Documentation for the 'CAMA' verification and validation toolkit

Abstract

The CAMA (Correleer Alles met Alles) toolkit is useful for algorithm developers and validators to perform a quick check on the validity of the output of a level 2 retrieval algorithms. It allows to read OMI swaths from various level 2 files, and correlates all given parameters (hence the name "correlate everything with everything"). This document describes how to use the CAMA toolkit, and lists the known limitations.

		Date	Signature
Author:	Maarten Sneep, Remco Braak, Ellen Brinksma, Mark Kroon	February 22, 2010	
Checked:	Pepijn Veefkind	September 5, 2006	
Approved:	Pieternel Levelt	September 5, 2006	
Archive:	René Noordhoek	September 5, 2006	

Distribution list:

OMI Science Team

PK Bhartia	NASA/GSFC	Joanna Joiner	NASA/GSFC
Bojan Bojkov	NASA/GSFC	Mark Kroon	KNMI
Remco Braak	KNMI	Nick Krotkov	UMBC
Ellen Brinksma	KNMI	Thomas Kurosu	Harvard-Smithsonian CfA
Eric Bucsela	UMBC	Gilbert Leppelmeier	G & S Associates
Edward A. Celarier	NASA/GSFC	Pieternel Levelt	KNMI
Kelly Chance	Harvard-Smithsonian CfA	Anssi Mï¿½kki	FMI
Jacques Claas	KNMI	Renï¿½Noordhoek	KNMI
Ruud Dirksen	KNMI	Bert van den Oord	KNMI
Marcel Dobber	KNMI	Maarten Sneep	KNMI
Larry Flynn	NOAA	Piet Stammes	KNMI
James Gleason	NASA/GSFC	Johanna Tamminen	FMI
Johan de Haan	KNMI	Aapo Tanskanen	FMI
Seppo Hassinen	FMI	Omar Torres	UMBC
Ernest Hilsenrath	NASA/GSFC	Alexandre Vassilkov	SSAI
Mirna van Hoek	KNMI	Pepijn Veefkind	KNMI
Marghi Hopkins	SSAI	Robert Voors	KNMI
Glenn Jaross	SSAI		

Change status:

Issue	Date	Comments	Affected pages
draft	May 19, 2006	First draft	All
1.0	June 15, 2006	First version	All
1.1	July 17, 2006	Added description of new features	All
1.2	August 21, 2006	Added caveat about the regression coefficients	All
1.3	February 22, 2010	Added descriptions for various new flags	All

Contents

1 Introduction

The CAMA toolkit originated in the need to perform visual inspection, statistical analysis and investigation of correlations with measurement geometry and other parameters found in OMI level 2 files. Reading of multiple files merely enhances the statistics. CAMA does not perform validation, but it helps to describe the properties of an OMI level 2 data product. The current version can be used to compare two versions of the same algorithm, to verify that changes have the intended effect, or rather do *not* have unintended side effects.

Following processing, CAMA will create a report with the following items:

- Introductory text, summary of the settings and filter statistics.
- A number of figures with captions, including:
  - Re-gridded geolocated averages
  - Along-track averages and standard deviation. The across track index is zero-based (starting from CAMA released on January 26, 2009).
  - Distribution histograms
  - Equal earth surface probability distribution histograms
  - Scatter density plots of all combinations of parameters
- Tables, including:
  - Covariance and correlation matrices
  - Average, standard deviation, skewness and kurtosis for each parameter
  - Minimum and maximum for each parameter
  - Linear regression coefficients
  - Pixel counts for all selected parameters

The CAMA tool is written to be flexible and versatile. A description of requested analysis is prescribed in a text file, and CAMA does the processing. The configuration allows for reading from any OMI level 2 product[1], perform simple arithmetic operations to combine several fields[2], and apply user-written filters – for example to translate a slant column to a vertical column using a geometric airmass factor. Several filters are supplied in the standard distribution.

Comparisons and correlations are calculated on a pixel by pixel basis, each file (of the same orbit number) is assumed to use the same latitude/longitude grid, because they are based on the same level 1b data. Different product files are matched based on the orbit number present in the file name. The results are written to an HDF-5 file and from that file a series of graphs are created. The results are summarised in a LaTeX report.

The toolkit is written to be memory efficient, and uses two passes: A first pass to calculate the averages, and a second pass to calculate the (co)variances to avoid the need for keeping all data in memory at once. With a limited number of parameters (6) a year of NO_2, TOMS O_3 and DOAS O_3 data (5316 orbits) takes about 16 hours to process on the TLCF cluster. In total this set sampled about 5×10^8 OMI measurements. Our experience with the O_2–O_2 cloud product has shown that by processing a single day of OMI data, roughly 15 orbits of data, the probability distribution functions no longer change significantly, but for other trace gases this may be very different.

2 References

Reference Documents

[RD1] R. J. Barlow. Statistics, a guide to the use of statistical methods in the physical sciences. The Manchester Physics series (John Wiley & Sons, 1989).

[1] Or data product stored in a similarly structured HDF-5 file

[2] For example subtracting OMI DOAS O_3 and OMI TOMS O_3 columns

[RD2] Gishan Dissanaike and Shiyun Wang. A critical examination of orthogonal regression 2003. Electronic paper no. 407560, DOI:10.2139/ssrn.407560.

3 Hard- and software requirements for CAMA

The CAMA toolkit is written in IDL, and was tested with version 6.2 and 6.3. A bug in the HDF-5 library of IDL in versions before release 6.3 can cause problems. An appropriate warning is printed when this bug is encountered. The bug is triggered by reading signed byte data from an HDF-5 file, causing all negative values to appear at zero. For IDL under Linux a *slow* work around is present to allow the run to continue. For Windows™ this workaround does not work correctly, and the run is aborted.

The latest version of CAMA was used exclusively with 6.3 – if you encounter problems when using CAMA with newer versions of IDL, please report back to us. Due to issues with HDF-5 support on earlier versions of IDL, all versions of IDL before 6.2 will *not* be able to run CAMA. The output of CAMA requires further processing, and the required and recommended software is listed in table 1.

Table 1: Requirements for other software when using CAMA.

Package	**Version**	**Status**	**Description**
TEX & co.	After ~2003	Required	To produce the PostScript output
GhostScript	7.00 or later, 8.5 preferred	Highly recommended	To produce PDF output. Version 8.5 recommended to prevent JPEG artifacts in the scatter plots.
pdftk	1.12	Optional	For compressing the (huge) PDF output, while preventing JPEG artifacts

The CAMA toolkit is in use at the KNMI on both Linux and Windows. When used on Linux the standard tools are assumed to be available (`sed`, `bash`, ...). The windows version at this moment requires more manual intervention. The toolkit should work on Mac OS X (as on any other Unix system) but this has not been tested explicitly.

The toolkit is written to be memory efficient, even when analysing very large data-sets. However, sufficient system memory is still a requirement. For a small number of parameters, and 100 histogram bins, the toolkit will work in 512 MB (Windows), but the memory required for comfortable processing increases quickly when adding parameters (the memory use is $\propto n^2$). A practical minimum of 1 GB main system memory is suggested to avoid swapping for most analysis runs. Please note that this number does not depend on the number of orbits that is to be analysed, as orbits are processed sequentially.

3.1 Installation

The CAMA source tree must be added to the IDL search path. Several additional IDL packages are required, and assumed to be available to IDL:

- The TEXtoIDL routines are used throughout CAMA for labeling purposes. You can find the latest version on `http://physweb.mnstate.edu/mcraig/textoidl/`
- A status line to show the progress, this requires `statusline.pro` from `http://cow.physics.wisc.edu/~craigm/idl/misc.html`
- From the coyote library, we use `undefine.pro`, `tvimage.pro`, `hak.pro`, `programrootdir.pro`, ... Please obtain the whole library, as it is a treasure trove anyway. `http://www.dfanning.com/documents/programs.html`

4 Short overview of the CAMA engine

A flow diagram of the CAMA software is shown in figure 1. The toolkit starts off with reading the configuration file. A sample configuration file is included in the distribution, and a full description can be found in section 6, starting on page 13.

After reading the configuration file, CAMA starts processing the orbits for "pass 1". This is a loop over all requested orbits calculating the averages, and preparing some histograms. At the start of the loop, all requested data for a particular orbit is read, possibly from multiple files. All orbit-files are assumed to use the official naming scheme used by OMI. A short overview of the naming scheme is given in table 2. CAMA will search on the short name and the orbit number, but can only recognise these items if the other parts of the file-name are present. When comparing an old version to an updated version of the same product, some items have to be added to the configuration file to allow CAMA to distinguish between the two versions, see section 6.3 for details. Optionally one can rename one of the file sets so that each has a unique short name. A tool written in IDL is available to rename a directory of files, giving them a new name. This tool is part of CAMA: "`rename_file_set.pro`". Documentation for this tool can be found in the file header.

Table 2: OMI-Aura level 2 file naming scheme.

`OMI-Aura_L2-OMCLDO2_2006m0421t2318-o09403_v002-2006m0505t142906.he5`	
OMI	Instrument name, here OMI on Aura
Aura	Satellite name.
L2	Level of the product, CAMA only handles level 2 files of OMI, not level 1b.
OMCLDO2	Short name, here for the OMI Cloud O_2–O_2 product.
2006m0421t2318	year, day and time of space craft midnight before the start of the orbit.
o09403	Orbit number.
v002	Collection number, here 002, further reprocessing yields 003, etc.
2006m0505t142906	year, day and time of this processing run.
he5	file type (HDF-EOS 5)

Each data-field is read in turn, filtered for missing data/fill value data, and fed to the (optional) filter and scaling function. The filter and scaling function is discussed in section 7 starting on page 18. After the filter and scaling function, the data will be filtered for out-of-range values, using item 9 of the field description, given in section 6.1. For fields that are a combination of other fields (such as an addition of two or more fields, or a subtracted value), the arithmetic operation takes place *before* the filter function, but the fields that are used for the addition, subtraction, etc. have been filtered already.

After reading the data, "pass 1" is used to calculate the sums for the averages and the pixel counts N per cross track position for each field.

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

In "pass 1" we also create the geolocated averages, the histograms and the scatter density plots.

After all orbits have been processed in "pass 1", the files are read again to obtain the covariance matrix and the variances per cross track position in "pass 2".

$$V(x) = \frac{1}{N-1}\sum_{i=1}^{N} (x_i - \bar{x})^2 \tag{2}$$

The problem of calculating the variance of a large data set is not trivial. By using an alternative formulation of equation 2: $V(x) = \overline{x^2} - \bar{x}^2$ and similar equations for the higher order moments, one could calculate the variance of the dataset in a single pass. However, this approach leads to numerical problems as one would need to calculate the difference between to nearly equal but (potentially) very large numbers. This leads to a significant loss in precision. Therefore a direct application of equation 2 in two passes (once to obtain $\bar{x}$ and a second time to calculate equation 2) is used in CAMA instead.

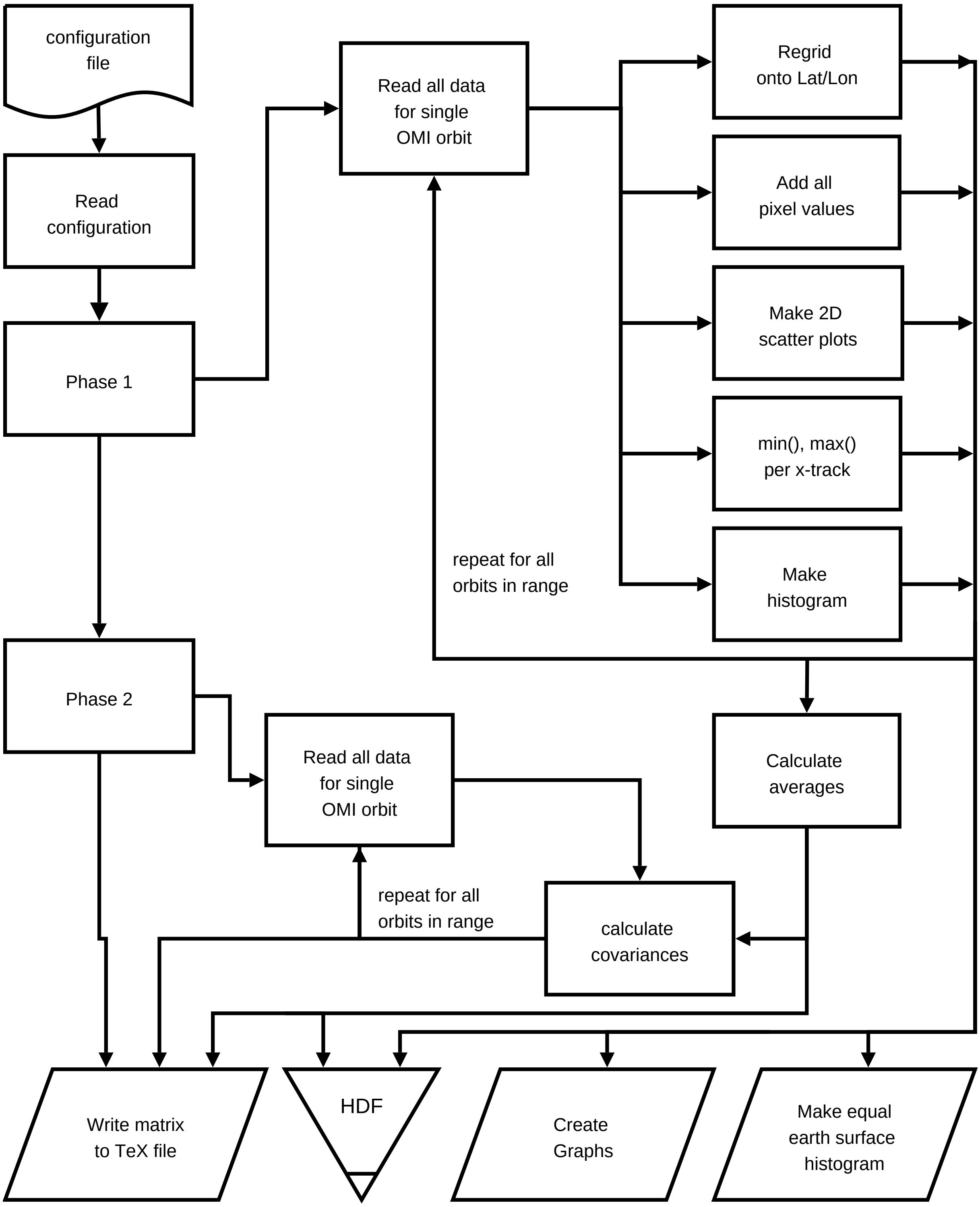

Figure 1: Flow diagram of CAMA. See text for a description.

The distribution of values is further described by the skewness γ and the kurtosis κ. Definitions vary in literature, but the definition implemented in CAMA is rather common – and is the same as the one used in the IDL `moment()` function.

$$\gamma = \frac{1}{N\sigma^3}\sum_i (x_i - \bar{x})^3 \tag{3}$$

$$\kappa = \frac{1}{N\sigma^4}\left\{\sum_i (x_i - \bar{x})^4\right\} - 3 \tag{4}$$

To get a measure for the relation of one variable $x_{(k)}$ with another $x_{(l)}$, we calculate the covariance matrix.

$$\mathrm{cov}(x_{(k)}, x_{(l)}) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{(k),i} - \overline{x_{(k)}})(x_{(l),i} - \overline{x_{(l)}}) \tag{5}$$

Rather than a dimensionally dependent covariance, it is often easier to interpret a correlation matrix:

$$\rho(x_{(k)}, x_{(l)}) = \frac{\mathrm{cov}(x_{(k)}, x_{(l)})}{\sqrt{V(x_{(k)})V(x_{(l)})}} \tag{6}$$

Inspection of equations 2 and 5 reveals that variances appear as the diagonal elements of the covariance matrix, scaled with a factor $N/(N-1)$. This factor is unity for sufficiently large N, as is already the case for a daily average of OMI, where $N \approx 10^6$.

$$V(x_{(k)}) = \mathrm{cov}(x_{(k)}, x_{(k)}) \tag{7}$$

With these numbers it is possible to calculate linear regression coefficients for all combinations of parameters. If we define $x_{(l)} = \hat{m}x_{(k)} + \hat{c}$, with $\hat{m}$ the estimated value of the slope and $\hat{c}$ the estimated value of the offset, then we find

$$\hat{m} = \frac{\mathrm{cov}(x_{(k)}, x_{(l)})}{V(x_{(k)})} \tag{8}$$

and

$$\hat{c} = \bar{x}_{(l)} - \hat{m}\bar{x}_{(k)} \tag{9}$$

These equations are listed in any introductory statistics book, for example [RD1], and are not repeated here. However, they silently ignore errors on the independent variable, and assume a constant error on the dependant variable. The resulting regression coefficients are known to be biased [RD2][3]. Because of this, the regression coefficients also have a somewhat disturbing feature: $\hat{m}(x_{(k)}, x_{(l)})$ and $1/\hat{m}(x_{(l)}, x_{(k)})$ may differ by tens of percents, even in cases where the correlation coefficient is as high as 0.9. In other words, it matters *which* is the independant variable.

There are other, unbiased estimators, and we are working to implement one of them [RD2]. However, truly meaningful regression coefficients should include estimates for the measurement errors in the individual points. These errors are not available for most of the OMI measurements, and work needs to be done to obtain error estimates by other means, which is also the reason that error estimates on the regression coefficients are absent.

After "pass 2", the analysis results are written to an HDF-5 file and the graphs are created. The LaTeX source files of the report are created as well. Since all information to re-create the graphs is present in teh HDF-5 file, it is possible to transfer the HDF-5 analysis file and the LaTeX sources to anotehr computer with IDL installed, and re-generate the report there.

4.1 Equal earth surface histogram

CAMA produces two types of histograms: "direct" histograms over all OMI pixels, and histograms that are scaled to account for the larger number of measurements over the polar regions. These "equal earth surface" histograms are based on the scatter density histograms of each parameter against the latitude, and after equalizing the total number of measurements for each latitude band, a cos(latitude) scaling is applied. This routine only works when the latitude is listed *last* in the list of requested fields as prescribed in the configuration file.

[3] The covariance matrix should include the measurement errors as weighting factors. In that case the errors are included and the outliers receive a reduced weight, but this does not remove the bias on the slope and offset.

4.2 CAMA caveats

There are some known issues in and limitations to what CAMA can do for you. The user should be aware of these limitations when interpreting the results.

- The geolocated averages are produced using a *very* simple algorithm, that only takes the pixel centres into account, no attempt is made to account for pixels that overlap multiple grid cells. This can lead to empty pixels in the gridded average when using a fine latitude/longitude grid *and* a limited time span (i.e. less than two days). The fact that OMI measurements can overlap multiple grid cells is *not* taken into account in the regridding process. For finer grids, one should consider to discard the outer OMI pixels.

- For reliable correlation coefficients the fields need to be 'synchronised', otherwise the covariance and the variances of the products may be calculated using (slightly) different data sets. A warning is printed in the caption of the appropriate table when the correlation coefficients are not reliable. See section 6.2 on page 15 for details.

- The filter functions expect the solar zenith angle (θ_0), the viewing zenith angle (θ), the solar azimuth angle (ϕ_0) and the viewing azimuth angle (ϕ). Not all of the US products include the azimuth angles, but include an azimuth difference instead. Aside from the fact that there is a sign ambiguity there[4], it trips CAMA. Make sure that the data-product from which the first parameter is read includes both azimuth angles, and not an azimuth difference. By reading the longitudes from a KNMI product, for example OMDOAO3, this limitation can be circumvented.

- The estimators for the linear regression coefficients as used by CAMA are known to be biased. They assume a constant (but unknown) error in the dependant variable $x_{(l)}$ and no error in the independant variable $x_{(k)}$. This is obviously never the case in real world data. See section 4 for details.

- Standard OMI data is produced in two-dimensional arrays: 60 positions across the track, and about 1650 measurements along track. Some products produce and array or a matrix for each point and generate three- or four-dimensional datasets. For the comparison a single element of such an array can be selected in a custom filter function (for example the "`slice`" function). However when doing this, the pixel count statistics for this fields are meaningless.

5 Calling CAMA

From within IDL, calling CAMA is as easy as any other procedure call. The procedures and functions are written such that pre-compilation is not needed, all requested functions will be compiled on the fly. The CAMA procedure has a series of optional keywords, listed below. If applicable, these keywords supersede values in the `description_file`.

```
IDL> CAMA, NAME=name, ORBITS=[9695,9738], DESCRIPTION_FILE="toms-doas-compare.txt", $
        analysis_results="toms-doas-compare.h5", /BATCHMODE
```

NAME=*name* The results will be stored in a directory with this name. Defaults to "`CAMA-`*`yyyymmmdd`*", where *`yyyy`* is the year, *`mmm`* is the three letter abbreviation of the month and *`dd`* is the day within the month. The directory will be created in the current working directory, unless an absolute path is given.

ORBITS=*[start,end]* Process these orbits, an array with two values (start – end, inclusive).

DESCRIPTION_FILE=*name* The file describing all requested fields, from which files to retrieve them, and how to preprocess them.

DISCBASE=*wildcard* Search for a list of disks that may contain the data. An '`ls -1d DISCBASE`' is performed to obtain the list. For use on the TLCF, typical value: "`/live*`" or "`/live[456]`". See section 5.1.1 for details. This option will *not* work on Windows.

[4] From discussions with A. Vassilkov I understand that the difference in the US products is the usual light path angle used in radiative transfer models.

DIRLIST=*list* Override the directory list in the configuration file, see item 6.2 in section 6.2 for details.

ANALYSIS_RESULTS=*name* Dump the results of the analysis to this HDF file (in the destination directory). If this file name contains a directory separator, the file is *not* created inside the normal destination directory, but at the absolute or relative location pointed to by this parameter. If the last character of this string is a '*', then the graph generation is skipped altogether.

PAPERSIZE=*paper* Set the output paper size for the report: "a0paper", . . . , "a6paper", "b0paper", . . . , "b6paper", "letterpaper", "executivepaper", and "legalpaper". This is passed to the LaTeX geometry package, and must be a string. Other papersizes can be obtained with "`papersize={`*`width`*`,`*`height`*`}`", for example to create a report on tabloid paper, the `papersize` argument should be set to: "`papersize={11in,17in}`". The default value for the `papersize` argument is "`a4paper`".

/DEBUG Print debugging information. This sets the IDL system variable `!quiet` to 0, which triggers the printing of extra diagnostic messages.

/BATCHMODE Do not try to open widget windows, at all. A CAMA run on the TLCF cluster will not finish without this parameter.

/PRESENTATION Create graphs suitable for inclusion in presentations (larger symbols, use Helvetica instead of Times).

The most important parameter is the "description_file" parameter, as it points to the configuration file that tells CAMA what to process. Calling CAMA without arguments will generate a series of interactive dialog boxes to obtain the information CAMA needs to run. This will not work on the TLCF, because interactive use is not possible, at least not at this side of the atlantic ocean, the windows end up on our proxy server..

5.1 Using CAMA on Linux

Of course it is perfectly possible to call CAMA from within IDL, but a shell script has been written to call CAMA from the command-line in Linux systems, and post process the output of CAMA with LaTeX, ps2pdf and a compressor to produce a clean PDF file. The shell script is part of CAMA, and available as `cama.sh`. Full help for this script is available with the "`--help`" option.

5.1.1 Special options for CAMA on the TLCF

The TLCF is a Linux cluster where all OMI data is available to OMI core science team members, provided they have an account on this system. CAMA will run on this system as well. However, there are some additional restrictions when using this system, and specific options exist for CAMA to work around these restrictions.

The first restriction is a safety measure in IDL. The file searching routines in IDL will never follow a symbolic link to a directory to avoid infinite recursive loops, and thus absolute file paths have to be provided. On the TLCF cluster, a path `/ecs2/OMCLDO2/2005/05/13` points to all data on a specific day. However, this is a symbolic link for easy access, and actual data are stored on a series of drives (named `/live1`, `/live2`, etc.). The drive names precede the directory names. There is an extra setting `disclist` to work around this limitation. The other extra settings, together with general syntax rules can be found in section 6.2 on page 15.

There are three methods of specifying the actual disks.

1. The easiest option is to use a wildcard in the "`dirlist`" setting, e.g.:

   ```
   ;# dirlist = /live*/ecs2/OMCLDO2/2006
   ```

   When using the file type aliases to compare different versions of the same algorithm, this is the *only* supported method for listing a sequence of directories.

2. The "`discbase`" setting, either an optional keyword parameter or an extra setting in the configuration file. This a wildcard search, that will generate the disclist parameter from a directory listing with a wildcard. A typical value is "`/live*`" to use all "`/live1`" through "`/live14`" (or however many there are) drives to search for the product files. Another example is "`/live[456]`" to use drives "`/live4`", "`/live5`", and "`/live6`".

3. The “`disclist`” extra setting in the configuration file is useful when the first part of the actual file paths varies, e.g., at the TLCF where data for one product, e.g., NO_2, may be distributed over many discs. It is used in combination with dirlist, which sets the second part of the filepaths. For example, combining

```
;# dirlist = /ecs2/OMDOAO3/2006/04:/ecs2/OMTO3/2006/04
;# disclist = /live12:/live13
```

   yields four possible file paths that will be searched, and is the same as setting `dirlist` to:

```
;# dirlist = /live12/ecs2/OMDOAO3/2006/04:/live12/ecs2/OMTO3/2006/04:
    /live13/ecs2/OMDOAO3/2006/04:/live13/ecs2/OMTO3/2006/04
```

   Please note that line breaks are *not* supported in the configuration file.

For some users, a second restriction is the fact that IDL cannot open displays over a remote connection, and thus cannot create widgets. It is not possible to detect this from within IDL[5], and therefore a keyword `/BATCHMODE` exists to turn off all interactive windows. All input and output will be handled through the terminal.

When running a long process on the TLCF, the user may want to be able to break the connection (sometimes this is forced by the TLCF because the connection is classified as idle). This can be done by spawning an IDL batch process and re-routing (most) output to /dev/null, e.g., `idl ominous.idl >& /dev/null &`. Make sure the batch file ends with the “`exit`” command! If this sounds complicated, use the `cama.sh` shell script in the background, as was pointed out in section 5.1, to run IDL.

Running CAMA on the TLCF can also be done through the `cama.sh` shell script, as was pointed out in section 5.1, or directly from the command line:

```
$ idl ominous.idl >& /dev/null &
```

after which you can log off. Another worthwile option is to use the `screen` program under linux. It allows you to place a run in the background, and reattach to that session later on, details can be found on the `screen` man page. The file `ominous.idl` in either case is a simple IDL batch file:

```
CAMA, DESCRIPTION_FILE="CAMA/WholeGlobe/20050900WG1.verigen", $
    /batchmode, discbase="/live*", papersize="letterpaper"
EXIT
```

5.2 Using CAMA on Windows

Using CAMA on a Windows machine to process OMI level 2 data puts rather high demands on the system performance. OMI level 2 data files are of considerable size, between 6 MB and 50 MB per orbit and $\sim$ 15 orbits per day, hence ample hard drive space is required for OMI data to be stored locally, or on a network drive. CAMA processing requires at least 512 MB of system memory to avoid continuous memory swapping, here we recommend 1 GB of system memory. CAMA processing requires a modern generation system CPU, here we recommend a Pentium 4 or similar at $>$ 2.0 GHz clock speed to avoid processing time beyond regular office hours. Processing CAMA output requires an additional $\sim$ 200 MB hard drive space, processing CAMA output with LaTeX adds another $\sim$ 200 MB and conversion to PDF requires an additional $\sim$ 3 – 5 MB.

Using CAMA on a Windows machine to process OMI level 2 data and working with the output of CAMA on a Windows machine requires the following software to be installed: IDL 6.3, LaTeX processing software such as MikTeX, with the TeXNicCenter front end, a PostScript viewer such as GhostView 4.7, a PDF creator such as “JAWS”, and a PDF viewer such as Adobe Acrobat Reader 7. Later versions of this software work even better.

Using CAMA on Windows starts with installing CAMA. The CAMA toolbox can be installed anywhere on a system hard drive by simply copying the entire directory. After installing make sure that in IDL the "Path" specification in "Preferences" points to the exact location of this directory plus its subdirectories.

Calling CAMA in IDL on a Windows machine works the same as with IDL on any other operating system. Simply type "CAMA" on the command line, complemented by filling in the desired options, such as the description file. For example:

[5] This may be a configuration issue, bit 16 in `!D.Flags` seems to be set at all times on the TLCF.

```
CAMA,DESCRIPTION_FILE="E:\Calculations\IDL\CAMA\WholeGlobe\20050900WG1.verigen"
```

where CAMA is instructed to read the specified "`.verigen`" file at the specified path.

The main caveat of using CAMA on Windows is a memory allocation leak. IDL on Windows only allows for ∼600 files to be opened and closed, even though CAMA has been carefully programmed to return Logical Unit Numbers (LUN's) to Windows. Processing single jobs larger than ∼600 files should not be performed with CAMA under windows.

For processing jobs involving many different operations on an OMI data set one can write an IDL program to instruct IDL to call the CAMA routine a number of times with each time a different "description file". However, to avoid running into the before mentioned caveat, IDL should be reset in between these calls and unfortunately this can only be performed manually in IDL on Windows. Advantageously, IDL on Windows allows for calling the `IDLDE.EXE` program directly from a DOS prompt and from a DOS batch file. For example, the file "`mass_run.bat`" contains lines of the following form:

```
"C:\RSIIDL\IDL61\bin\bin.x86\idlde.exe" \
    @E:\Calculations\IDL\CAMA\WholeGlobe\WG_mass_run_verigen_12.pro
```

where "`C:\RSIIDL\IDL61\bin\bin.x86\idlde.exe`" specifies the location of the local installation of the idlde.exe program and the text directly following the "`@`" sign specifies the location of a `.pro` file. In this `.pro` file regular CAMA calling routines can be written such a specified above. To achieve resetting IDLDE between parts of the processing job, the last line of code of the `.pro` file should read "`EXIT`" forcing IDL to exit. Subsequently, the next line of command in the batch file restarts IDLDE with the default setting and all counters reset to zero. In this way a large number of processing jobs can be programmed to conveniently run outside office hours. Please do make absolutely sure that the option "Confirm Exit" in section "General" of "Preferences" is turned *off*, otherwise IDL on Windows requires a manual confirmation from the user to exit.

Running LaTeX, viewing PostScript files, and creating and viewing PDF files is all possible under Windows with the before mentioned professional software. Current versions of PowerPoint™, Excel™ and Word™ can read the encapsulated PostScript files that CAMA generates. Older version of these applications require conversion to a bitmapped image, for example using ImageMagick or GhostScript.

6 CAMA configuration file

The configuration file describes what fields to process and where to search for the input data. The configuration file is a plain text file, and consists of two distinct parts: the description of the fields, and the other settings. There are some general rules for the format of the configuration file:

- A semi-colon (`;`) at the *start* of a line indicates a comment.
- The parameters are defined in a colon (`:`) separated list. Each parameter needs 14 fields, these are described in section 6.1.
- Additional, optional settings are given as special comments: `;# setting = value`
- A short text for the report can be given between `;@` and `;!`. Each line of that section starts with a `;`.

A sample configuration file is part of the distribution, and demonstrates all recognised settings.

6.1 Field descriptions

All these items are required. Each field is described by 14 items, separated by colons (`:`). An annotated example is shown in appendix B.

1. The field type:

   DATA A normal Data or Geolocation field.

IGNR A field that is read (and needed), but otherwise ignored. This is typically used for the longitude. However, this field is included when synchronising the parameters, hence filtering on this field is still possible. The 'ignored' fields are available for multiplication, addition, et cetera.

CALC A field that is fully calculated from the viewing geometry, like a total scattering angle.

DIFF A difference between two previously read fields.

PROD A product of two or more previously read fields.

RATO The quotient of two previously read fields.

PLUS The sum of two or more previously read fields.

2. The short name of the product file that is to be read: OMCLDRR, OMCLDO2, OMTO3, OMDOAO3, et cetera.

3. The swath name within the file, e.g. 'CloudFractionAndPressure' for the OMCLDO2 product. This name is most easily obtained by opening a product file with the HDFView application.

4. The HDF field name to read. Both the geolocation and the data fields are searched for this field-name. These names are most easily obtained by opening a product file with the HDFView application. In case of a `DIFF`/`PROD`/`RATO`/`PLUS` field, use two (or more) comma-separated *local names* to indicate the needed fields. Local names are given in field number 6.

5. Second HDF field name, for processing flags or whatever. Both the geolocation and the data fields are searched for this field-name. Can be empty.

6. The local name. This *must* be unique, is case insensitive, and no spaces, commas, hyphens, or plusses are allowed. This name is used for output and to create an IDL structure (it must be a valid IDL variable name).

7. A user-defined processing function name. Applied after filtering on fill values/missing data, and after scaling with the scale attributes. Can be empty.

8. Arguments, passed to the filter function. Can be a comma separated list of multiple values.

9. Limits, comma separated bottom and top. The filters are applied after the user processing function and the field can be empty. If only one value is given, it is assumed to be the first (lower value). To specify an upper limit alone, use `:,limit:`, where `limit` specifies the number value.

10. Histogram limits, comma separated bottom and top. Both must be present and finite.

11. Pretty name: The label along the axis in the various plots, and in the LaTeX output. Simple LaTeX syntax can be used: `$\theta_0\$` will yield θ_0 in the graphs and the report. Note that this function uses TeXtoIDL, and does not implement all of TeX. Greek letters, and super- and sub-script work, but multi-letter subscripts require some tricks, that are not compatible with TeX. For example "c_{eff}" can be obtained for graphs with "`$c_e_f_f$`". This requires some search-and-replace on the generated LaTeX files to obtain the report: search for `$c_e_f_f$` and replace with `$c_{\text{eff}}$` in the report.

12. Color table: The index number for the IDL colour table in the geolocation plots, a negative number uses the same colour table, but in reverse. If a procedure name is given, it is assumed to set the system color table. The procedure will be called without arguments, but the minus is still applied. Can be empty, color table 25 is used as default.

13. The altitude where Google Earth should show this product.

14. A longer description for use in Google Earth.

6.2 Extra settings

The extra settings are used to define general properties of the analysis, like the number of histogram bins, et cetera. All extra settings are specified as a meta-comment: a comment formatted as `;# setting = value`. Some of these settings control the same behaviour as the arguments to the CAMA procedure. The IDL command line overrides the configuration file when both are present.

n_histogram_bins The number of histogram bins in the 1D- and 2D-histograms. Defaults to 100.

n_omitrack The number of cross track positions of OMI. Defaults to 60.

grid_resolution The regridding resolution for the geolocation plots. Defaults to 2°.

global_mapping Switch to create the geolocation plots (or not). Defaults to 1 (True).

label_geolocation_plots Switch to add labels to the geolocation plots themselves (rather than relying on the caption alone). The default value is 0, meaning that no labels will be added, set the value to 1 to add the labels.

cross_track_band Specify the cross track indices to use in the analysis Defaults to `0,59`, comma separated `low,high` (inclusive).

figures_base The location where the results directory is created. Defaults to the current directory.

results_set The name of the resulting output set. A directory with this name will be created within the `figures_base` directory. Must be specified here, or as an optional keyword to the CAMA procedure.

dirlist Where to look for the input data. A colon (Linux) or semi-colon (Windows) separated list of directory names. When not present a series interactive dialogs are presented to the user to find the data files. Once all data locations are given, cancel the dialogs to continue with the processing run.

disclist A list of disks to prepend to the directory list items to work around the use of symbolic links on the TLCF (or perhaps multidisk Windows systems). A colon (Linux) or semi-colon (windows) separated list of mount points. A full description is given in section 5.1.1.

discbase A setting corresponding to the discbase keyword parameter to the CAMA procedure. Setting this variable will override the "`disclist`" setting.

orbit_start The first orbit to search for, this orbit is *included* in the analysis. The starting orbit must be given here, or the orbit range must be given as an optional keyword to the CAMA procedure.

orbit_end The last orbit to search for, this orbit is *included* in the analysis. The ending orbit must be given here, or the orbit range must be given as an optional keyword to the CAMA procedure.

asynchronous_filters If set to false (0), then if a value is discarded in one field, it is discarded in all fields (default). If set to true, the filters act on the fields independently, and the correlation matrix is not reliable as a result.

histogram_limits_for_geolocation_plot Force the histogram limits upon the geolocated averages.

reverse_filters Reverse the action of the filter limits: keep everything *outside* the indicated range.

area_of_interest Limit the geolocation plot to this region. A comma separated list of lat_{min}, lon_{min}, lat_{max}, and lon_{max}. To limit the actual CAMA analysis to these boundaries, use the limits on the latitude and longitude parameters, and synchronise the fields.

presentation Set to 1 to generate the graphs in presentation mode (larger font, Helvetica instead of Times).

analysis_results The name of the HDF-5 output file. If the name is not given as a parameter to the procedure and this setting is not present in the configuration file, then the name given in `results_set` is appended with `.h5` and used instead. If the name ends with a `*`, then the generation of the graphs is omitted[6].

[6] The star itself is removed from the file-name.

papersize Set the output paper size for the report: "a0paper", ..., "a6paper", "b0paper", ..., "b6paper", "letterpaper", "executivepaper", "legalpaper". This is passed to the LaTeX geometry package, and must be a string. Other papersizes can be obtained with "`papersize={`*`width`*`,`*`height`*`}`", for example to create a report on tabloid paper, the `papersize` argument should be set to: "`papersize={11in,17in}`". The default value for the `papersize` argument is "`a4paper`".

batchmode Turn on batch processing, switch off any interactive dialogs, as if `/BATCHMODE` keyword was specified.

scene_hunting When present, the scene locations matching *all* conditions are recorded in the HDF-5 file refered to in this setting. Try this on a limited data set first, the file may grow to huge sizes very quickly.

scene_hunting_fields When hunting for scenes, it is possible to record not only the location of matching pixels, but also one or more pixel values from the data-set. Set this extra setting to a comma separated list of local names (field 6 in the list given in section 6.1).

apply_bad_row_filter If set to 'yes' a file will be downloaded indicating which rows are affected by the OMI row anomaly. Rows affected by the anomaly will be removed if this is set to yes. Downloading the most recent file will require IDL 6.4 or later.

bad_row_filter_location The default location is '`http://www.knmi.nl/omi/research/validation/cama/badrows.txt`'. You can download the file and change this to a local '`file://`' url to use a local copy. This is primarily for use with IDL versions prior to IDL 6.4. Do not forget to regularly update the file if you use this method.

instrument_name Set the instrument name. The default is obviously 'OMI', but level 2 data providers for other instruments (notably GOME 2) have discovered this tool as well. This will replace the occurence of 'OMI' with the value you specify. Note that the LaTeX template also contains many occurances of OMI, but at least you can use the individual images in your reports.

skip_scatter_plots If set to 'yes', then the scatter plots are not generated. This can dramatically reduce the filesize of the output.

6.3 File type aliases

When comparing two versions of the same algorithm it is necessary to distinguish the old and the new version. The short names for both files are the same, and some special configuration is needed to deal with this issue. The special configuration comes in the shape of "file type aliases". In the configuration, one of the versions has to be given a non-existant short name (field 2 in section 6.1). This non-existant file type will act as an alias if two extra settings are defined:

***_alias** This is a special setting to link the non-existant short name to the real short name. If the non-existant short name is `OMDOAO3Z`, and is meant to represent a new version of `OMDOAO3`, then a line with

```
;# omdoao3z_alias = OMDOAO3
```

indicates the link.

***_dirlist** This special setting tells CAMA where to search for the new version. If the non-existant short name is `OMDOAO3Z`, then a line with

```
;# omdoao3z_dirlist = /live*/ps0003/OMDOAO3
```

gives the path to the tree with the new version. This parameter has the same rules as the standard `dirlist` parameter described before.

It is important to properly separate the two versions. The new version should *not* appear at all in the standard dirlist search path, and the old version should not appear in the directory tree indicated in this aliased directory list. Note that all subdirectories of either tree are searched as well.

6.4 Google Earth output

The CAMA configuration files can be used by the `make_bitmapped_omi_day_plot` as well to produce a single day plot for Google Earth. The field descriptions are completely the same as for a normal CAMA run, just some extra parameters are needed.

central_location Determine the point on Earth which will be the initial center point for Google Earth. A `lat,lon` pair; defaults to De Bilt, the Netherlands.

description A description of the CAMA run for display in Google Earth. The date which is displayed is appended to this string.

earth_view_year The year,

earth_view_month The month,

earth_view_day And the day to display.

earth_view_use_orbit_range Set this to 'yes' if you want to use a particular range of orbits, instead of selecting on the date in the product file. Use this for instance to plot zoom data. Defaulst to 'no'.

earth_view_orbit_start The starting orbit when `earth_view_use_orbit_range` is set to 'yes'. Defaults to `orbit_start`.

earth_view_orbit_end End of the orbit range (inclusive) when `earth_view_use_orbit_range` is set to 'yes'. Defaults to `orbit_end`.

earth_view_resolution The display resolution, in pixels per degree. Defaults to 50.

earth_view_base Directory location for Google Earth output. Defaults to `figures_base`, and has a similar function.

earth_view_name Name of the output set. Defaults to `results_set`, and has a similar function.

The calling sequence for producing Google Earth output is as follows:

```
make_bitmapped_omi_day_plot, description_file, $
    year=year, month=month, day=day, $
    output=name, $
    /reduce
```

description_file The CAMA file.

year, month, day The date to plot, these override `earth_view_year`, `earth_view_month`, and `earth_view_day` respectively.

output The name of the output data set, overriding `earth_view_name`.

/reduce A flag to reduce the file size by under sampling after plotting. This only works on Unix, and only when *ImageMagick* is installed.

There are some known limitations; in particular the handling of the international date-line is not particularly elegant.

6.5 Report description

The configuration file may contain a special comment block which is places in the report. The block starts with `;@` and ends with `;!`. An example is included below:

```
;@
; This will be placed in the report.
; You can use \LaTeX{} syntax here.
; \par
; Like: $\mu_0 = \cos \theta_0$
;!
```

The explicit `\par` is needed because empty lines are removed by IDL.

7 Custom filter functions

One of the more powerful features in CAMA is the ability to use filter and transformation functions that can be selected from within the configuration file. This feature can be used to filter data based on processing flags, or scale a slant column to a vertical column using a simple geometric air mass factor. This feature increases the versatility of CAMA considerably, as it is easy to write your own filter and scaling functions, using the supplied examples as a starting point.

Because these functions are called through the `call_function` IDL call, there is a rigid format for the function itself. A starting point for writing your own function is available in `verification_generator_pass.pro`.

7.1 Paremeters for a custom filter function

The filter function expects the following parameters, in this order:

Raw The input data, the data pointed to by field 4 described in section 6.1 starting on page 13.

Flags The extra field, that is the data pointed to by field 5 described in section 6.1 starting on page 13.

SZA The solar zenith angle θ_0 in degrees.

VZA The viewing zenith angle θ in degrees.

SAA The solar azimuth angle ϕ_0 in degrees east of north[7].

VAA The viewing azimuth angle ϕ in degrees east of north[7].

Parameter A string holding the contents of field 8 described in section 6.1 starting on page 13.

The minimal definition of a *do-nothing* filter looks like this:

```
function VERIFICATION_GENERATOR_PASS, raw, flags, sza, vza, saa, vaa, param
    compile_opt defint32, strictarr, strictarrsubs

    return, raw
end
```

[7] These fields are present in level 1B and the NL/KNMI level 2 products, but not in all US/GSFC products. Also note that a direct difference of the solar and viewing azimuth angles is *not* the azimuth difference used in most radiative transfer models, but off by 180°. The azimuth difference in the rotational Raman cloud product has performed this correction. The IDL function `azimuth_difference()` can be used to obtain the azimuth difference as used in many ratiative transfer models from the SAA and the VAA.

To use this function, store it in a file called "`verification_generator_pass.pro`", make sure it is available within the IDL search path and use "`PASS`" in field 7 in the configuration file. All filter functions start with "`verification_generator_`". This string is prepended to field 7 before calling the function to avoid function name collisions within IDL.

The returned array is used for further processing, it is safe to alter the `raw` argument inside the function. To remove particular values from further processing, set them to `NaN` (the IDL variable `!values.f_nan` or `!values.d_nan`, depending on the type of the raw data). Note that the raw array already has `NaN` values (at the beginning and end of the swath), and that the IDL maximum and minimum operators (> and <) can not be used. Instead this type of filtering should be done with `where()` function calls.

The `param` function argument can be extracted with the `extract_values_from_comma_separated_list()` function. This function has one required argument, and two optional ones.

param Required, the string as passed into the custom filter function.

length Optional. The number of expected real parameters. If absent or ≤ 0, the number of elements (separated by commas) is used instead. A fixed number can be forced by setting this argument to a value > 0. Missing values are set to either `NaN` (for floating point numbers), the largest negative value (for integer values), or an empty string.

type The type of the returned array. Optional, defaults to 5 (double). A list of possible values is available from the IDL documentation, see the help for the '`size()`' function.

7.2 Available custom filter functions

The following filter functions are included in the standard distribution.

pass A do-nothing filter. Not needed for actual use, leaving field number 7 empty will achieve the same. This is the recommended starting point for writing your own filter function.

ignore The same as the "pass" filter, it exists for historical reasons.

scale Scale the raw data with a factor given in the param argument.

airmass_factor Use a geometric air mass factor to scale a slant column into a vertical column.

rad_irrad_to_reflectance_scaling Scale the sun normalised radiance by π/μ_0 to obtain a reflectance.

total_scatter_angle Calculate the single scattering angle Θ.

logarithm Take the logarithm of the raw values.

relative_scaling Returns `raw/flags`, for example to create relative errors.

ascending_filter Filter the latitude to remove the ascending or descending part of the orbit. Set the parameter to a value ≥ 0 to keep the ascending, south to north, dayside part of the orbit, set the parameter to -1 to keep the descending, north to south, begin or end of the swath part of the orbit. If the second item in the parameter list is larger than zero, the latitudes are assumed to be passed in the `flags` field, described in item 5, section 6.1 starting on page 13. If the parameter is zero or absent, the latitudes are assumed to be passed in through the `raw` field, item 4 in the same list.

ctp Set the data value to the cross track index.

groundpixelflags_filter Filter the ground pixel quality flags to keep or remove land or water, ice & snow or no ice & snow. The array that is returned is 1.0 for accepted values, and `NaN` for rejected pixels. The use can be twofold: either synchronise, and get an analysis for all fields, of just read in (IGNR), and multiply with a particular field, to obtain a joint analysis in a single run.

nise_import Import the values from the NISE snow and ice-cover database.

omcldo2_fraction If the parameter is equal to 1, this filter will clip the raw values to the interval $[0,1]$.

omcldo2_pressure This filter adds a small amount of Gaussian noise to the raw data – to avoid Moiré patterns on the histograms – and uses the cloud fractions in the flags argument to remove cloud pressures when there is a cloud fraction less than the values passed in the parameter. The flags are assumed to be cloud fractions.

fraction_filter Filter on the flags field with a minimum and a maximum, similar to the filtering done with field 9 in section 6.1 on the filtered data itself.

omcldrr_filter The flags are assumed to be a bit field which can be used to filter the raw data. If any bit in the flags is set that is also set in the parameter, that pixel is removed, i.e. if `flag`&textttparam $\neq 0$ then the value of `raw` for that pixel is set to `NaN`.

selectaerosoltype Filter the data in order to include in the analysis only those pixels to which a certain aerosol type is assigned. To this end, specify 'AerosolType' (for the near-UV aerosol product) or 'AerosolModelMW' (for the multi-wavelength aerosol product) as the second HDF field name (parameter 5 in section 6.1). Further, specify one of the following values as the param argument: For the near-UV aerosol product: 1 for biomass burning, 2 for desert dust, or 3 for weakly absorbing. For the multi-wavelength aerosol product: 1 for weakly absorbing, 2 for biomass burning, 3 for desert dust, or 4 for volcanic. If there are any pixels with the specified aerosol type, the raw data for all other pixels are replaced by `NaN`. If no param is specified, the raw data are returned unchanged.

slice Select only the raw data for the specfied 'slice' of a multi-dimensional data field. For instance, if one of the datafields lists a quantity for seven separate wavelengths (i.e., has dimensions `nWavel` × `nXtrack` × `nTimes`), and you want to use the third of these wavelengths, specify '2' for the param argument. For even higher-dimensional data fields, give a comma-separated array as the param argument, e.g. '`2,3`'.

Note: Using multi-dimensional datafields, with or without applying the slice filter will result in incorrect pixel count statistics in the report.

sza_filter Remove pixels if the solar zenith angle is larger than the value provided in the parameter, with a default value of 88°.

8 Terms and conditions

The conditions for the use of CAMA are specified in section 8.1. The first condition is trivial, and will not cause any trouble. However, the second condition may raise some eyebrows. The reason to include this condition is to ensure that users will contact me before using the results in a publication. With CAMA there are about 101 ways to shoot yourself in the foot, and this condition will help me to check that CAMA is used correctly, and that the results can be trusted. If the use of CAMA is a minor part of a publication, then a simple acknowledgement will suffice, but this decision is *mine* to make. Maarten Sneep, 2006, KNMI. Maarten Sneep can be reached via email: maarten.sneep@knmi.nl.

8.1 Software license and disclaimer

8.2 Support

There is *no* support. That said, we do like to receive feedback on CAMA and the documentation. If you encounter problems, we will try to resolve those, or offer a way to work around them. We are happy to receive code to fix issues, additional filter functions, etcetera.

A Contributors

Maarten Sneep is the main author of CAMA, which he initially developed to investigate the OMI O_2–O_2 cloud product, and later extended so it could be used for all OMI products. The software was consequently tested, debugged, and extended further by three people: Remco Braak (Linux, various aerosol-related filter functions), Ellen Brinksma (Linux, TLCF), and Mark Kroon (Windows, various filters and operations).

B Sample configuration file

This is a sample configuration file. In a valid configuration file the field descriptions can not be broken across multiple lines. In this sample the broken lines are marked with "`---- `".

```
; There are some meta-comments, true comments start with '; ', possible meta-comments
; are listed at the end of this file.
; Each non-comment line describes one item to be taken used in the analysis, this file
; is for checking a cloud (OMCLDO2) file. Each item takes 14 fields, the fields are
; separated by colons (ASCII character 58, ":").
;
; The fields:
;  1 - Field type: DATA, CALC, IGNR, DIFF, PROD, PLUS, or RATO. (data or geolocation
;      field, calculated from viewing geometry, data field - but not correlated,
;      difference, product, sum, ratio).
;  2 - File identifier: OMCLDO2, OMCLDRR, OMDOAO3, OMTO3, ...
;  3 - Swath name.
;  4 - HDF field name. In case of a DIFF/PROD/PLUS/RATO field, use two (or more, where
;      that makes sense) comma-separated local names to indicate the needed fields.
;      Local names are in field number 6.
;  5 - Second hdf field name (flags or whatever). Can be empty.
;  6 - local name. This must be unique, no spaces allowed.
;  7 - User processing function name. Applied after filtering on fill
;      values/missing data, and scaling. Can be empty.
;  8 - parameter (passed to the filter function). Can be a comma separated list of
;      multiple values.
;  9 - limits, comma separated bottom and top. The filters are applied after the
;      user processing function. Can be empty. If only one value is given, it is
;      assumed to be the first (lower value). To specify an upper limit alone, use
;      :,limit:
; 10 - histogram limits, comma separated bottom and top. Both must be present.
; 11 - pretty name: The label along the axis in the various plots, and in the
;      LaTeX output.
; 12 - color table: The index number for the colour table in the geolocation plots,
;      a negative number uses the same colour table, but in reverse. If a procedure
```

```
;      name is given, it is assumed to set the system color table. Can be empty,
;      table 25 is used in that case.
; 13 - Reserved for future use. Leave empty for now.
; 14 - Reserved for future use. Leave empty for now.
;
; To obtain equal surface area plots, the latitude must be the last item in the list
; of items. Each parameter must be on a single line (that will not fit on the page,
; and is not done here. ---- is the continuation mark).
;
IGNR:OMCLDO2:CloudFractionAndPressure:Longitude::Longitude:
---- :::-180.0,180.0:longitude:1::
;
; Sea/land filter (use the (default) synchronised filters, or multiply with this
; value (using another (IGNR) field).
;
IGNR:OMCLDO2:CloudFractionAndPressure:GroundPixelQualityFlags::GroundPixelQF:
---- GROUNDPIXELFLAGS_FILTER:noland,water,noice::0.0,1.0:ground pixel:0::
;
DATA:OMCLDO2:CloudFractionAndPressure:CloudPressure:CloudFraction:CloudPressure:
---- OMCLDO2_PRESSURE:0.2::125.0,1025:cloud pressure:3::
;
DATA:OMCLDO2:CloudFractionAndPressure:CloudFractionNotClipped::CloudFraction:
---- :::-0.2,1.5:cloud fraction:1::
;
DATA:OMCLDO2:CloudFractionAndPressure:CloudFraction::CloudFractionClip:
---- :::0.0,1.0:cloud fraction (clipped):1::
;
PLUS:OMCLDO2:CloudFractionAndPressure:CloudFraction,CloudFractionClip::SumTest:
---- ascending_filter:0,1::-0.2,3.5:Summation test:23::
;
; Use ascending node only.
;
DATA:OMCLDO2:CloudFractionAndPressure:Latitude::Latitude:
---- ASCENDING_FILTER:::-90.0,90.0:latitude:25::
;
; End of the records
;
; Description: The file format defines some meta comments. All lines
; between ;@ and ;! (at the start of the line) will be included in the report,
; with the initial ;'s and spaces removed.
;
;@
; The number of histogram bins is set to 200. The grid resolution to
; $1^{\circ}$. Cloud pressures are filtered on cloud fractions $\ge 0.2$.
; Only pixels over water are included, and only pixels from the ascending
; part of the orbit.
;!
;
; The lines that start with ;# are variables for use in the main program. These are
; the default values that are used when the variable isn't found in this file.
; The function grab_var_from_description is used to extract these variables.
;
; The number of bins in the histograms. Default = 100.
;# n_histogram_bins = 200
```

```
;
; The number of cross track positions in an OMI swath. Default = 60.
;# n_omitrack       = 60
;
; The regridding resolution. Default: 2.0.
;# grid_resolution  = 1.0
;
; Create geolocated averages. Defaults to true (1).
;# global_mapping = 1
;
; Only include cross track positions indicated here. This can be used to investigate
; certain cross track dependencies.
; cross_track_band = first,last (inclusive)
;# cross_track_band = 0,59
;
; The name for the results set: the "name" parameter for the CAMA procedure.
;# results_set = CAMA-out
;
; The list of directories to search (colon separated on linux, semicolon separated
; on windows). Because of the risk of recursive loops, symbolic links are not
; followed. Use disclist to prepend these paths with a set of directories to
; look through.
;
;# dirlist = /nobackup/OMIDATA/OMCLDRR/:/nobackup/OMIDATA/OMCLDO2
;
; The orbits to run through
;# orbit_start = 08239
;# orbit_end   = 08259
;
; Destination directory (figures).
;# figures_base = /usr/people/sneepm/nobackup1/idl-output
;
; Use the old asynchronous filters (1) or the synchronised filters (0).
; Defaults to 0.
;# asynchronous_filters = 0
;
; For the geolocation plots you can either use the histogram limits (1),
; or the natural limits of the (averaged) data set (0).
;# histogram_limits_for_geolocation_plot = 1
;
; Limit the geolocation plot to a limited area (lat_min, lon_min, lat_max, lon_max)
;# area_of_interest = 0.,45.,8.,53.
;
; Reverse the filter action: keep everything outside the range (1)
; or 'normal' limits (0). Defaults to 0.
;# reverse_filters = 0
;
; All arrays with analysis results can be dumped into an hdf-5 file.
;# analysis_results = cloud-verification-basic.h5
;
; Presentation mode. If set to 1, use Helvetica and a bigger font in the graphs.
;# presentation = 1
```

C Sample CAMA output

The output of a sample run of CAMA and the configuration file to create this report on the TLCF cluster is attached here. The files are embedded in the pdf file, and can be saved to your hard drive from the contextual menu (right-click on the paperclip).

The CAMA output for a simple run on three days of OMI O_2–O_2 cloud data.

The CAMA configuration file to generate the report attached above.