

Technical Note

A Generic Framework for Using Multi-Dimensional Earth Observation Data in GIS

Yunfeng Jiang, Min Sun and Chaowei Yang *

Department of Geography and Geoinformation Science, George Mason University, Fairfax, VA 22030, USA; yjiang7@gmu.com (Y.J.); msun@gmu.edu (M.S.)

* Correspondence: cyang3@gmu.edu; Tel.: +1-703-993-4742

Academic Editors: Markus Neteler, Ioannis Gitas and Prasad S. Thenkabail

Received: 11 March 2016; Accepted: 28 April 2016; Published: 4 May 2016

Abstract: Earth Observation (EO) data are critical for many Geographic Information System (GIS)-based decision support systems to provide factual information. However, it is challenging for GIS to understand traditional EO data formats (e.g., Hierarchical Data Format (HDF)) given the different contents and formats in the two domains. To address this gap between EO data and GIS, the barriers and strategies of integrating various types of EO data with GIS are explored, especially with the popular Geospatial Data Abstraction Library (GDAL) that is used by many GISs to access EO data. The research investigates four key technical aspects: (i) designing a generic plug-in framework for consuming different types of EO data; (ii) implementing the framework to fix the errors in GIS when using GDAL to understand EO data; and (iii) developing extension for commercial and open source GIS (*i.e.*, ArcGIS and QGIS) to demonstrate the usability of the proposed framework and its implementation in GDAL. A series of EO data products collected from NASA's Atmospheric Scientific Data Center (ASDC) are used in the tests and the results prove the proposed framework is efficient to solve different problems in interpreting EO data without compromising their original content.

Keywords: Earth Observation data; data access; GDAL; GIS; plugin framework; QGIS

1. Introduction

Earth Observing System (EOS) produces big remote sensing data, which record and capture long-term facts of land surface, solid earth, atmosphere and oceans. In recent years, the importance of EO data has been widely acknowledged by both government and science communities for Earth system science research and a variety of applications, including disaster response, environmental planning, global change, insurance, and private investment [1]. Geographic Information Systems (GISs) are widely used to interrelate multiple types of information assembled from a variety of Earth Observation (EO) data and to visualize, query, overlay, and analyze these data to understand relationship, patterns, trend for a wide range of scientific, academic and private entities [2]. Therefore, EOS agencies, such as National Aeronautics and Space Administration (NASA), have increasingly used GIS, e.g., ArcGIS and QGIS (formerly also known as Quantum GIS), to analyze EO data to discover patterns and knowledge about the Earth's surface.

However, over the past decades, different formats and standards have been developed to organize and store EO data in format highly tailored for different applications and organizations. Many are in old formats, and specialized geospatial tools are required. This makes it difficult or impossible to incorporate EO data into GIS, precluding their visualization or analysis [3–5]. On the other hand, most main-stream GIS cannot process and utilize generic EO data, and there are unexpected issues or errors while importing EO data files. To reconcile the conflicts between EO data and GIS, initiatives have been made for developing a generic methodology to solve EO data compatibility in GIS. For example,

the Hierarchical Data Format (HDF) group proposed a comprehensive methodology to better support EO data in HDF formats. Unfortunately, no effective solution has emerged to solve this problem.

Among the open source software or tools capable of accessing EO data, Geospatial Data Abstraction Library (GDAL) is widely used for supporting GIS to access and process raster and vector data [6]. For example, ArcGIS relies on GDAL to read EO data in Hierarchical Data Format (HDF), the primary data format of EO data. Unfortunately, GDAL has some problems dealing with HDF data, especially multiple dimensional data (e.g., 3D, 4D, 5D) [7]. Therefore, ArcGIS supports only a fraction of NASA Hierarchical Data Format-Earth Observing System (HDF-EOS), which is the standard format to store data collected from three EOS satellites, including Terra, Aqua and Aura. In addition, limitations often occur while importing EO data into GIS [8,9]. For example, it is not uncommon to miss spatial reference, fail to retrieve NoData value (absence of a recorded value), and misunderstand multi-dimensional variables. Recently, some GIS software vendors worked on fixing these problems by developing specific versions of GDAL to assimilate more EO data. For example, Esri, the vendor of ArcGIS, developed Esri-specific GDAL to enhance the interpretation of EO data in ArcGIS Desktop 10.3. However, due to the heterogeneity of data products from different data centers, many limitations remain thus impeding the usage of EO data products. Furthermore, integrating the fixes in a particular branch of GDAL (e.g., Esri-specific version) into the main GDAL open-source trunk is time-consuming and labor intensive. Ideally, these issues can be corrected in satellite mission operations, but it is too expensive, most times impossible, to revise an operational mission. Therefore, it remains a challenge to provide a functional linkage between GDAL and GIS.

An XML-based plug-in framework is proposed to address the problem of using GDAL to access EO data. The XML file specifies the particular problems occurring in different data products and, accordingly, invokes a series of related functions to fix the problems. Specifically, the proposed framework rotates an image by 90 degree, inverts an image upside down, or interprets 3D/4D/5D variables. To demonstrate its functionality, GIS extensions are developed based on the proposed framework. Unlike traditional EOS geospatial tools that only consume partial EO data, the proposed framework enables GIS to support more EO data. Additionally, the framework is flexible and extendable allowing users to manually add a new function for coping with new problems as well as new EO data products. The HDF4/HDF5 data drivers in GDAL (version 2.0.0) are also improved to address the above limitations and to enhance the capability of processing multi-dimensional variables at the level of source code. The enhanced GDAL allows GIS developers to create new modules for consuming different EO data in their GIS applications.

This paper reports research in seven sections. This first section introduces the study by analyzing challenges, potential solutions, and proposed research. Section 2 reviews related work (e.g., EO data and its major formats), tools and libraries for parsing data, problems while using GIS to process data, and relevant research on addressing these problems. Section 3 proposes a generic XML-based plug-in framework to address the problems of incorporating GIS and EO data. Section 4 presents details of system implementations and optimizations. Section 5 demonstrates the efficiency of the proposed framework and corresponding tools using case studies. Section 6 discusses the contributions of the study, and Section 7 summarizes the research and discusses future research directions.

2. Materials and Related Work

This chapter introduces EO data and its major formats, tools and libraries for parsing EO data, and the problems of using GIS to process the data. It also reviews relevant research over the past decades.

2.1. EO Data Format, Tools, and Related Work

Since 1999, NASA's Earth Science Division (ESD) has launched more than 20 satellites to help develop a scientific understanding of the Earth system and its response to natural and human-induced changes and to enable improved prediction of climate, weather, and natural hazards [10]. Satellite instruments or sensors are accommodated on these flagship EOS satellites (Aqua [11], Terra [12],

Aura [13]). These instruments are named according to the satellite or platform and the capabilities of the sensor or instrument. Each instrument has its own specialized function and produces different types of data for various applications. For example, the instrument, Measurement of Pollution in the Troposphere (MOPITT), captures the lower atmosphere, focusing on the distribution, transport, source, and sinks of carbon monoxide in the troposphere, and captures how carbon monoxide interacts with land and ocean biospheres [13,14].

Different instruments or sensors produce different EO data and are usually consistent in data format. The standard format for NASA EO data is HDF, a multi-objects-based format originated from the National Center for Supercomputing Applications (NCSA) and updated by the HDF Group [2], non-profit organization. There are two distinct varieties of HDF, HDF4 (version 4 and earlier) and the latest HDF5 [15]. Because many Earth science data require geo-location, the HDF group developed the HDF-EOS format with additional conventions and data types for HDF files based on HDF format. HDF-EOS (Hierarchical Data Format-Earth Observing System) is a self-describing file format for transferring various types of data among different machines based on HDF. For example, NASA uses HDF-EOS as the primary data format for EO data, storing data from EO satellites (*i.e.*, Terra, Aqua and Aura), supporting four geospatial data types (grid, point, zonal, and swath), and providing uniform access to diverse data types in geospatial context. The HDF-EOS stores and organizes large volumes of numeric data and is supported by many commercial and non-commercial software platforms [16].

To make EO data usable to the Earth science research communities and the general public, a variety of tools handling EO data have been developed to easily compare, analyze, and visualize data from a variety of EO data sources. Some are general tools that allow users to browse and edit general HDF-EOS data irrespective of their origin (e.g., HDF Viewer developed by the HDF group for supporting HDF4 and HDF5 format [15]). Others are designed for specific EO data product (e.g., Advanced Infrared Sounder (AIRS) tool that is only available to manipulate AIRS data files). In general, both tools work with EO data well, allowing users to manipulate and visualize EO data. But, only a limited number of functions are provided by the two tools (e.g., reformatting, re-projection, mosaicking, data quality assessment, image processing, multispectral analysis). In addition, with growing demands of EO data, a single pure EO data tool is insufficient for many research initiatives that require analysis of multi-disciplinary and multi-source data. Therefore, scientists and engineers resort to GIS tools for helping using EO data.

The past decades witnessed the rapid development of GIS, provides much more powerful functions to visualize and analyze EO data than traditional HDF tools. Users prefer GIS tools for the following reasons: (1) GIS interrelates multiple types of information assembled from a diversity of data sources and formats to visualize, query, overlay, and analyze data; (2) GIS provides more advanced functions for spatial data in managing, mapping, modeling and decision supporting. However, the value of EO data in GIS applications has not been fully explored due to a series of limitations in data integration. Fortunately, some new initiatives are developing a common methodology to address this shortcoming (e.g., facilitating public access to EO data) [16]. NASA funded to the development of a NASA HDF-EOS Web GIS Software Suite (NWGISS) to provide standards-based access and services to NASA EO data for the GIS community according to Open Geospatial Consortium specifications [17–19]. The project integrates with Grid technology for sharing data and computing resources among NASA data centers and implements geospatial web services. Qin *et al.* [20] provided an efficient solution to access geospatial data with the parallel raster Input/Output model of using GDAL on the basis of Message Passing Interface.

2.2. Common Challenges when Using EO Data in GIS

A variety of data products from Atmospheric Science Data Center (ASDC) (e.g., MOPITT, Tropospheric Emission Spectrometer (TES), Clouds and Earth's Radiant Energy System (CERES), Multi-angle Imaging SpectroRadiometer (MISR)) are stored in HDF4, HDF5 or NetCDF formats. The MOPITT level 3 version 5 (HDF4), MOPITT level 3 version 6 (in HDF5) and TES level 3 (HDF5) data products are selected for case studies. The investigation reveals a variety of limitations when using datasets in GIS. The major findings of this research are as follows:

- Problem 1: 90 degrees rotated and inverted upside down image

Both problems happened at the pixel level when visualizing EO data. Specifically, the images from EO data are rotated 90° or inverted upside down (Figure 1). The explanation for the 90° rotation are three folds: (1) dimension size in metadata is erroneously switched; (2) pixel values are transcribed incorrectly; and (3) GDAL (version 2.0.0) included in ArcGIS returns a null value of dimension size because the metadata field is interpreted as a “band”, whereas the field in the HDF file is “nprs”, “nlon”, and “nlat”.

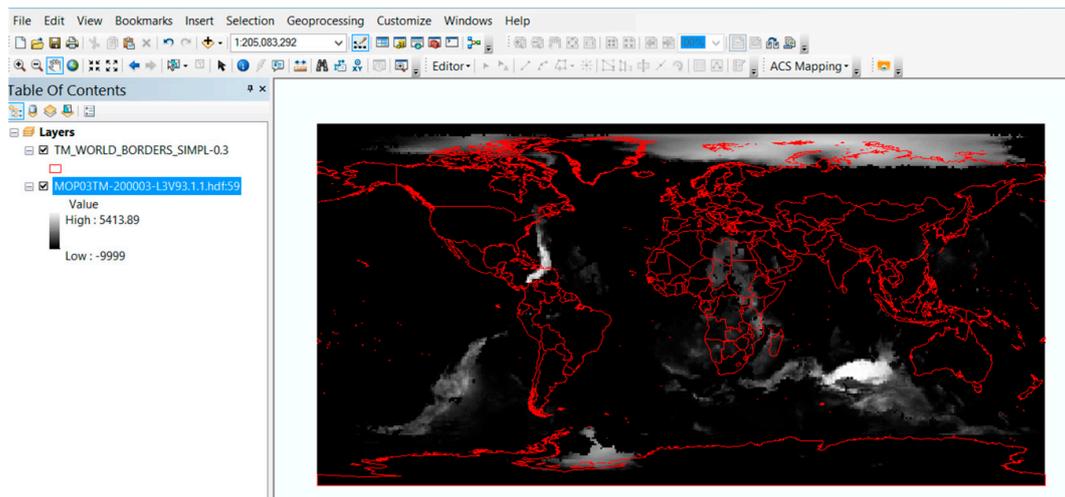


Figure 1. Rotated Earth Observation (EO) data in ArcGIS.

- Problem 2: Missing geo-reference information

Geo-reference or information related to geographic location is commonly used in GIS. The EO data have geo-reference information and are expected to be extracted by GIS. However, such information is sometimes missing. Research reveals that GDAL does not support all HDF data products and will result in errors in geo-reference information. Therefore, GIS fails to recognize the HDF spatial information (Figure 2).

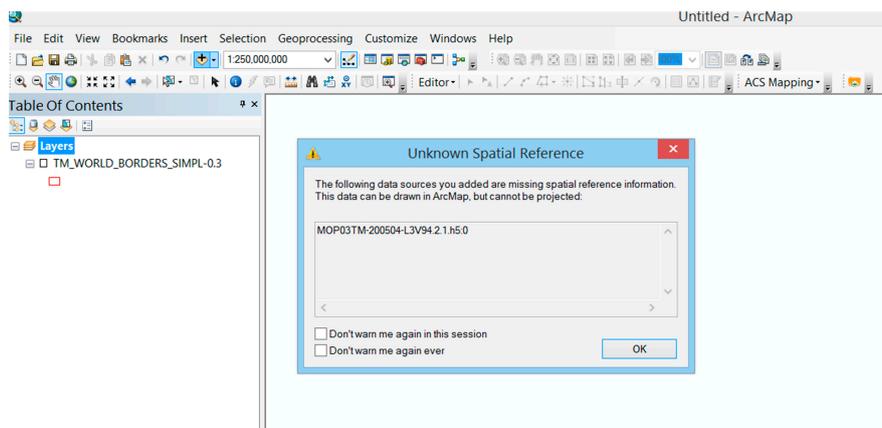


Figure 2. A pop-up message showing ArcGIS cannot interpret the geo-reference information of data.

- Problem 3: 3D subsets cannot be interpreted correctly

Most GISs are unable to correctly display 3D or 3D+ EO data. In some instances for HDF data, the dimension size is incorrectly transcribed in order. For example, in MOPPIT level 3, one subset

should have 9 bands with each in 360×180 (width \times height) format. However, the data record for the dimension information is $360 \times 180 \times 9$, or transcribed as $9 \times 360 \times 180$ (width \times height) images. This explains the observed GIS display of a very narrow rectangular image with gray color. In another instance, GDAL returns null value for the dimension size because the metadata field that GDAL reads is “band” whereas the field in HDF file stores that information in “nprs”, “nlon”, and “nlat”. A third instance is one in which the HDF driver does not support reading 5D subsets. For 4D subsets, the driver cannot recognize the size of each dimension correctly. The final instance is that mainstream GIS does not support the display of multiple-dimensional (4D and 5D) HDF datasets. For example, some GIS use the first three bands as color channels (e.g., Red, Green, and Blue) (Figure 3). Furthermore, GDAL is often used in GIS for processing a variety of vector and raster data formats, but does not support 4D or 5D datasets.

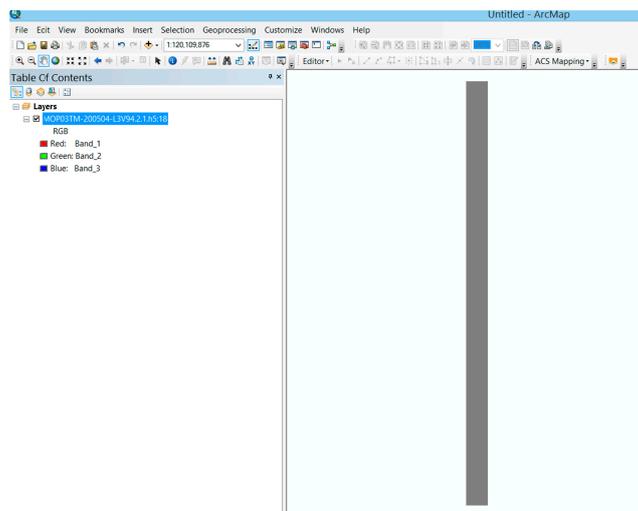


Figure 3. Display error for the 3D EO data in ArcGIS.

- Problem 4: Missing metadata

Missing useful metadata (e.g., NoData value, a mask for representing the absence of data) is more likely to produce an inappropriate image or lead to crash when displaying HDF datasets in ArcGIS (Figures 4 and 5). Two errors are responsible for this limitation: metadata are not filled in at the initial stage or the metadata exist in the physical data file, but ArcGIS or GDAL fails to interpret these data appropriately.



Figure 4. ArcGIS crashes when reading data with misinterpretation on metadata.

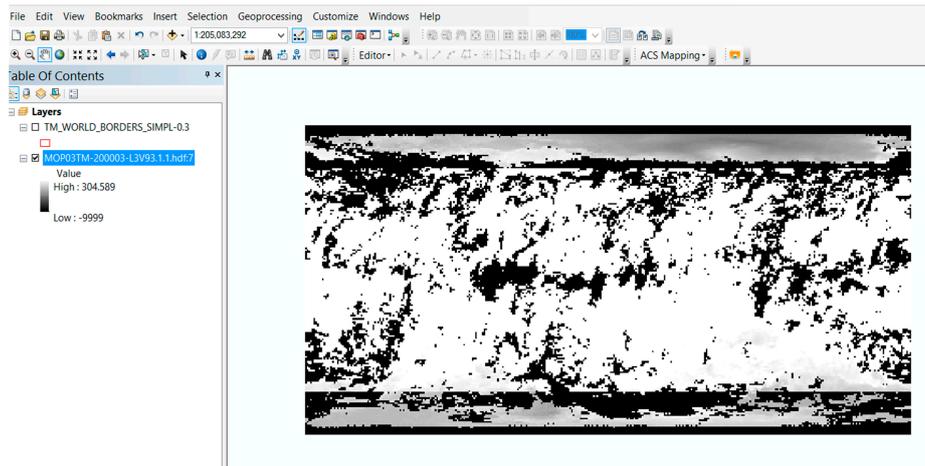


Figure 5. Image displayed by interpreting a NoData value (−9999) as a normal value.

3. Methods

Some of the problems described in Section 2.2 happen in more than one data products while each data product could have its unique problem. Accordingly, we propose a generic plug-in framework to fix the problems for facilitating the consumption of EO data in GIS. The generic part of the framework is used to solve the common problems and the plug-ins are able to handle the particular unique ones in a specific data product. The framework allows users to detect and repair problems and interpret the variables of EO data correctly. The plug-in framework is composed with four primary layers logically: GDAL/HDF, function, plug-in, and GIS extension (Figure 6).

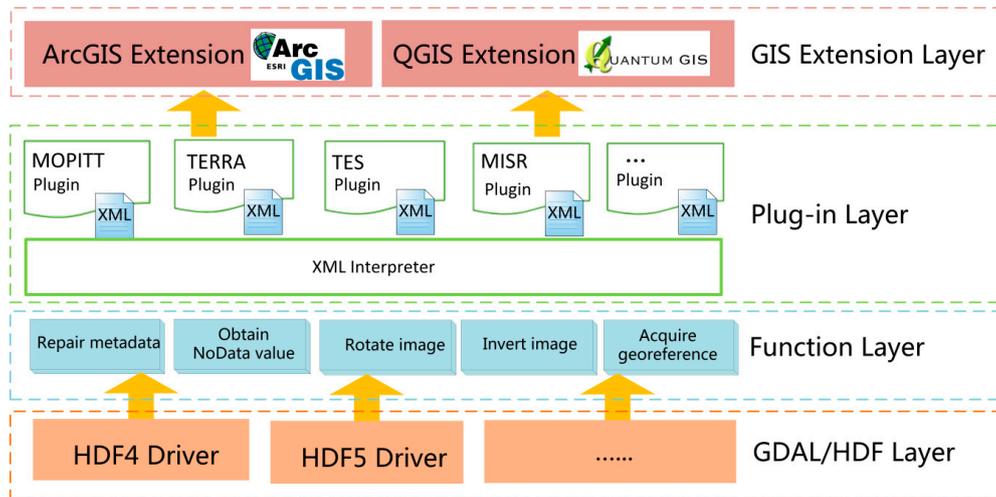


Figure 6. A generic XML-based plugin framework.

3.1. GDAL/HDF Layer

Most EO data are stored in HDF format, which is widely used in scientific communities. In this layer, limitations associated with HDF data in GIS are addressed by focusing on HDF drivers in GDAL. This layer is placed at the lowest level of the framework (Figure 6) but plays a critical role in accessing EO data as HDF data drivers are designed to handle various sets of HDF data. In this layer, we revised the source code of GDAL HDF data driver to overcome some internal issues and enhance its ability to support multi-dimensional data. Therefore, the HDF4 and HDF5 drivers are optimized to parse the HDF dataset, especially for those that are multi-dimensional.

3.2. Function Layer

To address limitations when using EO data in GIS, a series of functions are designed in the function layer on top of the GDAL/HDF layer (Figure 6). Generally, one function corresponds to one problem. For example, the output image from MOPPIT level 3 (HDF4) data is rotated by 90 degree. The Rotate90degree function is designed for rotating one 2D image as input by 90 degrees. Each function is applied on all data products with the same problem. At this stage, a number of functions are developed as follows: Rotate90Degree; InvertUpsideDown; OpenXmlFile; Get3DDimension; Get4DDimension; Get5DDimension; GetNoDataValue; and GetGeoreference. As new limitations emerge, additional functions can be developed. In the study, all functions are categorized into four groups according to their functionality, including metadata, image, interpreting, and georeference functions. Each of these groups is discussed below.

3.2.1. Metadata Functions

Metadata is data describing data and provide some basic information about EO data. Incomplete metadata or missing metadata lead to unexpected results. For example, any GIS will fail to open data files if the data type is not recognized. To avoid similar problems, a series of metadata functions retrieve useful user-defined information from metadata, including recognizing data types of image pixel value, getting image dimension, obtaining NoData values, or acquiring the time period.

3.2.2. Image Displaying Functions

Occasionally, ArcGIS displays EO data as a raster image inappropriately due to the inappropriate organization of the pixel value. The investigation of all test EO data products revealed the two most common problems are the image being rotated by 90 degrees or inverted upside down. Such problems were attributed to the image being written into the HDF files. Therefore, two corresponding functions are developed: Rotate90Degree; and InvertUpsideDown.

3.2.3. Interpreting Functions

Most EO data products store a multiple dimensional variable comprised of more than three bands. Nevertheless, many GISs fail to process and appropriately display such datasets to the source of the limitation is the third party library, GDAL, which is unable to fully support these data. For example, when opening a MOPPIT level 3 data with one variable nested in nine raster bands (e.g., temperature), ArcGIS treats the first three raster bands as color bands (RGB) and ignores the other six raster bands. The solution to this problem is to develop related functions targeted for correctly interpreting multiple dimensional EOS datasets, including 3D, 4D, and 5D.

3.2.4. Spatial Reference Functions

Different from traditional scientific data, EO data are featured with spatial reference information. The GIS uses the reference to integrate different layers from different data sources resulting in a thematic map. Without spatial reference information, GIS is unable to integrate, combine or map EO data in GIS. Examining the output EO data in GIS, the geolocation information is easy to lose. To better process EO data in GIS, a spatial reference function acquires the spatial reference of each EO data.

3.3. Plug-In Layer

The Plug-in layer is the most critical of the generic framework; as it is the bridge connecting the function and GIS extension layers. When accessing and displaying EO data in GIS, each type of data may have their own limitation (Section 2.2). To solve these issues, the plug-in calls their corresponding types of functions in the function layer (Section 3.2). The plugin user assigns the repairing functions needed by a data product according to its limitation. In this layer, each data product is defined as a plugin associated with an XML file defining a series of problems. For example, the XML of MOPPIT

data plugin includes three functions: rotate 90 degrees; interpret multiple dimensional datasets; and obtain NoData value. Through parsing the XML, GIS calls the user-defined functions to fix the problems of corresponding data product files in GIS. Since the EO data provider is familiar with the data model, they create the XML plug-ins for EO data user. The plug-in mode facilitates users to add one new data product into the GIS applications by adding a plug-in in lieu of extra development and assuming existing functions meet the demands.

3.4. GIS Layer

The GIS extension layer serves end users of the EO data. The layer is developed on the basis of the plug-in layer. The GIS extension includes specialized GIS tools for enhanced productivity and analysis. Many GIS software offer a range of optional extension that dramatically expands the capabilities of GIS, e.g., ArcGIS (ESRI 2016) and QGIS (QGIS 2016). The proposed GIS extension better supports raster data in the form of HDF files which are the most common format for storing EO data. Different from GIS's build-in functions of loading HDF data, the proposed GIS extension appropriately displays EO data associated with an XML file as a plug-in. Two major GIS extensions are customized based on enhanced GDAL to support HDF data, including ArcGIS and QGIS, the most popular GISs.

4. Results

Section 3 provides an overview the generic XML-base plugin framework. This section introduces the development environment and implementation of the framework.

4.1. Development Environment

Since both commercial and open source GIS mostly utilize GDAL to access EO data, we implement our framework in GDAL. The development compatibility and extensibility was evaluated by adopting the standard programming API of the open source GDAL. The entire framework is implemented using C++ programming language in Microsoft Studio on the basis of the optimized GDAL, being easy to translate into other programming languages (e.g., Java). Later on, for demonstrating purpose, we also built the ArcGIS extension using C# programming language with Microsoft Studio, which is installable on machines equipped with ArcGIS software, and QGIS extension with C++ programming language in Qt environment.

4.2. GDAL Enhancement

Section 2.2 describes technical problems (e.g., missing NoData value), which are more likely to be caused by the GIS's third party library, GDAL, instead of GIS software itself. To overcome this kind of problems, the GDAL's source code is optimized based on the version of GDAL (2.0.0) at time of developing the improved driver and compiled on the official released HDF4 (4.2.6) and HDF5 (1.8.7) libraries from HDF group. The development results will improve GDAL and enhance the GDAL capability for accessing HDF4/HDF5 data, including gaining the correct dimensional information of HDF variables and bands and extracting geo-reference information. In addition, the performance of opening and closing HDF data is enhanced by fixing some GDAL internal defects. For example, the procedure of freeing sub-dataset is time consuming, due to the release of each dataset's Ground Control Points (GCPs) in an unreasonable way. The new GDAL's HDF driver is more stable and powerful in accessing HDF data.

4.3. XML-Based Plug-In

A number of functions are developed to address limitations when using EO data in GIS. Since different EO data have different issues in GIS, it is hard to organize and manage functions to repair these specific problems. For example, one data product may need fewer functions to remedy the issue. Therefore, the proposed strategy addresses the issue using an XML-based plug-in to choose the appropriate functions for EO data product. An example XML of configuring functions is as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<ProductPlugin productType="MOPPIT3" productFormat="HDF4">
  <DataAccessProblem>Description Problems</DataAccessProblem>
  <Correction>
    <Function>InvertImageUpsideDown</Function>
  </Correction>
  <Correction>
    <Function>AddFilledValue</Function>
  </Correction>
  <Correction>
    <Function>Display3Dimension</Function>
    <RasterXDim>2</RasterXDim>
    <RasterYDim>1</RasterYDim>
    <RasterBandDim>0</RasterBandDim>
  </Correction>
  <Correction>
    <Function>Display4Dimension</Function>
    <RasterXDim>3</RasterXDim>
    <RasterYDim>2</RasterYDim>
    <RasterBandDim>1</RasterBandDim>
    <Raster4Dim>0</Raster4Dim>
  </Correction>
  <Correction>
    <Function>Display5Dimension</Function>
    <RasterXDim>4</RasterXDim>
    <RasterYDim>3</RasterYDim>
    <RasterBandDim>2</RasterBandDim>
    <Raster4Dim>1</Raster4Dim>
    <Raster5Dim>0</Raster5Dim>
  </Correction>
</ProductPlugin><?xml version="1.0" encoding="utf-8"?>
<ProductPlugin productType="MOPPIT3" productFormat="HDF4">
  <DataAccessProblem>Description Problems</DataAccessProblem>
  <Correction>
    <Function>InvertImageUpsideDown</Function>
  </Correction>
  <Correction>
    <Function>AddFilledValue</Function>
  </Correction>
  <Correction>
    <Function>Display3Dimension</Function>
    <RasterXDim>2</RasterXDim>
    <RasterYDim>1</RasterYDim>
    <RasterBandDim>0</RasterBandDim>
  </Correction>
  <Correction>
    <Function>Display4Dimension</Function>

```

```

<RasterXDim>3</RasterXDim>
<RasterYDim>2</RasterYDim>
<RasterBandDim>1</RasterBandDim>
<Raster4Dim>0</Raster4Dim>
</Correction>
<Correction>
<Function>Display5Dimension</Function>
<RasterXDim>4</RasterXDim>
<RasterYDim>3</RasterYDim>
<RasterBandDim>2</RasterBandDim>
<Raster4Dim>1</Raster4Dim>
<Raster5Dim>0</Raster5Dim>
</Correction>
</ProductPlugin>

```

The plug-in XML files store the type and format of EO data product, and the problems encountered and functions (in Correction tag) needed to be resolved. The XML is usually configured by data provider in advance. Figure 7 shows the information flow of producing an XML-based plugin. Since the parser is integrated into the developed GIS extension, the GIS will precisely call the pre-defined functions to fix the issues of the selected data product associated with such similar XML files. The end user can just put the XML file and data file under the same file path on a storage.

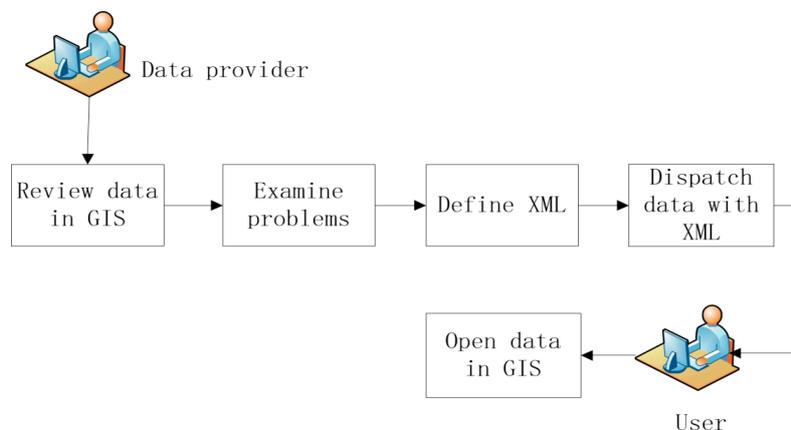


Figure 7. Workflow of defining and using XML plugin.

4.4. Workflow of Processing HDF Dataset

The workflow of processing an EOS HDF data using ArcGIS extension is illustrated (Figure 8). The EO data product's type and format are established by the selected file's full name or its metadata. Thereafter, the system automatically scans all files in the current folder where the selected data file resides and searches the associated plugin XML file. If available, the parsing of the XML file establishes a series of predefined functions to solve the issues of the current data product. When processing the data in ArcGIS, these functions operate on all bands of the selection before displaying the data. The customized Graphic User Interface (GUI) of the developed extension allows users to choose the variables and their sub bands. Finally, ArcGIS displays data layers generated from the selected bands by using ArcGIS ArcObjects, the Application Program Interfaces (APIs).

The GUI of selecting HDF sub-datasets when opening HDF data in ArcGIS is shown in (Figure 9). This recognizes the HDF data from the plugin XML file dispatched with the type of data product and retrieves the user-predefined functions shared by all variables and bands. The newly designed GUI allows a user to select multiple bands from the same or different variables for integration into the same workspace and creating multiple corresponding layers simultaneously in ArcGIS.

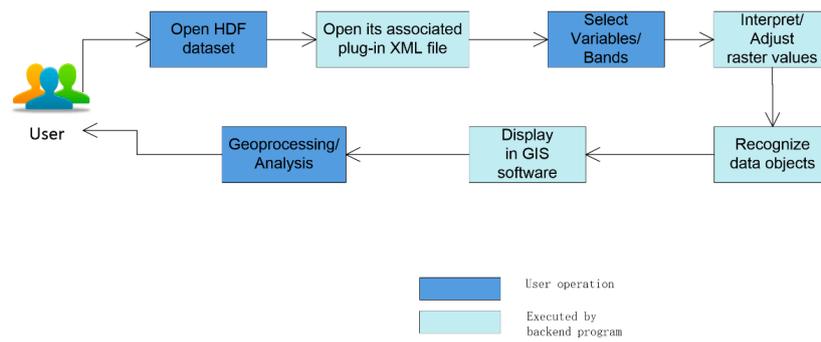
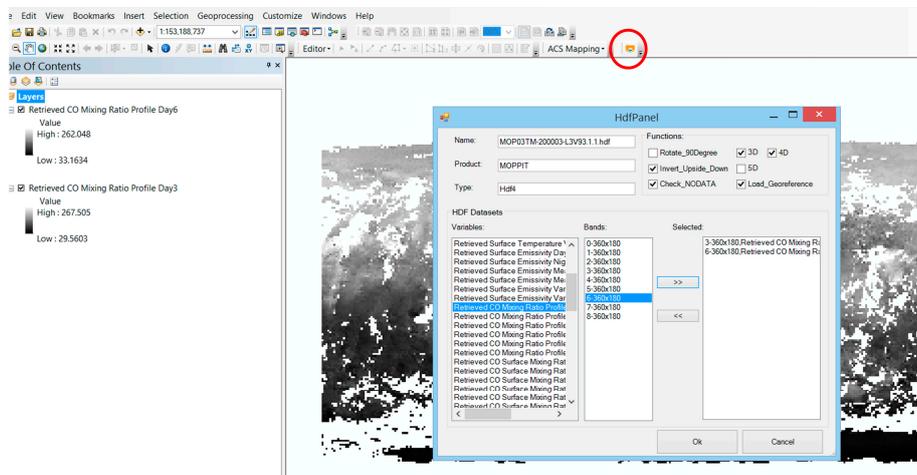
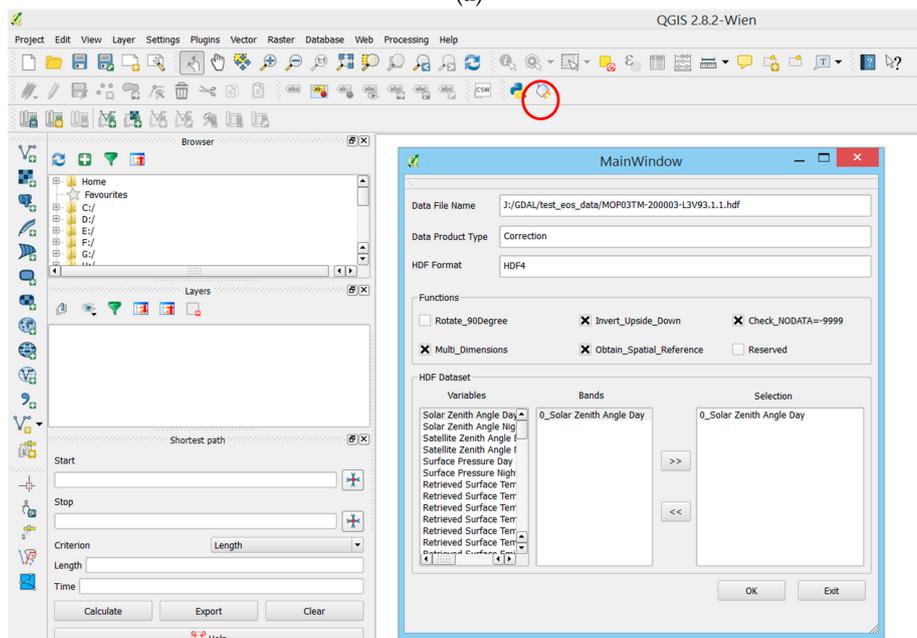


Figure 8. EO data Processing workflow using Geographic Information Systems (GIS) and the plug-in framework.



(a)



(b)

Figure 9. GUIs of ArcGIS (a) and QGIS (b) extensions incorporating the proposed framework.

4.5. Integrate the Framework into GDAL

The initial purpose of developing such a generic framework is to allow GIS developers to integrate it into GIS applications for enhancing their capabilities of processing EO data, especially for these data having problems when they are used in GIS. Ever since, more and more people would like to process these EO data by using GIS tools. On the other hand, it is equally important to those users who would like to directly use GDAL to process the EO data with problems, e.g., transferring the data into different image formats. Therefore, packing the development into GDAL source code is significant to expanding the usage of the EO data. In order to contribute the framework to GDAL, it is necessary to put the implementation into GDAL at a source code level. Figure 10 shows the primary method in which functions are implemented in HDF4 and HDF5 data drivers in GDAL (Version 2.0.0).

From Figure 10, the plug-in framework is incorporated into the source code of original GDAL, specifically in HDF4 and HDF5 data drivers. Within the framework, there are five major components for overcoming the problems that are introduced above. During the implementation, five general components are developed into four classes: HDF4ImageDataset, HDF4RasterBand, HDF5ImageDataset, and HDF5RasterBand. Different from the conventional method of opening a HDF data file, the new GDAL edition requires an XML file to be related with the targeted HDF data file that need to be fixed. As a matter of fact, the XML file helps determine the functions to be invoked when opening the HDF data file using GDAL.

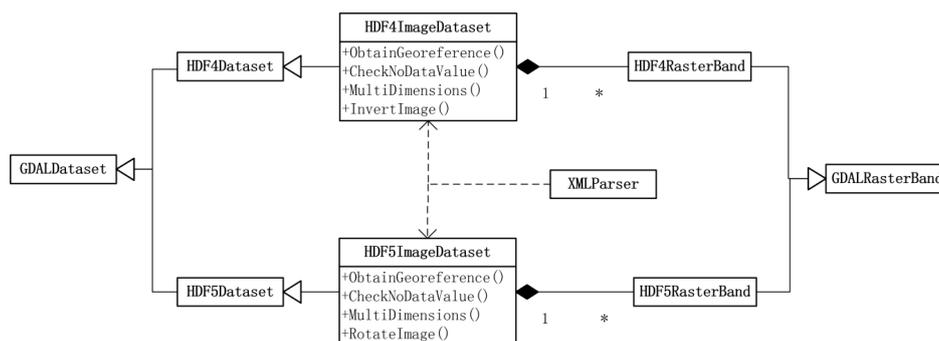


Figure 10. Implementation of the plug-in framework in Geospatial Data Abstraction Library (GDAL) source code. One HDF4/HDF5 image dataset (represented as 1 next to the solid diamond) includes multiple HDF4/HDF5 bands (represented as * oppose to 1).

Therefore, there are two ways to implement the framework. One is to develop a middleware on the basis of GDAL without changing its source code. GIS developer can integrate the library into their applications or produce an extension based on the library. Another is to capsule the framework into GDAL source code, which most of effort have been done in GDAL. Both are effective solutions but contribute to different communities.

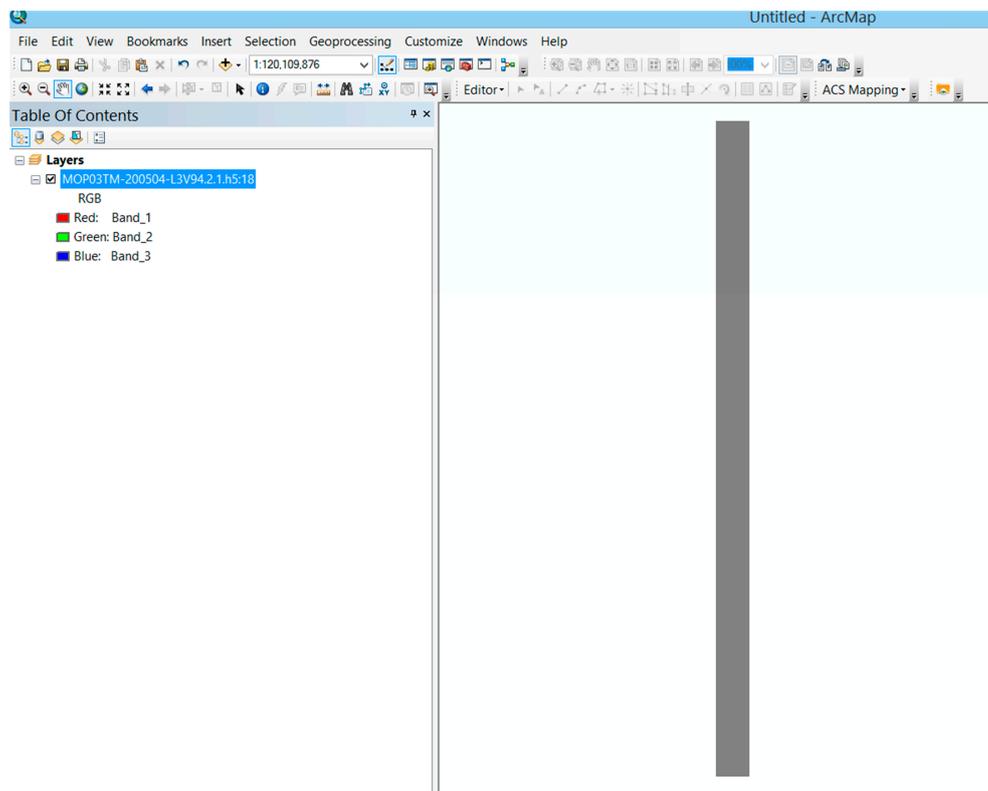
5. Experiments and Discussion

To validate the feasibility and functionality of the proposed plug-in framework, we developed an ArcGIS extension and a QGIS extension to test a diversity of EO data that have been collected at NASA's ASDC. In the study, a set of experiments were conducted in ArcGIS (10.3) to demonstrate the performance of the developed extension, according to the problems presented in the Section 2.2. To compare the difference before and after the improvements, some GIS data were included and assembled onto the generated raster layer with a red color, e.g., world boundary ESRI SHAPE files.

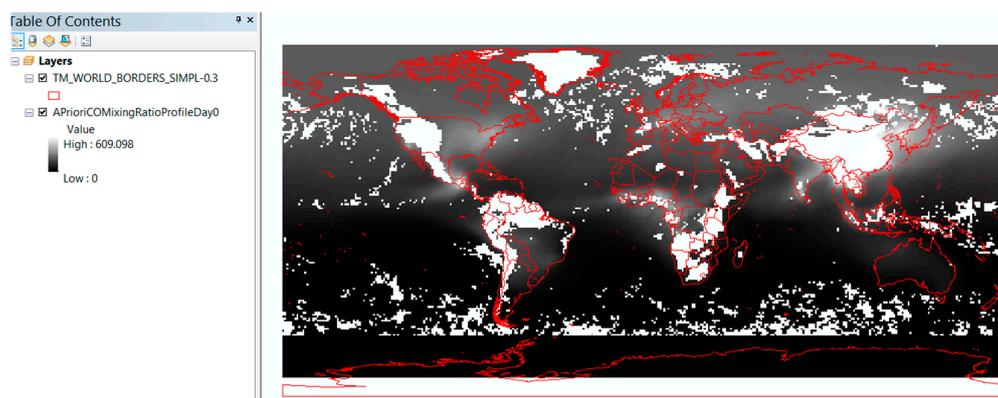
5.1. Case 1: Interpret Multiple Dimension HDF Data

When opening EO data in ArcGIS, the most serious problem is that it cannot interpret and display 3D/4D/5D HDF sub-datasets properly. These types of problems occur to most EO data.

This experiment uses a MOPPIT HDF data that has a 3D sub-dataset (“Retrieved CO Mixing Ratio Profile Day”) as the testing data to show the improvements of the framework and extension developed. Figure 11a shows the displaying result of a selected 3D sub-dataset that is generated in ArcGIS. It is observed that the image is abnormally displayed and its bands are wrongly treated as color band (RGB). Figure 11b shows the correct data displaying that is generated from the first band of the selected 3D HDF sub-dataset after the improvements. The extension has greatly enhanced the capability of ArcGIS in accessing and processing multiple dimensional HDF datasets.



(a)

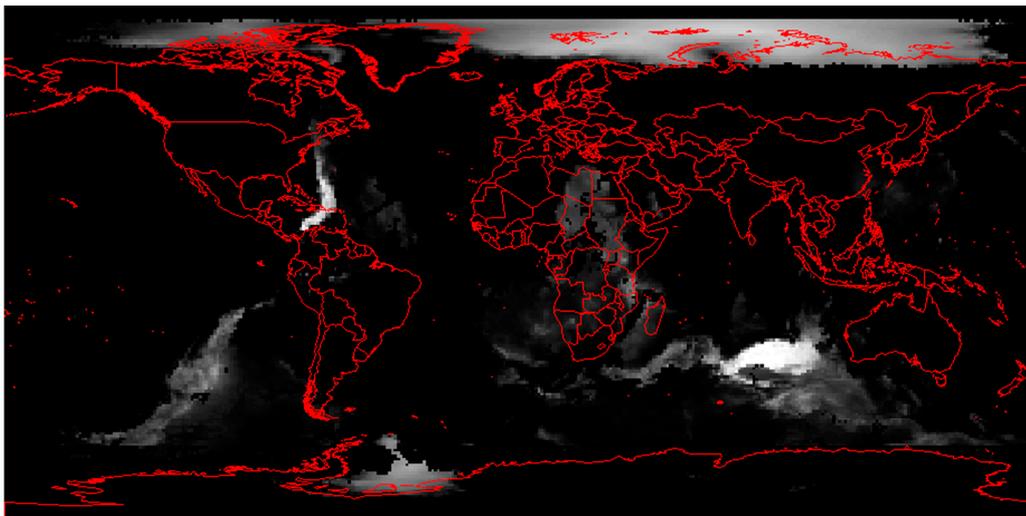


(b)

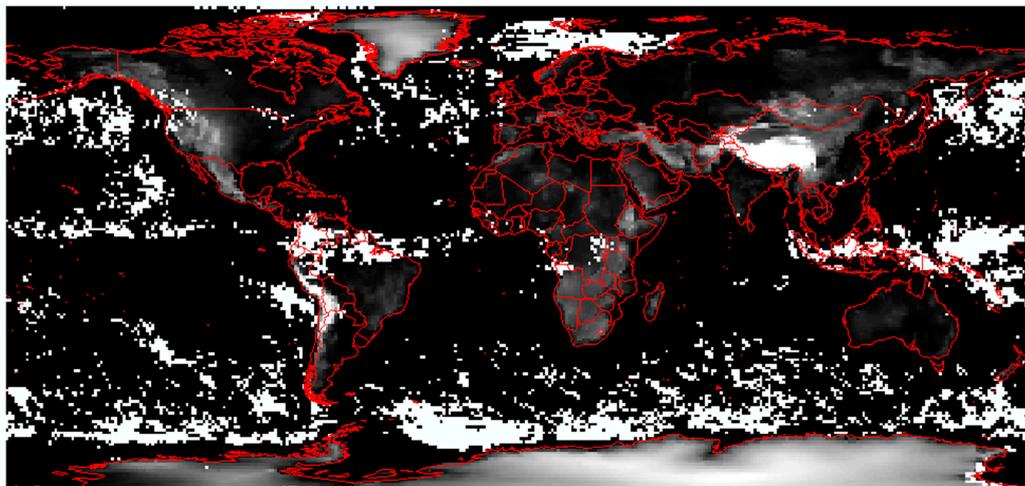
Figure 11. Three-dimensional EO data display in ArcGIS before (a) and after (b) using the enhanced GDAL.

5.2. Case 2: Rectify Image Inverted Problem

When opening some EOS HDF4 data in ArcGIS, e.g., MOPPIT data, it is found that the raster data generated from one of its 2D sub-dataset was inverted upside down. “MOP03TM-200003-L3V93.1.1.hdf” is used as the sample data to show the problem. Figure 12a shows the displaying result of one selected sub-dataset, “DEM Altitude Night” in ArcGIS. Comparing with the overlay of world map boundary, the raster image is inverted. Using the extension and framework, this problem is solved and Figure 12b shows the correct raster image.



(a)

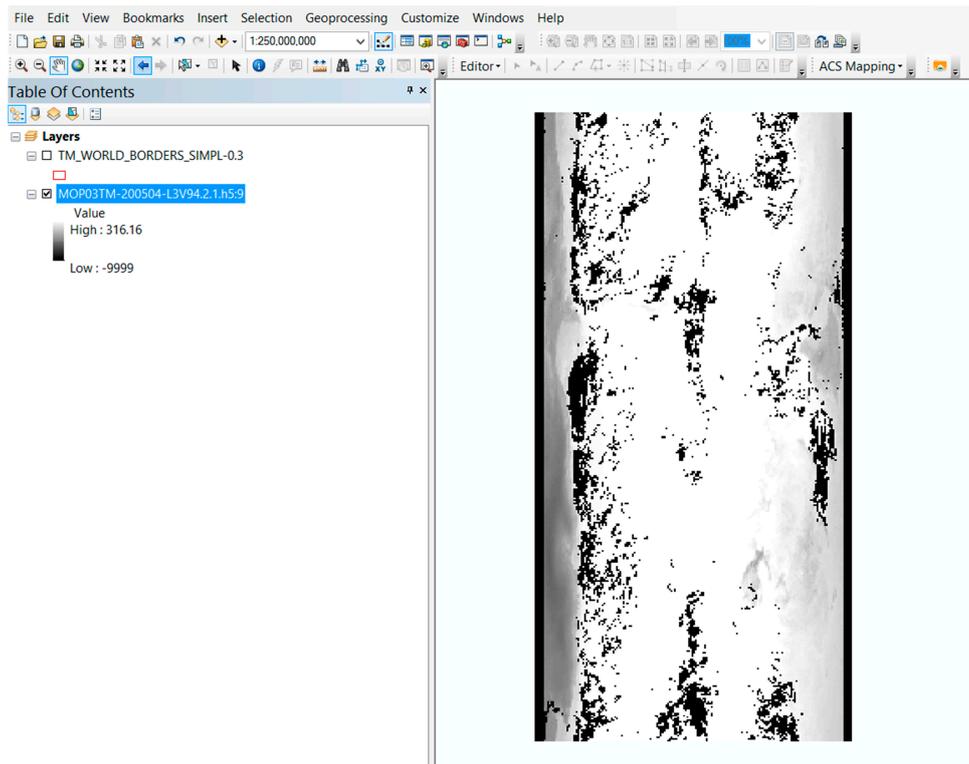


(b)

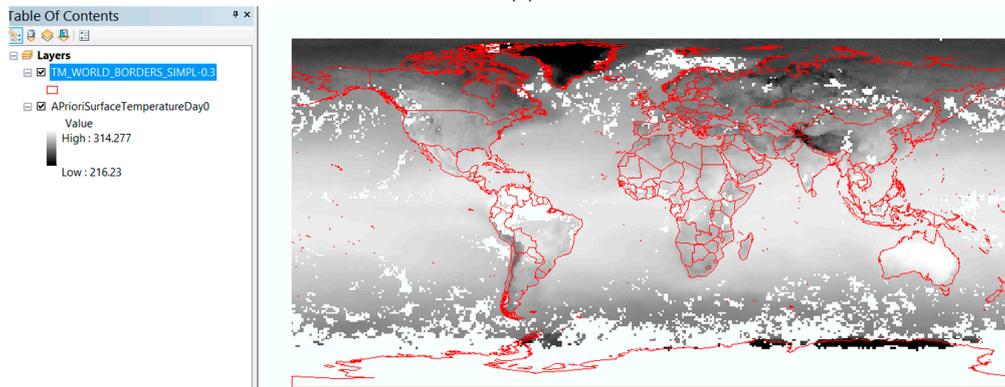
Figure 12. Image 180 degree rotated (a) compared to the correct interpretation (b) after incorporating the enhanced GDAL (note that, the difference on the pixel value of (a) from (b) is caused by the adjustment of missing data value interpretation problem as discussed in case 4).

5.3. Case 3: Repair Images Rotated by 90 Degrees

The problem of 90 degree rotation usually occurs to EOS HDF 5 data files when opening in GIS. MOPPIT HDF5 data (MOP03TM-200003-L3V94.2.1.he5) is selected as the sample data. Figure 13a shows the result of selecting 2D sub-dataset (“A Priori Surface Temperature Night”) of the HDF data file in ArcGIS. The raster image is rotated 90 degrees. While Figure 13b shows the correct displaying result with the ArcGIS extension and framework developed.



(a)



(b)

Figure 13. Image, 90 degree rotated (a) compared to the correct interpretation (b) after incorporating the enhanced GDAL.

5.4. Case 4: Assign Missed NoData Value

Often times, developers and users overlooked putting NoData value into HDF metadata or retrieve them from HDF data file, which leads to that GIS are unable to properly understand and display HDF data with missing NoData value in GIS. MOPFIT HDF data (MOP03TM-200003-L3V94.2.1.he5) is used as the sample to show the importance of NoData value. Figure 14a shows the wrong result without assigning NoData value in ArcGIS, while Figure 14b shows the correct data when using NoData value as -9999. The upper white part in Figure 14b represents the area with NoData value, and the red outlines are the world boundary at the national level.

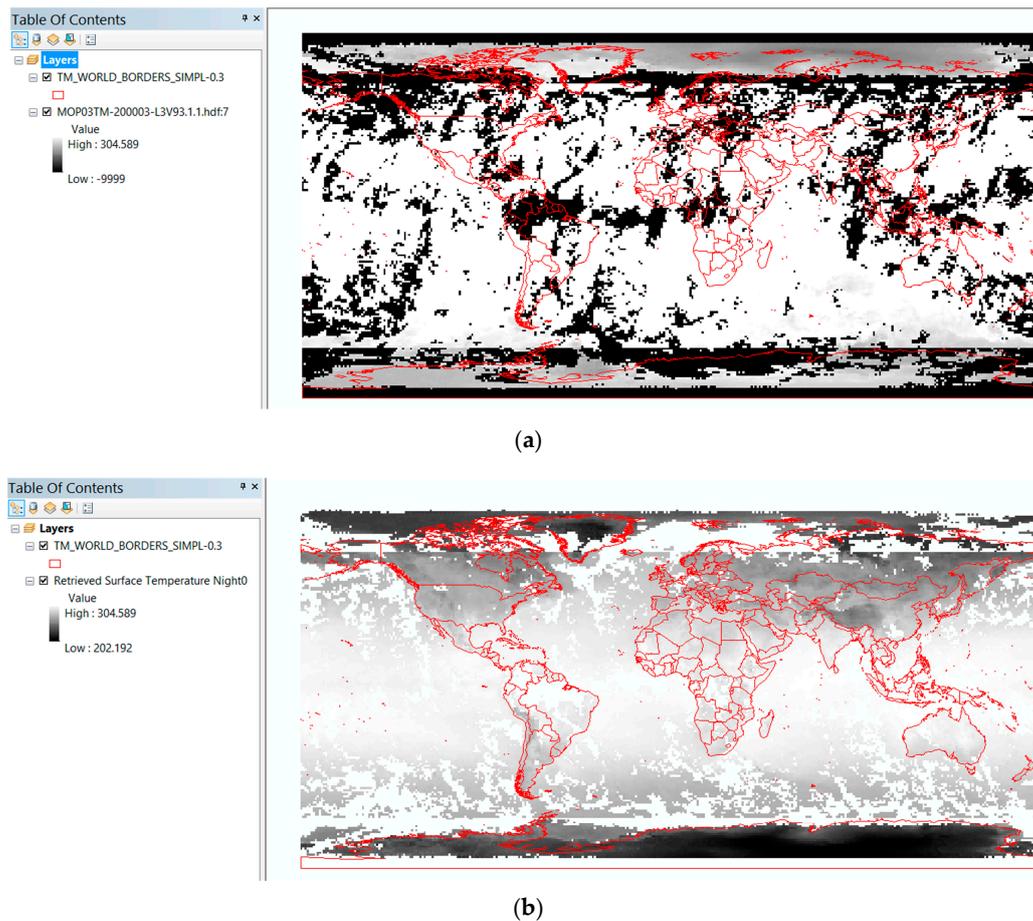


Figure 14. Color scale including No-data value -9999 (a) and excluding No-data value (b) by incorporating the enhanced GDAL.

6. Software Availability

To contribute the framework to GDAL, the implementation into GDAL at a source code level was completed and is available at the NASA EOSDIS program's share sites for NASA EO data users. The link for downloading the open source based on GDAL 2.0.0 version is at [21]. The user guide and sample data can be found at [22]. A further investigation is being planned to (a) integrate all improvements into the GDAL open source release; and (b) produce cloud computing golden image, a snapshot of a computer running status with all software well configured and tested, for remote sensing community [23] and the general public.

7. Conclusions and Future Work

As an integral part of a NASA project, this research is a significant contribution for integrating EO data into GIS. It focuses on upgrading the delivery of EO data for consumption in GIS by solving data interpretation issues. The proposed XML-based plug-in framework enables GIS to efficiently and effectively access and interpret HDF-EOS data, especially for multiple dimensional data. To enhance GDAL's ability to process EO data, GDAL source code was optimized for interpreting multi-dimensional variables and overcoming the limitation of accessing complicated HDF data. Based on the enhanced GDAL and the generic plug-in framework, cross-platform GIS extensions were developed to facilitate GIS better user access of EO data. Although different GIS have their own APIs, the proposed framework and workflow can be adopted in the same fashion as we tested in both ArcGIS and QGIS. A series of experiments demonstrated that ArcGIS extension solves a number of practical

problems when integrating the data in GIS, especially for multiple dimensional data. Furthermore, the work is encapsulated into GDAL source code and shared with the NASA EOSDIS community. It is expected that the methodology can be applied in other domains. For example, the same strategy could be used to test additional EO data products from NASA's other data centers and make the tools more elastic, scalable, and robust in manipulating and interpreting EO data in different GIS and relevant decision support tools. When a large number of use cases are accumulated for the solution, a knowledge-based heuristic solution may also be investigated for dealing with more future problems.

Acknowledgments: This research is supported by NASA AIST (NNX15AM85G), EOSDIS, and the NSF spatiotemporal innovation center (IIP-1338925). Jeff Walter provides financial support. Brandi M. Quam provides organizational support from NASA ASDC (NNL11A00B). Joe Lee from HDF verified the source code, Brain E. Tisdale and Tiffany J. Mathews from ASDC tested the results. George Taylor proofed an earlier version of the paper.

Author Contributions: Chaowei Yang originated the framework idea with Mike Little. Yunfeng Jiang developed the framework, wrote the main manuscript, performed experiments, and prepared the figures and tables. Min Sun developed the ArcGIS extensions, tested the experiments and offered suggestions on the methodology. Chaowei Yang and Yunfeng Jiang wrote the manuscript. Min Sun revised the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information System
EOS	Earth Observing System
ASDC	Atmospheric Scientific Data Center
HDF	Hierarchical Data Format
NASA	National Aeronautics and Space Administration
MOPPIT	Measurement of Pollution in the Troposphere
AIRS	Advanced Infrared Sounder
TES	Tropospheric Emission Spectrometer
CERES	Clouds and Earth's Radiant Energy System
MISR	Multi-angle Imaging SpectroRadiometer
APIs	Application Program Interfaces

References

1. Farr, T.; Granger, S.; Kopp, S. Integrating remote sensing data into geographic information systems. *Trans. Am. Geophys. Union* **2011**, *92*, 154. [[CrossRef](#)]
2. Bagwell, R.; Lindsay, F.; Lynnes, C.; Yang, M. Using NASA remote sensing data in a geographical information system. In Proceedings of the American Public Health Associate (APHA) 2011 Annual Meeting, Washington, DC, USA, 29 October–2 November 2011.
3. Van de Vegte, J.; de Cerff, W.; van den Oord, G.; Sluiter, R.; van der Neut, I.; Plieger, M.; Wilhelmi, O.V. Atmospheric data access for the geospatial user community. *Proc. SPIE* **2007**, *6749*. [[CrossRef](#)]
4. Wilkinson, G. A review of current issues in the integration of GIS and remote sensing data. *Int. J. Geogr. Inf. Sci.* **1996**, *10*, 85–101. [[CrossRef](#)]
5. Wright, D.; Goodchild, M. Data from the deep: Implications for the GIS community. *Int. J. Geogr. Inf. Sci.* **1997**, *11*, 523–528. [[CrossRef](#)]
6. GDAL. Available online: <http://www.gdal.org/> (accessed on 5 December 2015).
7. Yang, C.; Sun, M.; Liu, K.; Huang, Q.; Li, Z.; Gui, Z.; Jiang, Y.; Xia, J.; Yu, M.; Lostritto, P. Contemporary computing technologies for processing big spatiotemporal data. In *Space-Time Integration in Geography and GIScience*; Springer: Dordrecht, The Netherlands, 2015; pp. 327–351.
8. ESRI ArcGIS. Available online: <http://doc.arcgis.com/en/arcgis-online/create-maps/display-imagery.htm> (accessed on 15 September 2015).

9. QGIS. Available online: <https://www.qgis.org/en/site/getinvolved/development/cppplugindevelopment.html> (accessed on 6 April 2016).
10. NASA Earth Science Mission. Available online: http://eosps.nasa.gov/files/mission_profile.pdf (accessed on 10 August 2015).
11. Goodchild, M. Integrating GIS and remote sensing for vegetation analysis and modeling: methodological issues. *J. Veget. Sci.* **1994**, *5*, 615–626. [[CrossRef](#)]
12. Hill, L. *Georeferencing: The Geographic Associations of Information*; MIT Press: Cambridge, MA, USA, 2009.
13. Lo, C.; Quattrochi, D.; Luvall, J. Application of high-resolution thermal infrared remote sensing and GIS to assess the urban heat island effect. *Int. J. Remote Sens.* **1997**, *18*, 287–304. [[CrossRef](#)]
14. Petropoulos, G.; Kalivas, D.; Griffiths, H.; Dimou, P. Remote sensing and GIS analysis for mapping spatio-temporal changes of erosion and deposition of two Mediterranean river deltas: The case of the Axios and Aliakmonas rivers, Greece. *Int. J. Appl. Earth Observ. Geoinf.* **2015**, *35*, 217–228. [[CrossRef](#)]
15. NSIDC (National Snow & Ice Data Center). Available online: <http://nsidc.org/data/hdfeos/intro.html> (accessed on 10 August 2015).
16. Tsou, M. Integrating Web-based GIS and image processing tools for environmental monitoring and natural resource management. *J. Geogr. Syst.* **2004**, *6*, 155–174. [[CrossRef](#)]
17. Raskin, R.; Pan, M.; Mattmann, C. Enabling semantic interoperability for earth science data. In Proceedings of 4th NASA Earth Science Technology Conference, Palo Alto, CA, USA, 22–24 June 2004.
18. Shalaby, A.; Tateishi, R. Remote sensing and GIS for mapping and monitoring land cover and land-use changes in the Northwestern coastal zone of Egypt. *Appl. Geogr.* **2007**, *27*, 28–41. [[CrossRef](#)]
19. Di, L.; Chen, A.; Yang, W.; Zhao, P. The integration of grid technology with OGC web services (OWS) in NWGIS for NASA EOS data. In Proceedings of the Eighth Global Grid Forum, Seattle, WA, USA, 25 June 2003.
20. Qin, C.Z.; Zhan, L.J.; Zhu, A. How to apply the Geospatial Data Abstraction Library (GDAL) properly to parallel geospatial raster I/O? *Trans. GIS* **2014**, *18*, 950–957. [[CrossRef](#)]
21. GDAL Enhancement for ESDIS. Available online: <https://git.earthdata.nasa.gov/projects/GEE/repos/gdal-enhancements-for-esdis/browse/gdal-2.0.0> (accessed on 26 April 2016).
22. Enhanced GDAL's User Guide and Sample Code. Available online: <http://199.26.254.185:8080/pages/viewpage.action?pageId=1016592> (accessed on 26 April 2016).
23. Yang, C.; Xu, Y.; Nebert, D. Redefining the possibility of digital Earth and geosciences with spatial cloud computing. *Int. J. Dig. Earth* **2013**, *6*, 297–312. [[CrossRef](#)]



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).