bit_4_description = "num_particles_for_avg is equal
to 0. Value set to missing_value" ;
                qc_perimeter_avg:bit_4_assessment = "Bad" ;
        float orientation_avg(time) ;
                orientation_avg:long_name = "Average orientation" ;
                orientation_avg:units = "degree" ;
                orientation_avg:ancillary_variables = "qc_orientation_avg" ;
                orientation_avg:missing_value = -9999.f ;
                orientation_avg:cell_methods = "time: mean" ;
                orientation_avg:comment = "Average orientation for all particles
within this time bin" ;
        int qc_orientation_avg(time) ;
                qc_orientation_avg:long_name = "Quality check results on field:
Average orientation" ;
                qc_orientation_avg:units = "unitless" ;
                qc_orientation_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_orientation_avg:flag_method = "bit" ;
                qc_orientation_avg:bit_1_description = "Value set to missing_value" ;
                qc_orientation_avg:bit_1_assessment = "Bad" ;
                qc_orientation_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_orientation_avg:bit_2_assessment = "Bad" ;
                qc_orientation_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_orientation_avg:bit_3_assessment = "Indeterminate" ;
                qc_orientation_avg:bit_4_description = "num_particles_for_avg is equal
to 0. Value set to missing_value" ;
                qc_orientation_avg:bit_4_assessment = "Bad" ;
        float aspect_ratio_avg(time) ;
                aspect_ratio_avg:long_name = "Average aspect ratio" ;
                aspect_ratio_avg:units = "unitless" ;
                aspect_ratio_avg:ancillary_variables = "qc_aspect_ratio_avg" ;
                aspect_ratio_avg:missing_value = -9999.f ;
                aspect_ratio_avg:cell_methods = "time: mean" ;
                aspect_ratio_avg:comment = "Average aspect ratio for all particles
within this time bin" ;
        int qc_aspect_ratio_avg(time) ;
                qc_aspect_ratio_avg:long_name = "Quality check results on field:
Average aspect ratio" ;
                qc_aspect_ratio_avg:units = "unitless" ;
                qc_aspect_ratio_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
```

```
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_aspect_ratio_avg:flag_method = "bit" ;
                qc_aspect_ratio_avg:bit_1_description = "Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_1_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_2_assessment = "Bad" ;
                qc_aspect_ratio_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_aspect_ratio_avg:bit_3_assessment = "Indeterminate" ;
                qc_aspect_ratio_avg:bit_4_description = "num_particles_for_avg is
equal to 0. Value set to missing_value" ;
                qc_aspect_ratio_avg:bit_4_assessment = "Bad" ;
        float complexity_avg(time) ;
                complexity_avg:long_name = "Average complexity" ;
                complexity_avg:units = "unitless" ;
                complexity_avg:ancillary_variables = "qc_complexity_avg" ;
                complexity_avg:missing_value = -9999.f ;
                complexity_avg:cell_methods = "time: mean" ;
                complexity_avg:comment = "Average complexity for all particles within
this time bin" ;
        int qc_complexity_avg(time) ;
                qc_complexity_avg:long_name = "Quality check results on field: Average
complexity" ;
                qc_complexity_avg:units = "unitless" ;
                qc_complexity_avg:description = "This field contains bit packed
integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_complexity_avg:flag_method = "bit" ;
                qc_complexity_avg:bit_1_description = "Value set to missing_value" ;
                qc_complexity_avg:bit_1_assessment = "Bad" ;
                qc_complexity_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_complexity_avg:bit_2_assessment = "Bad" ;
                qc_complexity_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_complexity_avg:bit_3_assessment = "Indeterminate" ;
                qc_complexity_avg:bit_4_description = "num_particles_for_avg is equal
to 0. Value set to missing_value" ;
                qc_complexity_avg:bit_4_assessment = "Bad" ;
        float geometric_cross_section_avg(time) ;
                geometric_cross_section_avg:long_name = "Average geometric cross
section" ;
                geometric_cross_section_avg:units = "mm^2" ;
                geometric_cross_section_avg:ancillary_variables =
"qc_geometric_cross_section_avg" ;
                geometric_cross_section_avg:missing_value = -9999.f ;
                geometric_cross_section_avg:cell_methods = "time: mean" ;
                geometric_cross_section_avg:comment = "Average geometric cross section
for all particles within this time bin" ;
        int qc_geometric_cross_section_avg(time) ;
                qc_geometric_cross_section_avg:long_name = "Quality check results on
field: Average geometric cross section" ;
                qc_geometric_cross_section_avg:units = "unitless" ;
                qc_geometric_cross_section_avg:description = "This field contains bit
packed integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_geometric_cross_section_avg:flag_method = "bit" ;
                qc_geometric_cross_section_avg:bit_1_description = "Value set to
missing_value" ;
                qc_geometric_cross_section_avg:bit_1_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_2_description =
"num_particles_for_avg set to missing_value. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_2_assessment = "Bad" ;
                qc_geometric_cross_section_avg:bit_3_description =
"num_particles_for_avg is less than warn_min. Could indicate lack of samples for
statistically-significant results" ;
                qc_geometric_cross_section_avg:bit_3_assessment = "Indeterminate" ;
                qc_geometric_cross_section_avg:bit_4_description =
"num_particles_for_avg is equal to 0. Value set to missing_value" ;
                qc_geometric_cross_section_avg:bit_4_assessment = "Bad" ;
        float mean_pixel_intensity_avg(time) ;
                mean_pixel_intensity_avg:long_name = "Average mean pixel intensity" ;
                mean_pixel_intensity_avg:units = "unitless" ;
                mean_pixel_intensity_avg:ancillary_variables =
"qc_mean_pixel_intensity_avg" ;
                mean_pixel_intensity_avg:missing_value = -9999.f ;
                mean_pixel_intensity_avg:cell_methods = "time: mean" ;
                mean_pixel_intensity_avg:comment = "Average mean pixel intensity for
all particles within this time bin" ;
        int qc_mean_pixel_intensity_avg(time) ;
                qc_mean_pixel_intensity_avg:long_name = "Quality check results on
field: Average mean pixel intensity" ;
                qc_mean_pixel_intensity_avg:units = "unitless" ;
                qc_mean_pixel_intensity_avg:description = "This field contains bit
packed integer values, where each bit represents a QC test on the data. Non-zero bits
indicate the QC condition given in the description for those bits; a value of 0 (no
bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity_avg:flag_method = "bit" ;
                qc_mean_pixel_intensity_avg:bit_1_description = "Value set to
missing_value" ;
                qc_mean_pixel_intensity_avg:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_2_description = "num_particles_for_avg
set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity_avg:bit_3_description = "num_particles_for_avg
is less than warn_min. Could indicate lack of samples for statistically-significant
results" ;
                qc_mean_pixel_intensity_avg:bit_3_assessment = "Indeterminate" ;
                qc_mean_pixel_intensity_avg:bit_4_description = "num_particles_for_avg
is equal to 0. Value set to missing_value" ;
                qc_mean_pixel_intensity_avg:bit_4_assessment = "Bad" ;
        float mean_pixel_intensity_variability_avg(time) ;
                mean_pixel_intensity_variability_avg:long_name = "Average mean pixel
intensity variability" ;
                mean_pixel_intensity_variability_avg:units = "unitless" ;
                mean_pixel_intensity_variability_avg:ancillary_variables =
"qc_mean_pixel_intensity_variability_avg" ;
                mean_pixel_intensity_variability_avg:missing_value = -9999.f ;
                mean_pixel_intensity_variability_avg:cell_methods = "time: mean" ;
                mean_pixel_intensity_variability_avg:comment = "Average mean pixel
intensity variability for all particles within this time bin" ;
        int qc_mean_pixel_intensity_variability_avg(time) ;
                qc_mean_pixel_intensity_variability_avg:long_name = "Quality check
results on field: Average mean pixel intensity variability" ;
                qc_mean_pixel_intensity_variability_avg:units = "unitless" ;
                qc_mean_pixel_intensity_variability_avg:description = "This field
contains bit packed integer values, where each bit represents a QC test on the data.
Non-zero bits indicate the QC condition given in the description for those bits; a
value of 0 (no bits set) indicates the data has not failed any QC tests." ;
                qc_mean_pixel_intensity_variability_avg:flag_method = "bit" ;
                qc_mean_pixel_intensity_variability_avg:bit_1_description = "Value set
to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_1_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_2_description =
"num_particles_for_avg set to missing_value. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_2_assessment = "Bad" ;
                qc_mean_pixel_intensity_variability_avg:bit_3_description =
"num_particles_for_avg is less than warn_min. Could indicate lack of samples for
statistically-significant results" ;
                qc_mean_pixel_intensity_variability_avg:bit_3_assessment =
"Indeterminate" ;
                qc_mean_pixel_intensity_variability_avg:bit_4_description =
"num_particles_for_avg is equal to 0. Value set to missing_value" ;
                qc_mean_pixel_intensity_variability_avg:bit_4_assessment = "Bad" ;
        float flatness_avg(time) ;
                flatness_avg:long_name = "Average flatness" ;
                flatness_avg:units = "unitless" ;
                flatness_avg:ancillary_variables = "qc_flatness_avg" ;
                flatness_avg:missing_value = -9999.f ;
                flatness_avg:cell_methods = "time: mean" ;
                flatness_avg:comment = "Average flatness for all particles within this
time bin" ;
        int qc_flatness_avg(time) ;
                qc_flatness_avg:long_name = "Quality check results on field: Average
flatness" ;
                qc_flatness_avg:units = "unitless" ;
                qc_flatness_avg:description = "This field contains bit packed integer
values, where each bit represents a QC test on the data. Non-zero bits indicate the QC
condition given in the description for those bits; a value of 0 (no bits set)
indicates the data has not failed any QC tests." ;
                qc_flatness_avg:flag_method = "bit" ;
                qc_flatness_avg:bit_1_description = "Value set to missing_value" ;
                qc_flatness_avg:bit_1_assessment = "Bad" ;
                qc_flatness_avg:bit_2_description = "num_particles_for_avg set to
missing_value. Value set to missing_value" ;
                qc_flatness_avg:bit_2_assessment = "Bad" ;
                qc_flatness_avg:bit_3_description = "num_particles_for_avg is less
than warn_min. Could indicate lack of samples for statistically-significant results" ;
                qc_flatness_avg:bit_3_assessment = "Indeterminate" ;
                qc_flatness_avg:bit_4_description = "num_particles_for_avg is equal to
0. Value set to missing_value" ;
                qc_flatness_avg:bit_4_assessment = "Bad" ;
        float lat ;
                lat:long_name = "North latitude" ;
                lat:units = "degree_N" ;
                lat:valid_min = -90.f ;
                lat:valid_max = 90.f ;
                lat:standard_name = "latitude" ;
        float lon ;
                lon:long_name = "East longitude" ;
                lon:units = "degree_E" ;
                lon:valid_min = -180.f ;
                lon:valid_max = 180.f ;
                lon:standard_name = "longitude" ;
        float alt ;
                alt:long_name = "Altitude above mean sea level" ;
                alt:units = "m" ;
                alt:standard_name = "altitude" ;

// global attributes:
                :command_line = "masc_flake_anal_vap.py -s oli -f M1 -b 20151025 -e 
20160610 -D 2 -R" ;
                :Conventions = "ARM-1.1" ;
                :process_version = "$" ;
                :dod_version = "mascparticlesavg-c1-1.0" ;
                :input_datastreams = "olimascM1.b1 : 1.1 : 20151025.052710" ;
                :site_id = "oli" ;
                :platform_id = "mascparticlesavg" ;
                :facility_id = "M1" ;
                :data_level = "c1" ;
                :location_description = "North Slope of Alaska (NSA), Oliktok Point, 
Alaska" ;
                :datastream = "olimascparticlesavgM1.c1" ;
                :doi = "10.5439/1239673" ;
                :masc_version = "v3.0: 3x 5MP Unibrain cameras, 12mm lenses" ;
                :history = "created by user shkurko on machine research at 2016-06-17 
18:11:32, using $" ;
}
```

■

ARM
CLIMATE RESEARCH FACILITY

www.arm.gov

U.S. DEPARTMENT OF
ENERGY

Office of Science