Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

High Performance Software Framework for the Calculation of Satellite-to-Satellite Data Matchups (MMS version 1.2)

Thomas Block[1], Sabine Embacher[1], Christopher J. Merchant[2] and Craig Donlon[3].

[1]Brockmann Consult GmbH, Max-Planck-Str. 2, 21502 Geesthacht, D
[2]Department of Meteorology, University of Reading, Reading, RG6 6AL, UK
[3]ESA-ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, NL

Correspondence to: Thomas Block (tom.block@brockmann-consult.de)

Abstract. We present a Multisensor Matchup System (MMS) that allows systematic detection of satellite based sensor-to-sensor matchups and the extraction of local subsets of satellite data around matchup locations. The software system implements a generic matchup-detection approach and is currently being used for validation and sensor harmonisation purposes. An overview of the flexible and highly configurable software architecture and the target processing environments is given. We discuss improvements implemented with respect to heritage systems, and present some performance comparisons. A detailed description of the intersection algorithm is given which allows a fast matchup detection in geometry and time.

1 Introduction

There is increasing exploitation of Earth observation (i.e., satellite remote sensing) data for a range of scientific and societal applications, in relation to environmental monitoring including climate science (e.g., Hollmann et al., 2013). Often, long (multi-decadal) data records are required for such applications, so that observed changes can be put in the context of a time-series of earlier variability, for example to determine how common is a particular observation. Since satellite missions last typically 5 years, such long data records need to be constructed from a series of data records from similar-but-not-identical sensors. The differences between members of such sensor-series consist of factors such as detailed spectral responses (i.e., slightly different sensitivities to Earth radiation at different wavelengths), differences in calibration procedures and performance in flight, the effects over time through each mission of degradation of the sensors, and so on. Such differences can cause significant discontinuity in the derived records of environmental variables between different sensors, and steps need to be taken to minimise such artefacts, essentially to maximise the signal for real natural variability compared to sensor-difference effects. A key tool for harmonising satellite-derived data records across multiple sensors is to compare the radiances observed by sensors when viewing, to as close an approximation as possible, the same environmental scene simultaneously. Such paired observations are referred to as matchups, and are critical to exploit for multi-sensor data records. This paper addresses progress on the important practical step of creating datasets of matchups to use for harmonising or bias-correcting across sensors to create consistent, multi-decadal data records.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

The detection of satellite data matchups is a numerically intensive process that incorporates geographic searches in large satellite datasets, which may be of order a hundred Terabyte in size. A highly performing search and data extraction system has to be operated in a parallel processing environment to achieve manageable processing times for this task. Despite the importance of being able to analyse long time series of matchup data, surprisingly few general concepts are published to detect and extract the data. Some publications use a relatively small number of satellite intersections that have been detected manually (e.g. Illingworth 2009) whereas other teams develop custom solutions for restricted pairs of sensors (e.g. Bali, 2015).

For the European Space Agency Climate Change Initiative project for Sea Surface Temperature (SST CCI), a first system to identify and generate sensor/sensor matchups has been developed in an early phase of the project (Boettcher et al., 2012). This system, initially targeted to operate on a single dataset, has been used and extended during the project lifetime (the past 6 years). Increasing input dataset sizes, stricter requirements on matchup conditions and the necessity to operate on small matchup time differences led to the decision to create a completely new software that implements all the lessons learned during operations of the earlier software. The system is operated on the Climate and Environmental Monitoring from Space Facility (CEMS) at the Centre for Environmental Data Analysis (CEDA)[1]. The same architecture is also being further developed within the framework of the European Union project "Fidelity and Uncertainty in Climate data records from Earth Observation" (FIDUCEO).

The initial system exhibited some performance bottlenecks, predominantly caused by the way a database was used, too many read accesses to satellite data products and a fixed processing time interval. Especially the use of the database as storage location for satellite acquisition metadata and matchup candidate locations (i.e. read and write access to the database) prevented a significant parallelization to improve the overall performance. Design goals for the new system architecture have been defined as:

- Use database only for storage of satellite metadata, distribute numerically expensive geometric calculations to parallel nodes and keep the matchup candidate data locally.
- Implement a system that allows reading of the satellite data products late in the matchup detection process. The software shall detect matchup candidate pairs reliably before actually accessing the file system – reducing several hundred possible candidate products to a small number of consolidated candidates.
- Design the system as extensible frame with a large number of plug-in points.
- Facilitate integration of new sensors.
- Implement a scalable system that can run on a notebook as well as allowing a high parallelization on a dedicated processing environment, avoid single-points of access.

[1] http://www.ceda.ac.uk/services/analysis-environments/

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

Following these design goals we have implemented a high performance matchup system that already has been generated various long-term sensor matchup datasets, ranging from 1979 to 2016, covering combinations of 7 visible and microwave sensors from 23 different satellite platforms.

The MMS consists of three major components:

- A database server for the storage of satellite data metadata records
- An "IngestionTool" to extract the metadata from the satellite acquisitions and store it in the database
- A "MatchupTool" that performs the matchup processing and writing of the result data.

The interaction of these components implements the functionality described in the following sections.

2 Definition

In the context of the SST-CCI and FIDUCEO projects, we define a Satellite Data Matchup as a coinciding measurement of the same location on earth at almost the same time using different space borne instruments. The location criterion is defined as the spherical distance in kilometres between pixel centre locations, the time constraint as pixel acquisition time difference.

In most use-cases there exist additional criteria on top of this raw matchup definition, which include for example cloud free condition, constraints on the viewing angles, water pixel constraint and many more; we summarize these conditions as screening criteria (section 6).

The result of a matchup-processing run is stored as a Multisensor Matchup Dataset (MMD, section 7) which includes all data required to analyse the matchups without the need to access the original satellite input datasets. These files include satellite data extracts covering a symmetrical window of n by m pixels around the matchup point, copying all variables of the input data as geometrical subset to the MMD.

3 Matchup processing

Based on the intention to assemble a generic matchup generation system, we have identified several common steps. During the analysis, we also identified the steps/concepts that require sensor or domain specific handling; all of these have been encapsulated as abstractions that hide sensor/domain specific implementations. Our design incorporates a database containing satellite acquisition metadata, which has to be filled as a first step:

1. Ingest satellite metadata into database

Any matchup generation process starts with a database query:

2. Request sensor metadata for time interval and sensor combination to process
3. Perform matchup candidates pre-selection (section 4)
4. Perform matchup candidates fine selection (section 5)
5. Run condition and screening processes (section 6)

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

6. Assemble subset data and write MMD (section 7)

These steps are implemented using two distinct programs, an IngestionTool (performing step 1) and a MatchupTool (performing steps 2 to 6).

4 Metadata and Candidates Pre-Selection

4.1 Metadata extraction

The satellite metadata stored in the database has been constructed in a way that allows detecting matchup candidate pairs reliably without the need to open the associated satellite data products. The detection is a two-step process that consists of the detection of a geometric intersection followed by a calculation of the acquisition times of both sensors for the common area. If the acquisition times match, taken into account the maximal pixel time-difference allowed, a set of possible matchup pairs has been detected.

The geometry metadata for each satellite product is constructed using the approach sketched in Figure 1. Two entities are being generated by regularly sampling longitude/latitude pairs from the geolocation data rasters, a bounding polygon and a time axis line-string. The bounding polygon is constructed by stepping around the borders of the rasters, in a counter-clockwise direction, the time axis by sampling in flight direction at the centre of the swath acquisition – ignoring the one-pixel-offset occurring in products with an even number of pixels per line. In case of self-overlaps (which happen often for e.g. Advanced Very High Resolution Radiometer (AVHRR) data) the geometries are split into two segments, to ensure valid geometry objects.

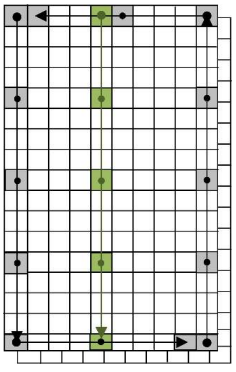

Figure 1: Construction of the satellite product boundary polygon (black) and time axis line-string (green). The rasters are the longitude and latitude data arrays, acquisition time increases top-down.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

4.2 Intersection Detection

A geometric intersection between two satellite acquisitions can simply be calculated by performing an intersection of the associated satellite data bounding geometries. The calculation of the timing intersection is not as simple as this; the acquisition times of the data are stored with the satellite data, in most cases either per pixel or per scan line. Which means that the timing data is stored in the swath coordinate system (x/y acquisition raster) whereas to be useful for geometric intersection calculation, the timing data needs to be in a geographic (longitude/latitude) coordinate system. Mathematical transformations

$$f(x,y) = (\phi,\lambda) \text{ and } g(\phi,\lambda) = (x,y) \quad (1)$$

between the coordinate systems can be calculated using e.g. polynomial approximations but these operations require access to the full longitude and latitude data raster. To avoid these time-consuming reading and approximation operations, the MMS implements a different approach based on two assumptions with respect to polar orbiting satellite data:

1. The satellite moves locally with constant velocity on a continuing orbit path.
2. Acquisitions per time instant are located on an axis normal to the nadir point path, which is true for push-broom type instruments and almost true for scanning instrument types where the instrument field of view tracks along the nadir plane.

Consequently, a time-axis geometry is constructed using the approach described above, i.e. we create a line-string geometry at the nadir pixel by sampling the location data at regular intervals, e.g. every 20 scan-lines. When now associating the sensing start and stop times of the acquisition with the start and end-points of the line-string, we can approximate the acquisition time of every geo-location on the time axis by taking into account the line-length along the axis:

$$t_{point} = \left(t_{stop} - t_{start}\right)\frac{\Delta_{point-start}}{\Delta_{stop-start}} \quad (2)$$

where Δ denotes the length along the line-string, which can be calculated by well-known algorithms (e.g. Goodwin, 1910). Based on assumption (2) we can also associate time information to any point within the swath geometry by applying a normal projection along a great-circle of the point onto the time axis. This approach allows calculating the overflight times of any geometric intersections between two polar orbiting sensors. For the implementation of the geodesic calculations, we rely on the Google S2 library that allows fast geometric calculations on the surface of a 3-dimensional spheroid; please also refer to (Varun, 2016).

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

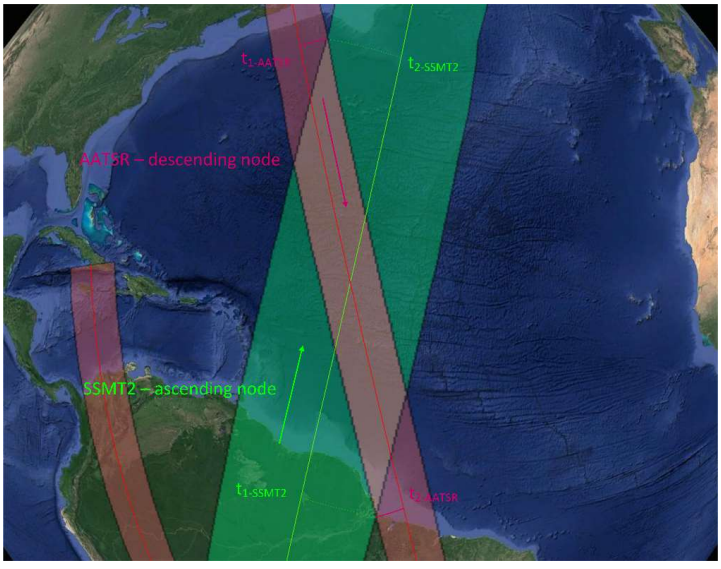

Figure 2: Intersection between Advanced Along-Track Scanning Radiometer (AATSR) and Special Sensor Microwave Water Vapor Profiler (SSM/T-2). AATSR on a descending node, SSM/T-2 on an ascending node. The projection of the intersection geometry extreme locations onto the time axes of both sensors (on the centre of the swath) allows detecting the overflight time. If the time intervals $t_{1\text{-AATSR}} - t_{2\text{-AATSR}}$ and $t_{1\text{-SSMT2}} - t_{2\text{-SSMT2}}$ are within the configured maximal time delta, possible matchups can be found inside the intersection geometry.

4.3 Time Estimation Errors

The estimation errors introduced by this approach vary between some milliseconds up to 17 seconds for certain AVHRR data. The wide range of errors varies between sensor types and is generally becoming larger for instruments generating wider swath data. However, in the context of natural variability of most environmental parameters at the spatial scales of observation from space, O(10 s) is a relative small error.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

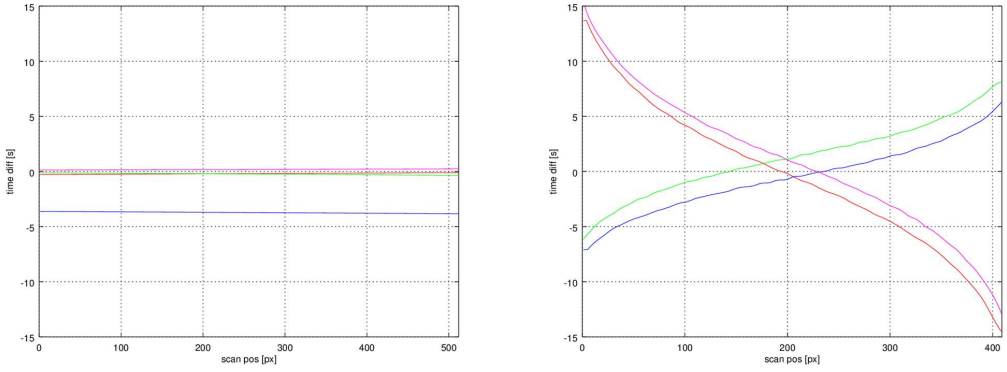

Figure 3: Time error per scan line for AATSR (left) and AVHRR (right). Plotted is the difference between the real and the estimated acquisition time per pixel for the beginning (red), middle (blue and green) and end (magenta) of a complete product.

The errors distribution per scan-line varies from almost constant (AATSR) to a quasi-cubic distribution function (AVHRR). The relative high estimation precision for AATSR is due to the small swath of 512 km whereas for the AVHRR case with a swath width of 2892 km the effects of interpolation errors become visible. The larger error line in the AATSR case (blue line in Figure 3) is generated by the ellipsoid effects, see below.

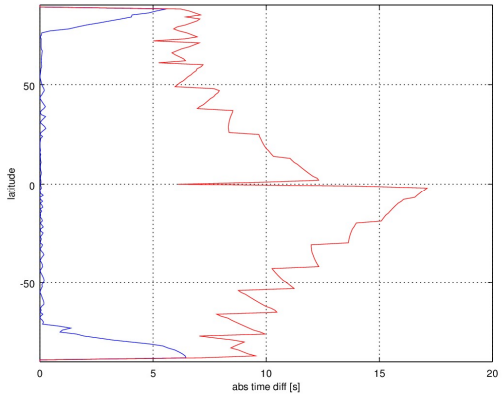

Figure 4: Plot of absolute value of acquisition time estimation error versus latitude for SSM/T-2, minimum value (blue) and maximum value (red).

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

The estimation errors are mainly originating from two effects, which both can be identified when plotting the estimation error versus latitude (see Figure 4). The effects are predominantly visible when inspecting the maximum error value plot in red. The saw-tooth like structure is created by the interpolation error that occurs due to the spatial sampling of the time axis. The minima of the saw-tooth are located on sampling nodes of the time axis, where no interpolation is required, the maxima in the middle between two points using a maximal interpolation length. This error can be minimized by using a smaller interpolation interval, although it has to be balanced with the data storage volumes.

On top of the saw tooth is a constant maximal error that slowly increases from the poles to the equator. This effect is due to the underlying assumption that the earth is a sphere. The deviation of the ellipsoid is rising to a maximum at the equator when embedding the sphere into the true earth form (almost an ellipsoid) so that both geometric forms touch at the poles. The sharp dip at the equator cannot be explained at the moment and needs further investigation.

In the context of matchup candidate pre-selection we can easily compensate for these timing errors by adding a sensor-dependent grace interval (i.e., time tolerance) on top of the configured matchup maximum time difference, set by scientific requirements on the matchups according to the purpose of the MMD. Thus, we ensure that we do not reject possible matchup candidate pairs due to interpolation errors.

4.4 Performance Comparison

The following table demonstrates the performance gains obtained by the improved pre-selection algorithm. The time axis approach is described in section 4.1, the full access algorithm implements the standard approach to open each file and read geolocation and timing information to determine the overflight times for the matchup pre-selection.

Table 1: Performance comparison of the pre-selection algorithms for some project use-cases[2]. Acronyms in this table are: ATSR – Along Track Scanning Radiometer, E2 – ERS2 satellite platform, AVHRR - Advanced Very High Resolution Radiometer, Nxx – NOAA xx satellite platform, HIRS - High-resolution Infrared Radiation Sounder, MA – MetOp-A satellite platform, AMSUB – Advanced Microwave Sounding Unit B, SSM/T-2 - Special Sensor Microwave/Temperature-2, F15 – DMSP – F15 satellite platform, AMSRE - Advanced Microwave Scanning Radiometer - Earth Observing System, AQ – Aqua satellite platform, AATSR - Advanced Along Track Scanning Radiometer, EN – Envisat satellite platform

Sensors	ATSR-E2/ AVHRR-N14	AVHRR-N11/ AVHRR-N10	HIRS-MA/ HIRS-N17	AMSUB-N15/ SSM/T-2-F15	AMSRE-AQ/ AATSR-EN
Time interval	2000-08-14 – 2000-08-18	1989-05-01 – 1989-05-07	2009-04-06 – 2009-04-12	2000-10-16 – 2000-10-22	2005-02-16 – 2005-02-17
Time axis	1205 s	822 s	102 s	104 s	35 s
Full access	2507 s	1415 s	124 s	119 s	125 s

[2] Test have been performed on a standard desktop PC: Intel Core I7-5820K, 16 GB RAM, 256 GB SSD for operating system, 3 TB SATA HD for satellite data, Linux Kubuntu 14.04 64 bit, MongoDB server v 3.2.1, Oracle JDK 1.8_45. All tests have been executed three times, the times in the table are the averaged execution times. MMD writing times have been excluded from these measurements.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

As demonstrated by the examples in Table 1, the performance gained varies largely between approx. 20 percent up approx. 350 percent, depending on the input data. Generally, we can conclude here that the performance improvement is small when processing small input data files e.g. AMSU-B or HIRS, where the data volumes are in the range of 5 MB and the acquisition raster is small (e.g. 946 x 90 pixels for HIRS). In these cases, the time for loading the geolocation and time datasets is comparably small. The situation changes significantly when processing satellite data with larger volumes, as for example AATSR (ca. 40000 x 512 pixels, ca. 800 MB), where performance improvements up to a factor of 2 to 3 can be observed.

Another set of parameters that influences the performance is the duration of the satellite acquisition and the maximal pixel time difference allowed. When the time difference is small compared to the acquisition duration of a product, for example 5 minutes pixel time difference using AVHRR data with an orbit acquisition time of approx. 1:44 hours, there exist many geometric intersections within the acquisition time interval of both orbit files. Most of these intersecting geometric areas will be discarded because the common overflight times differ by more than the allowed 5 minutes time interval. This decision can be made without access to the datasets using the time axis approach, thus avoiding opening a large number of product pairs that are discarded later.

5 Matchup detection

The fine matchup detection process is operating on the reduced set of pre-selected matchup candidate pairs. This stage of the processing aims to detect all pairs of pixels that comply with the raw matchup conditions as defined in section 2, i.e. time and location condition. The numerically expensive calculations are executed on the consolidated list of candidates, iterating over all pairs of files that have been identified in the pre-selection stage.

A first step opens both satellite data files, reads the geo-location information and constructs a geo-coding approximation that implements the mapping between the (ϕ, λ) and (x/y) coordinate systems as described by Eqn.1. In addition, a TimeCoding object is constructed for each of the data files which maps the sensor specific internal time information to the MMS system reference time (UTC seconds since 1970).

For the primary or reference sensor, all pixels contained in the intersection geometry are collected into a list and the matching closest (in a geodesic sense) pixel location in the associated sensor data is calculated.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

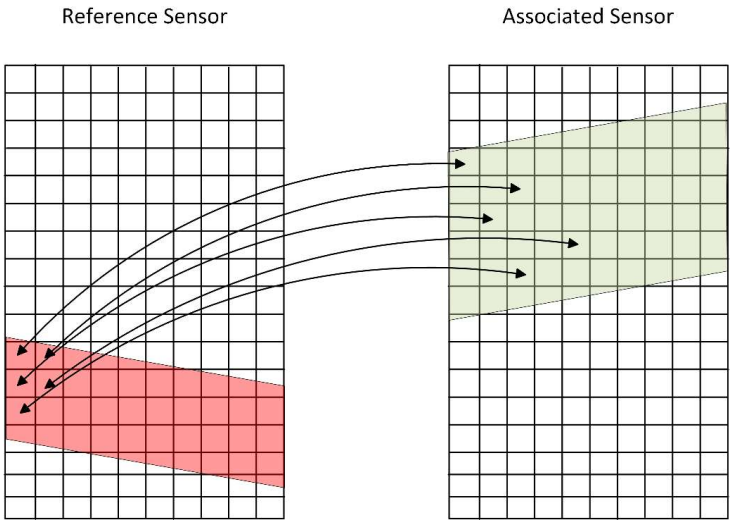

Figure 5: Fine matchup detection, all pixels contained in the intersection geometry of the reference sensor (red) are associated with the closest pixel in the associated sensor data contained in the intersection geometry (green). Both intersection geometries cover the same area in the geographic coordinate system.

Based on these pairs of locations, the acquisition times for both sensors are calculated for each matchup. This list of pixel pairs is immediately scanned for acquisition time difference and spherical distance of pixel centres; all pairs not conforming to the processing parameters (i.e. maximal time difference allowed and maximal pixel centre distance) are rejected at this stage. The remaining list contains for the pair of satellite acquisitions being processed the following values:

- Matchup geo-location (lon/lat) for both sensors
- Pixel raster position (x/y) for both sensors
- Acquisition time in UTC seconds since epoch for both sensors.

This list serves as the basis for further processing.

6 Matchup conditions and screenings

The result sets generated up to this processing step contain verified matchup locations conforming to the basic search criteria of acquisition time constraint and relative local neighbourhood based on the sensor geo-coding calculations. These can easily contain many ten thousands of results per intersecting acquisition geometries. In most cases, this list needs to be narrowed down to a smaller subset that conforms to a set of constraints given by the research context. These constraints can be for

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

example constraints in the viewing geometries, atmospheric conditions, ground conditions, spatial subsets, instrument conditions and many more.

The MMS implements a two-step processing stage to perform this narrowing down procedure, using a very fast first stage, the so-called conditions, and a slower second stage, the screenings. During the second stage, the numerically more intensive screenings operate only on a significantly reduced number of possible matchups, increasing the performance of the overall processing.

As we expected these processes to vary a lot between the projects and the scientific requirements, we have implemented both processing stages using a general engine/plugin approach. Each engine operates a chain of plugins that are loaded dynamically based on the configuration file, using the Java SPI mechanism (Seacord, 2002).

Each plugin implements a narrow interface and is identified by its unique name. The configuration of the engine is a structured XML document, containing subsections for each plugin; the order of the subsections defines the order of processing. Plugin configuration tags allow to identify the plugins, all sub-configurations are passed to the plugins to parse so that the engine is completely decoupled form the plugin-module configurations.

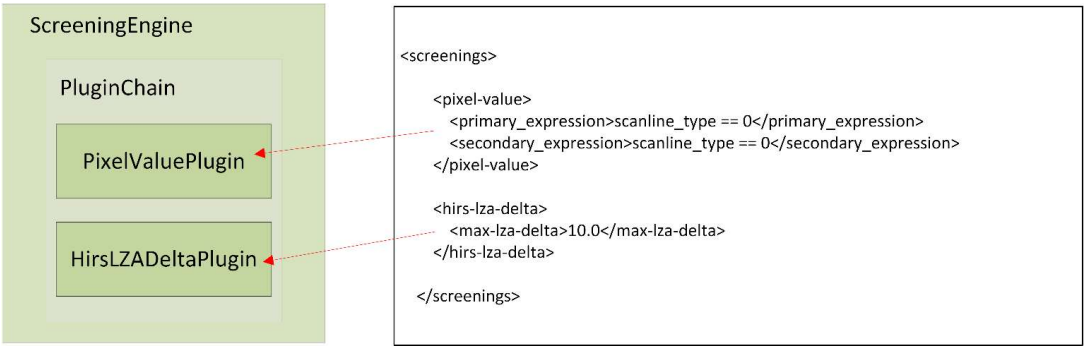

Figure 6: Screening engine with plugins and configuration snippet

6.1 Conditions

Condition plugins operate on the core matchup information solely, i.e. longitude/latitude, product raster x/y positions and acquisition times. These entities are kept in memory during processing, thus any condition plugin executes very fast and is not slowed down by e.g. disk access times. Although an extremely reduced parameter set is available, a number of useful condition processes have been implemented, e.g.:

- BorderDistance: reject matchups where the matchup raster x/y position is too close to a swath border
- OverlapRemove: reject pixels where subset windows overlap.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

6.2 Screenings

In contrast, screening plugins also are given access to the satellite products associated to the matchups so that any variable from any input file can be used for a screening algorithm. At this stage, it might also be possible to access auxiliary datasets, although this has not been requested yet. A number of sensor specific and generic algorithms are available; the following list just covers some examples:

- Angular: perform screening on viewing geometry constraints (e.g. maximal Viewing Zenith Angle (VZA), maximal VZA difference)
- BuehlerCloud: a cloud screening for microwave sensors based on (Buehler et al, 2007)
- PixelValue: screening that allows to evaluate mathematical expressions on matchup pixels (e.g. *reflec_nadir_0870 > 7.5 && reflec_fward_0670 > 4.3*). This plugin is very generic; it also allows defining complex flag combinations to be evaluated.

7 Matchup files

The results of the matchup detection and screening processes finally are stored in Multisensor Matchup Datasets (MMDs) using either NetCDF3 or NetCDF4 format. The design of these files shall enable scientist to work with the matchup data without the need to access the original input data files. To allow for this, we copy all input data variables that are contained in a symmetrical n by m pixel window around each matchup to the MMD (the sizes are configurable per sensor). To reflect the calibration purpose of the software, we copy the data always in the original input format, without any modifications or scaling applied. All variable attributes are also copied to the MMD.

The final MMD contains for each sensor input variable a three dimensional dataset where the x and y dimensions are the extensions of the extraction window and the z-dimension is the matchup index. For sensors that originally contain 3-dimensional input data (e.g. HIRS radiance variable), we split the data into separate 2-dimensional channel variables. This approach ensures that every (x/y/z) tuple for each variable of each sensor belongs to the same location and time.

In addition to the input data we also store the raw matchup information, namely original raster x and y position, the unified acquisition time in UTC seconds since epoch and for each z-layer the names of the input data files. This ensures that the MMD content is completely traceable and every matchup pair refers to its data origin. One MMD file for each processing interval (e.g. one week) is generated on a parallel processing environment.

Geosci. Model Dev. Discuss., doi:10.5194/gmd-2017-54, 2017
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 31 May 2017

Geoscientific Model Development Discussions

8 Further work

In addition to the sensors currently supported (AMSR-E, AMSU-B, (A)ATSR, AVHRR, HIRS, MHS, SSM/T2) there are already extensions scheduled to support AIRS, IASI, MVIRI and SEVIRI. For the geostationary sensors (MVIRI and SEVIRI) a different pre-selection approach will be taken. The acquisition time per scene/slot is relatively short and the area covered can be considered as constant. In this case, it is expected that the time information alone yield an appropriate pre-selection criterion. At the time of writing this text, a first development version of the MMS is also capable of performing satellite/in-situ data matchup processing. This first engineering version can operate on SST data collected for the SST-CCI project; further extensions scheduled will cover AERONET data and GRUAN radiosonde measurements.

9 Summary and Conclusion

We have demonstrated and implemented a general matchup processing system using a novel and fast intersection detection algorithm for satellite based sensor overlaps. The time-axis approach for acquisition time detection and the use of a novel library for spherical calculations allow performance gains of up to a factor of 3.5 compared to a conservative implementation. We introduced a two-step screening system that is highly configurable to adapt the matchup process to several scientific requirements.

This paper concentrated on the algorithmic facets of the software system, disregarding other structural improvements we have implemented. For further information on architecture and usage, we like to refer to documents on the project websites[3].

The software has been used for operational processing on the CEMS parallel environment in both projects. A Python based processing scheduler implements the glue code to the Load Sharing Facility (LSF) interfaces (Ault et al., 2004). At the time of writing, the operational database is a MongoDB server that contains approximately 1.6 million metadata records referencing approx. 130 TB of satellite data products. The overall performance observed when operating on whole mission datasets matches our initial expectations; as example: processing AVHRR NOAA 18 vs NOAA 17 (covering May 2005 to December 2010) is executed in 2:30 hours, using 72 parallel nodes on the CEMS cluster.

10 Code Availability

All components of the system described above are available as GNU General Public License (GPL) licensed open source software modules. The software is published using a Github code repository[4] including a Maven based project setup. A binary distribution will be made available using the FIDUCEO project web page. The system is coded in Java and Python, it has been tested on Windows 10 and several Linux 64 bit operating systems using Java 1.8 and Python 2.7 / 3.2. We used a test-driven development approach following Kent Beck (Beck, 2003), resulting in an extremely stable system with an overall test-coverage

[3] FIDUCEO: http://www.fiduceo.eu/, SST-CCI: http://www.esa-sst-cci.org/
[4] https://github.com/FIDUCEO/MMS

of > 95%. Database servers mentioned in the manuscript are not part of the MMS software, these can be obtained from the database vendors.

11 Acknowledgements

The development of this system was funded by the European Space Agency (ESA) project SST-CCI that is part of the ESA Climate Change Initiative (CCI). Major parts of the research were also supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 638822 (FIDUCEO project).

The MMS software system relies on a number of publicly available libraries:

- Apache Commons: https://commons.apache.org/
- FasterXML: https://github.com/FasterXML
- Google S2: https://code.google.com/archive/p/s2-geometry-library/
- SNAP: http://step.esa.int/main/toolboxes/snap/
- Unidata NetCDF: http://www.unidata.ucar.edu/software/thredds/current/netcdf-java/

12 References

Michael R. Ault, Mike Ault, Madhu Tumma, and Ranko Mosic (2004). Oracle 10g Grid & Real Application Clusters. *Rampant TechPress.* p. 24.

Bali, M., Mittaz, J., Maturi, E., and Goldberg, M.: Inter-comparison of IASI and AATSR over an extended period, *Atmos. Meas. Tech. Discuss.*, 8, 9785-9821, doi:10.5194/amtd-8-9785-2015, 2015.

Beck, Kent, Test Driven Development by Example, *Addison-Wesley*, 2003.

M. Böttcher, R. Quast, T. Storm, G. Corlett, C. Merchant, C. Donlon: Multi-sensor match-up database for SST CCI, *ESA Sentinel-3 OLCI/SLSTR and MERIS/AATSR Workshop 2012 in Frascati, Italy*. 10/2012, doi: 10.6084/m9.figshare.1063282

S. A. Buehler, M. Kuvatov, T. R. Sreerekha, V. O. John, B. Rydberg, P. Eriksson, and J. Notholt: A cloud filtering method for microwave upper tropospheric humidity measurements, *Atmos. Chem Phys. Discuss*., 7, 7509–7534, 2007

H. B. Goodwin, The haversine in nautical astronomy, *Naval Institute Proceedings*, vol. 36, no. 3 (1910), pp. 735–746

Hollman, R., Merchant, C. J., Saunders, R., Downy, C., Buchwitz, M., Cazenave, A., Chuvieco, E., Defourny, P., de Leeuw, G., Forsberg, R., Holzer-Popp, T., Paul, F., Sandven, S., Sathyendranath, S., van Roozendael, M. and Wagner, W. (2013) The ESA climate change initiative: satellite data records for essential climate variables. *Bulletin of the American Meteorological Society, 94 (10)*. ISSN 1520-0477 doi: 10.1175/BAMS-D-11-00254.1

Illingworth et al.: Intercomparison of integrated IASI and AATSR calibrated radiances at 11 and 12 µm, *Atmos. Chem. Phys.*, 9, 6677-6683, doi:10.5194/acp-9-6677-2009, 2009.

Seacord, Robert and Wrage, Lutz. Replaceable Components and the Service Provider Interface. CMU/SEI-2002-TN-009. Software Engineering Institute, Carnegie Mellon University. 2002.

Varun Pandey, Andreas Kipf, Dimitri Vorona, Tobias Mühlbauer, Thomas Neumann, and Alfons Kemper. 2016. High-Performance Geospatial Analytics in HyPerSpace. *Proceedings of the 2016 International Conference on Management of Data (SIGMOD '16)*. ACM, New York, NY, USA, 2145-2148. doi: http://dx.doi.org/10.1145/2882903.2899412