ОТ ATCOR К НОВОЙ СИСТЕМЕ

А. Макаров[1], Р. Рихтер[1], П. Швинд[1], А. де Мигель[2], П. Рейнарц[1]

[1]Немецкий аэрокосмический центр (DLR)
Веслинг, Германия
[2]Elecnor Deimos Space
Мадрид, Испания
e-mail: aliaksei.makarau@dlr.de

Система атмосферной коррекции (ATCOR) является одним из самых известных и точных кодов атмосферной коррекции для земли для спектрального диапазона 0,4–2,5 µ*m*. Она генерирует продукты требуемого уровня 2А и включает дополнительные функции, такие как создание карты классификации, обнаружение и удаление дымки/перистых облаков, топографическую коррекцию, теневую компенсацию и т. д. Появившись более 25 лет назад, ATCOR постоянно совершенствуется и ее функциональность растет. Структура, процессуальное оформление, и коммерческий характер системы шин (Interactive Data Language, IDL) побудило нас внедрить новую конструкцию системы, основанную на языке Python, который не зависит от условий лицензии.

В этой статье мы дадим обзор требований, новую структуру системы, вопросы передачи программного обеспечения, и исходы при разработке и осуществлении.

Ключевые слова: атмосферная коррекция; ATCOR; излучающая передача; оптические данные.

FROM ATCOR TO THE NEW SYSTEM

A. Makarau[1], R. Richter[1], P. Schwind[1], A. de Miguel[2], P. Reinartz[1]

[1]German Aerospace Center (DLR), D-82234 Weßling, Germany;
[2]Elecnor Deimos Space, Ronda de Poniente 19 Edificio Fiteni VI Tres Cantos, Madrid 28760 Spain

The ATmospheric CORrection (ATCOR) system is one of the most known and accurate atmospheric correction codes for land for the 0,4–2,5 µ*m* spectral range. It generates the required Level 2A products and includes additional features such as generation of the classification map, haze/cirrus detection and removal, topographic correction, shadow compensation, etc. Started more than 25 years ago ATCOR is continuously improving and the functionality grows. The structure, procedural implementation, and the commercial nature of the bus system (Interactive Data Language, IDL) motivated us to implement a new system design based on the Python language, which is independent of license conditions.

In this paper we give an overview of the requirements, new system structure, software transfer issues, and outcomes of the design and implementation.

Keywords: atmospheric correction; ATCOR; radiative transfer; optical data.

INTRODUCTION

Atmospheric correction (Level 2A or L2A acronym) includes the following derived products: bottom of atmosphere (BOA) reflectance (albedo), aerosol optical thickness map, water vapor map. Additional functions, among them cirrus detection/removal, haze detection/removal, shadow removal, approximate bidirectional reflectance distribution function (BRDF) correction, ozone correction, smile correction, and generation of quality layers are the special features of ATCOR [1, 2]. This variety of functions in the ATCOR system is performed on an operational basis. The initial choice of the IDL bus system [3] was motivated by no appropriate alternative for research and development software at the end of the 1980th. Nowadays the ATCOR development and support with the IDL language may require more cost than with the use of modern object oriented approaches and freely available programming languages with rich development libraries.

The aim of the project is to transfer the software system into a modular and highly configurable structure with several base elements for further development and support. The base elements are the module (the module estimates a specific feature like water vapor), the data (orthorectified top of atmosphere radiance L1C, or L2A product consisting the raster data as well as the corresponding metadata and sub products), input/output modules, and the scenario. The scenario is the particular way to process the input L1C data given the developed and validated methodology. The methodology is based on the physics of a process, while the application may vary according to the instrument and the L1C product. The new system is to be operated in the ground segment for a variety of optical multi- and hyperspectral missions with small, medium, and wide field-of-view sensors, and for the airplane mode (EnMAP, MERIS, Landsat 8, Sentinel 2, DESIS, Hyspex) as well as the software installed locally on the user side.

SYSTEM NEW STRUCTURE

The base requirements for the new atmospheric correction system are: modularity (ability to include/skip a module), setting the input parameter set for each module, ability to process small or wide field-of-view, smile, and airplane sensor data, ability to process large volumes of data, data with different ground sampling distance (GSD in Sentinel 2), parallel execution of modules, selection of L1C data reader and L2A data writer according to customer requirements, to extend for L2A products for water area, reflectance retrieval function selection for a specific wavelength region (visible, shortwave infrared; thermal emissivity and temperature), object oriented paradigm of software.

The parts of the new system are super classes defining: a *module* (performs analysis of data), input and output products (define input L1C and output L2A data, sub-, and auxiliary products), input/output classes (L1C reader and L2A writer), and classes defining all necessary metadata. The newly defined modules or products are the offsprings of these base classes. The new system has the following processing modules:

- *decirrus* – cirrus detection and removal [4];
- *dehaze* – haze detection and removal [5];

- *slopasp* – slope and aspect data fast calculation;
- *dtm* – digital elevation model module (prepares the necessary data derived from the elevation map);
- *rt* – radiative transfer calculation module (to estimate the optical properties of the atmosphere: path radiance, direct/diffuse, irradiance/transmittance, spherical albedo of the atmosphere, direct transmittance sun to ground);
- *masking* – classification module;
- *at_estimation* – aerosol type estimation module;
- *ozone* – correction for ozone influence [6];
- *aot* – aerosol optical thickness estimation module (based on dense dark vegetation approach);
- *sc_lp* – path radiance scaling;
- *wv* – water vapor estimation module [7];
- *wv_smile* – water vapor estimation module with account of sensor smile [8];
- *reflectance* – BOA reflectance calculation;
- *reflectance_smile* – BOA reflectance calculation with account of sensor smile [8];
- *brdf* – empirical BRDF correction;
- *quality_layer* – to calculate quality layers for cloud, water, and snow;
- *energy_balance* – calculation of the radiation and heat fluxes, leaf area index, fraction photosynthesis, etc.;
- *log* – logging facilities module.

A module has inputs (L1C, L2A products, sub products, etc.) and the estimated data are stored in the appropriate class (the L1C/L2A product, sub-product, metadata, etc.). Any processing module can be extended by the developer or a new module included in the chain and run. Each module has test functionality for a self-check. The input and reference data are stored and used.

The *rt* module has an option for the radiative transfer parameter set calculation: use of lookup tables (LUT) or high resolution database (MODTRAN 5.4.0 based; fast, for operational run), or a run of a radiative transfer code (time consuming, for research to be done in 2017). A set of predefined atmospheres and aerosol types (rural, urban, maritime, desert) for the pressure/temperature/humidity profiles, refraction, etc. are handled for the retrieval of the atmospheric properties. The in-house radiative transfer code developed in DLR [9] is planned to be integrated in 2017.

The data and metadata are stored in the following classes:

- *l1c_product* – the input (L1C data) for the atmospheric correction;
- *l2a_product* – the final result of the atmospheric correction (L2A) containing the bottom of atmosphere (BOA) surface reflectance/albedo, estimated aerosol optical thickness, water vapor, and classification maps;
- *ele_variables* – the class defining the sub-products and metadata related to elevation map and topography properties;
- *mask_lists* – class defining masks for the land and atmosphere classes, etc.;
- *rt_variables* – defines the optical properties of the atmosphere calculated by a radiative transfer code or taken from a LUT.

The *sensor* class defines the properties specific only for a particular sensor such as the time of acquisition, wavelength grid, physical properties, and the properties relevant for all optical instruments such as visibility/AOT grids, thresholds, correlations, atmosphere properties, etc.

A special attention should be paid to the class names *scenario*. This class defines the specific workflow of the processing chain with regard to input L1C data. Since a processing module defines a specific analysis of the data and the physical process is independent of the data structure the scenario is in responsibility to prepare the data and appropriately apply the module. The scenario has to assume one of several ways of the L1C data analysis (small/wide field of view, large data volume (Hyspex instrument), different GSD for bands (Sentinel 2), several data cubes, or in the far future – multitemporal data).

Introduction of a new sensor requires only two steps: a) preparation of the LUT (automatic) and writing the scenario. The programming language of the new system is Python 2,7 with the standard libraries GDAL, numpy, scipy. PyCharm was employed as the integrated development environment.

CHALLENGES ENCOUNTERED

It is to be noted that the IDL and Python are too different development systems and are not comparable (IDL is a vector processing oriented system). Nevertheless, the issues found with the use of Python can be interesting for the reader.

Programming functions and primitives produced unexpected differences in the results. With the difference reasoned by «unknown unknowns» the ATCOR system transfer required almost each programming function/primitive to be tested, validated, and runtime analyzed. Double check of the programming functions/primitives between the IDL (ATCOR) and Python (new system) implementation allowed localizing such issues. Stricter typization in IDL is an advantage to do more accurate development.

Several groups of these issues can be outlined:

1. Interpolation accuracy. The IDL and Python linear interpolation function produce different results leading to a selection of an erroneous value and error propagation.
2. N-D interpolation time. The N-dimensional interpolation in IDL is one order faster than in Python.
3. Regression calculation time. Linear regression is employed in AOT and water vapor estimation modules increasing their runtime in Python.
4. Function call time. Each function irrespectively of the complexity requires time to be called. Here, the IDL function call is two times faster.
5. Matrix operations time. These operations are slow in Python resulting in a runtime increase of the smile mode.
6. IDL functions implementation in Python (i.e. BOXCAR filter). There are specific IDL functions not implemented in Python. Reimplementation and run in Python is not so efficient compared to IDL.
7. Convolution time. Convolutional operations are two orders faster in IDL.
8. XDR data load and unpack time. This operation is not efficient in Python and a load of LUTs is several orders slower in Python.
9. Automatic N-D data tiling and process threading on CPU cores. This is the main feature of the IDL making the runtime of an IDL program much faster. Unfortunately, Python requires very careful design of processes threading.
10. IDL efficient swapping. IDL never stops with a high load of a data, while Python throws exception related to memory issues and stops.

These issues made new system two times slower for the standard and 5 times slower for smile atmospheric correction comparing to the IDL implementation. Nevertheless, the requirements on the runtime are fulfilled. The accuracy of the derived L2A products is 1-to-1 to the IDL ATCOR given ground truth data.

CONCLUDING REMARKS

The atmospheric correction system is migrated into Python, validated, and run in the DLR ground segment. As a chain it is integrated into DLR satellite data production as the L2A processor. The new system is employed for the Sentinel 2, Landsat 8, DESIS, and EnMAP L2A data production.

Current development activities are on the comparison of atmospheric correction codes (CEOS-WGCV Atmospheric Correction Inter-comparison Exercise). Current research activities are on the new methods for fully automatic shadow compensation in optical data. In the following next years the new modules are to be developed and integrated: atmospheric correction for airplane mode, correction over water/Ångström estimator.

LITERATURE

1. Richter R. Correction of satellite imagery over mountainous terrain. Applied Optics. 1998. Vol. 37. P. 4004–4015.
2. Richter R., Schläpfer D. Geo-atmospheric processing of airborne imaging spectrometry data. Part 2: atmospheric/topographic correction // J. Remote Sensing. 2002. Vol. 23. P. 2631–2649.
3. Interactive Data Language IDL. Exelis Visual Information Solutions. Inc. a subsidiary of Harris Corporation. http://www.harrisgeospatial.com. 2016.
4. Richter R., Wang X., Bachmann M., Schläpfer D. Correction of cirrus effects in Sentinel-2 type of imagery // International J. of Remote Sensing. 2011. Vol. 32. № 10. P. 2931–2941.
5. Makarau A., Richter R., Müller R., Reinartz P. «Haze Detection and Removal in Remotely Sensed Multispectral Imagery» IEEE Transactions on Geoscience and Remote Sensing 2014. Vol. 52. № 9. P. 5895–5905.
6. Richter R., Heege T., Kiselev V., Schläpfer D. Correction of ozone influence on TOA radiance //International J. of Remote Sensing. 2014. Vol. 35. № 23. P. 8044–8056.
7. Schläpfer D., Borel C., Keller J., Itten K. Atmospheric Precorrected Differential Absorption Technique to Retrieve Columnar Water Vapor. Remote Sensing of Environment. 1998. Vol. 65. № 3. P. 353–366.
8. Richter R., Schläpfer D., Müller A. Operational atmospheric correction for imaging spectrometers accounting for the smile effect. IEEE Transactions on Geoscience and Remote Sensing. 2011. Vol. 49. № 5. P. 1772–1780.
9. Efremenko D., Doicu A., Loyola D., Trautmann T. Acceleration techniques for the discrete ordinate method. J. Quant Spectrosc Radiat Transfer 2013. Vol. 114. P. 73–81.