P1.11 A NEW APPLICATION UTILIZING GOOGLE EARTH TO PROJECT VERTICAL INFORMATION OF CLOUDS AND AERSOLS OBTAINED BY CALIPSO

Paul T. Ewing*, N. A. Gasperoni, T. N. Hillyer, B. S. Hughes, J. W. Carpenter, and J. Gleason
DEVELOP National Program: NASA Langley Research Center, Hampton, VA

1. INTRODUCTION

The *Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation* (CALIPSO) satellite, launched in April 2006 by NASA, measures the vertical distribution of clouds and aerosols in the Earth's atmosphere. The satellite is equipped with the *Cloud-Aerosol LIdar with Orthogonal Polarization* (CALIOP) instrument which measures various atmospheric parameters. CALIPSO has the capability of measuring previously unavailable information regarding high altitude aerosols and cloud tops. Particles in the upper atmosphere travel greater distances leading to air quality concerns (Winker 2006).

Aerosols are produced both naturally and anthropogenically and their presence in the atmosphere plays a key role in the change of the Earth's climate. The changes that aerosols pose to the climate vary dramatically from reflecting sunlight back into space thereby cooling Earth, to absorbing sunlight and warming the atmosphere. They also have the possibility, in large concentrations, to introduce serious health concerns, especially to those with respiratory problems.

1.1 CONCEPT DESIGN

This paper describes a technique to integrate CALIPSO data and project it into the Google Earth application. This CALIPSO visualization tool is composed of a server connecting three software programs (see figure below). The first program is the Geometric Parameter Calculator, which performs calculations using the CALIPSO data retrieved from the Langley Atmospheric Science Data Center. The second component, the Image Rendering Utility, generates a color coded image representing the total attenuated backscatter. The last program, the KML Builder, loads the visuals into Google Earth (Google). Google Earth was chosen as the user interface because it utilizes the Space Act Agreement (partnership between NASA and Google), it is widely known, and it is freely distributed.

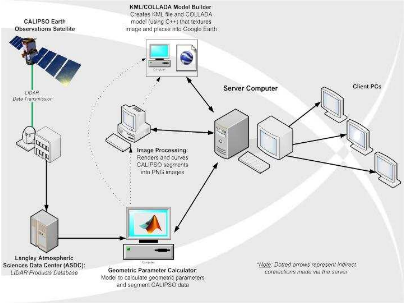
KML/COLLADA Model Builder:
Creates KML file and COLLADA model (using C++) that textures image and places into Google Earth
CALIPSO Earth Observations Satellite
LIDAR Data Transmission
Computer
Server Computer
Client PCs
Image Processing:
Renders and curves CALIPSO segments into PNG images
Langley Atmospheric Sciences Data Center (ASDC):
LIDAR Products Database
Computer
Geometric Parameter Calculator:
Model to calculate geometric parameters and segment CALIPSO data
*Note: Dotted arrows represent indirect connections made via the server

Figure 1.1.1: Illustration of the concept design, a series of computer programs working together to create Google Earth visuals.

2. METHODS

The process for integrating CALIPSO data into Google Earth involves texture mapping backscatter images onto rectangular polygons. The requirements for this approach include a pre-curved texture and the polygon's geolocation on the Earth's surface. To satisfy these needs, the process was broken into three components.

First, the Geometric Parameter Calculator reads CALIPSO data stored within *Hierarchical Data Format* (HDF) files (HDF Group). The initial step is to partition the backscatter data in order to account for the data's curved path due to the Earth's rotation and to control the output file size. Then through a series of geometric calculations the geolocation parameters are found.

These variables are then utilized by the other two components, the Image Rendering Utility and KML Builder. Following a color rubric, the Image Rendering Utility generates a *Portable Network Graphic* (PNG) file representing the segmented backscatter data. The utility uses a polar coordinate transformation to plot the data concentric to the Earth's surface. Lastly, the KML Builder uses the geolocation parameters and the backscatter image to build a polygon and then texture map the image onto it. The generated KML is readable by Google Earth.

3. GEOMETRIC PARAMETER CALCULATOR

3.1 SPHERICAL HEADING

The current release of Google Earth (Version 4) requires each rectangular polygon to be positioned by a single control point and a heading about the *z* axis. The Geometric Parameter Calculator partitions the data and converts a segment's latitude and longitude points into heading and sets the midpoint of the segment as the control point.

Heading is found using spherical trigonometry (see figure below).

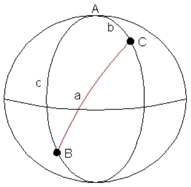

Figure 3.1.1: This figure shows the spherical triangle that is used to find heading in Equation

[*] Paul Ewing, Pennsylvania State University (Student), Huntingtown, MD, email: pte5000@psu.edu

Heading of the segment is the angle between *c* and *a*. The variable *c* is the line of longitude which faces due North and *a* is the arc of the great circle joining points *B* and *C*. *B* and *C* are the initial and final Earth surface points of the segment, respectively (Katsiavriades 2004). Using the sine rule, heading (angle *B*) is calculated.

3.2 SEGMENT GEOMETRY

A series of geometric equations are used to find the segment dimensions and the dropping distance. Dropping distance is the length (in meters) from the polygon's control point to the surface of the Earth. The control point is always located below the surface such that the curved CALIPSO image is flush with the Earth's surface. The process for making this curved image is discussed later in the methods of the Image Rendering Utility.

Figure 3.2.1 displays geometric relationships used throughout this calculator. The variables labeled in this diagram are referenced in the following discussion.

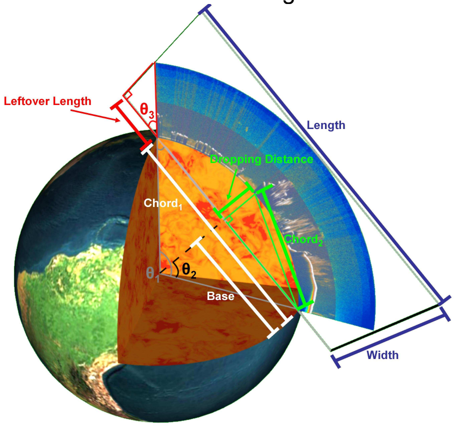

Figure 3.2.1: Illustration showing the different variables used throughout this tool.

The first step in finding dropping distance is computing θ_1, the angle created by connecting the center of the Earth to the endpoints of the CALIPSO image. This is done using Equation 3.2.2,

$$\theta_1 = \frac{length_{segment} \times horizontal\ resolution}{radius_{Earth}}, \quad (3.2.2)$$

where the segment length is the number of rows in the data for that particular partition. The horizontal resolution is *333 m* and the radius of the Earth is assumed to be *637,100 m*.

Pythagorean's Theorem can be applied to find the dropping distance, once the remaining two sides (*Base* and *Chord$_2$*) of the right triangle are known.

3.3 DIMENSIONS

The last step of this program is to find the dimensions of the rectangular polygon. To start, the leftover length needs to be found, which is the distance between the end of the CALIPSO data and the end of the image (see Figure 3.2.1). First, find θ_3 by utilizing Equation 3.3.1,

$$\theta_3 = 180 - 90 - \theta_2, \quad (3.3.1)$$

because there are *180°* in a triangle and the known two angles (*90°* and θ_2 which is $\theta_1 / 2$).

The leftover length can be found using this hypotenuse and θ_3 in Equation 3.3.2,

$$length_{leftover} = \cos(\theta_3) \times 20000, \quad (3.3.2)$$

by using the fact that the CALIPSO backscatter image has an altitude of *20,000 m*.

To compute the horizontal distance of the image, *length*, add together the two leftover lengths and *Chord$_1$* seen in Equation 3.3.3,

$$length = 2 \times length_{leftover} + chord_1. \quad (3.3.3)$$

Width is found by Equation 3.3.4,

$$width = dis\tan ce_{dropping} + 20000, \quad (3.3.4)$$

which simply combines the dropping distance and the known altitude of backscatter data (*20,000 m*).

4. IMAGE RENDERING UTILITY

The raw CALIPSO segments are color-coded and fitted to the curve of the Earth by an image processing utility. This component imports the backscatter data from the Geometric Parameter Calculator and outputs a *Portable Network Graphic* (PNG) file representing the vertical distribution of clouds and aerosols.

4.1 COLOR CODING

The utility first reads in the binary data from the files produced by the Geometric Parameter Calculator and then uses a *Visualization ToolKit* (VTK, Kitware Inc.) color look-up table created by team technical advisor, Jonathan Gleason, to assign colors to each scalar data point. During this process, the Image Rendering Utility can either filter the backscatter to be transparent, non-transparent, or semi-transparent. For instance, when integrating CALIPSO and CloudSat into Google Earth, a transparent filtering method is used to allow visualization of both satellites' data. Non-transparent backscatter images are usually preferred by the CALIPSO Science Team when showing only CALIPSO data in two dimensions. The PNG format was selected for transparency support.

4.2 PNG CREATION

This component then uses *ImageMagick* (ImageMagick Studio LLC) to output a two dimensional image. The resulting visual is written to a PNG file. It is then vertically cropped to show the bottom *20 km* of CALIPSO backscatter data. Figure 4.2.1 displays a two dimensional image of CALIPSO backscatter data before curving is applied to account for the Earth's spherical nature.

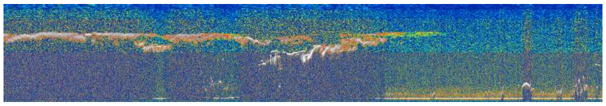

Figure 4.2.1: Two dimensional representation of color coded backscatter data prior to curving.

The graphic is then transformed to fit the Earth's surface by converting the rectangular image coordinates into polar coordinates based on varying vertical resolutions of the data set (see Figure 4.2.2).

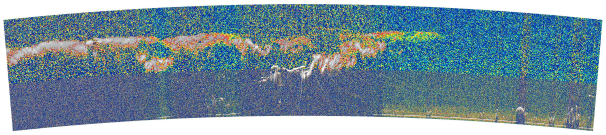

Figure 4.2.2: Curved image of CALIPSO backscatter data.

The distance from the Earth's center for each image pixel is found by multiplying the image column by the vertical resolution and adding it to the radius of the Earth. This data set is then rotated to be centered on the *90°* axis. The polar coordinates are converted back to Cartesian coordinates. The plot is then shifted horizontally to align all data points in the first quadrant by the Equation 4.2.3,

$$x_{after\ shift} = x_{before\ shift} + 2 \times width_{image} \times \cos(\frac{\pi - \theta_1}{2}) + 2 \times \sin(\frac{\theta}{2}) \times radius_{Earth}, \tag{4.2.3}$$

where $x_{after\ shift}$ represents the horizontal shifted component, $x_{before\ shift}$ is the original horizontal point, and remaining variables are seen in Figure 3.2.1

A vertical shift is computed in Equation 4.2.4,

$$y_{after\ shift} = y_{before\ shift} - \sin(\frac{\pi - \theta_1}{2}) \times radius_{Earth}, \tag{4.2.4}$$

where $y_{after\ shift}$ represents the vertical shifted component and $y_{before\ shift}$ is the original vertical point.

Now that the horizontal/vertical shifts are completed which arranges the image on the proper canvas in *ImageMagick*, the PNG is finalized. This backscatter graphic is then sent to the KML Builder.

5. KML BUILDER

The next step is to format the necessary files for loading into Google Earth, executed by the KML Builder. These files follow the *COLLAborative Design Activity* (COLLADA, KHRONOS Group) and *Keyhole Markup Language* (KML, KML Support Group), file formats used natively by Google Earth. The templates used for editing were extracted from *Google SketchUp* (Google). *SketchUp* is a three-dimensional modeling program which creates textured objects uploadable into Google Earth.

5.1 TEMPLATE AND POLYGON CONSTRUCTION

The syntax of the KML and COLLADA formats was determined by using SketchUp to create a *1000 m* by *1000 m* polygon in the y-z plane. The output from Google SketchUp was examined and constructed into a set of templates. These templates can be modified by the KML Builder.

5.2 KML BUILDER FRAMEWORK

1. KML Network Linking File

Loading KML files individually can be tedious due to the many segments associated with a half orbit data file. To alleviate this issue, the KML schema allows for a single Network Linking File to reference other Geographic Positioning KML Files which automates most of the loading process.

2. Geographic Positioning KML File

The KML files contain the segment's latitude, longitude, and heading. These variables are retrieved from the Geometric Parameter Calculator. Google Earth, uses the values in the Geographic Positioning KML to reference the COLLADA model containing the backscatter image (PNG).

3. COLLADA Model and VTK Rendered PNG

The final stage builds the rectangular polygon (COLLADA Model) and texture maps onto it a backscatter image (VTK Rendered PNG) using the corners of the polygon (see Figure 5.2.1). This is achieved by using the segment's width, length, dropping distance, and the PNG created in the Image Rendering Utility.

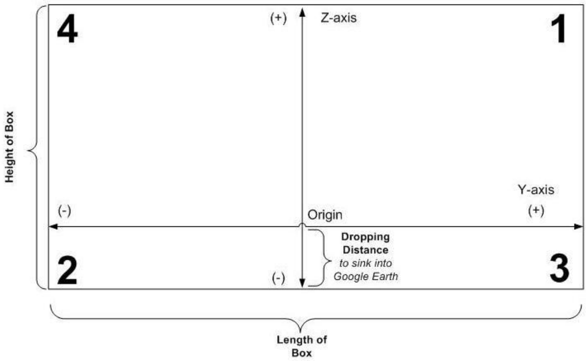

Figure 5.2.1: Segment box to show the order of coordinates based on the corners of the rectangle.

6. RESULTS

The following visual, Figure 6.1, displays the final product created from the data of the three instruments, CALIOP, CPR, and MODIS. These satellite observation systems were integrated into Google Earth to display a strong visual representation of the atmosphere.

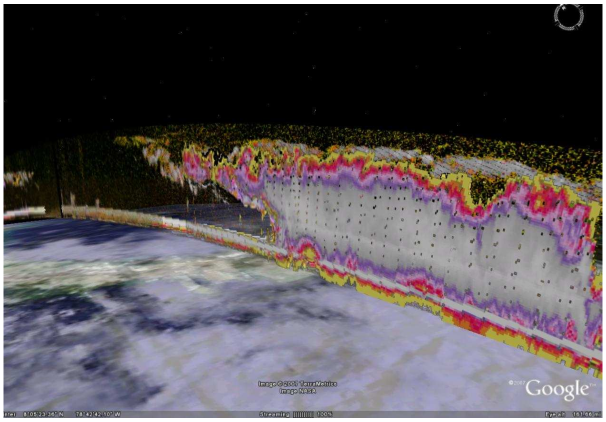

Figure 6.1: Screen capture of the final product, integration of CALIPSO, MODIS, and CloudSat data in Google Earth.

The main goal of this project was to create a visualization showing CALIPSO data in Google Earth which can be seen in Figure 6.2. The vertical distribution of clouds and aerosols can be seen from the CALIPSO backscatter data with the MODIS true color image overlaid on the Earth's surface.

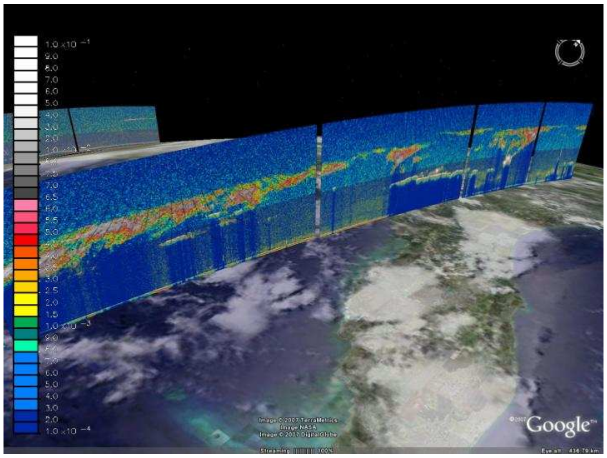

Figure 6.2: This visual displays an example of CALIPSO data three dimensionally in Google Earth.

7. DISCUSSION

7.1 SIGNIFICANCE OF RESULTS

The result of the CALIPSO Data Visualization Project is a versatile tool that can benefit numerous fields of study. The tool allows the end-user to view accurate and precise three-dimensional representations of aerosol backscatter data recorded by the CALIPSO satellite. By using Google Earth, one can pan the viewing camera to the location of interest where CALIPSO data is displayed. The data is displayed in three-dimensional segments extending from Earth's surface. Each segment shows the observed aerosol/cloud distribution based on the defined color rubric.

With this deliverable server-based model, the large assimilation of aerosol backscatter data can be displayed by the free KML-supporting Earth browser, Google Earth.

7.2 ERROR ANALYSIS

The segmentation of the CALIPSO path approximates segment length due to computational constraints. This estimation results in an approximate 5km gap between segments when viewed in Google Earth.

There is a significant error for segments close to the poles due to the extreme curvature of the CALIPSO satellite swath path. The mathematics implemented to calculate heading, yield unpredictable and inaccurate results close to the poles due to increased curvature of the satellites orbit path with respect to longitude. Calculations made every pixel inaccurate because the difference between the values used in the formula is minimal. The calculations per pixel are computationally intensive leading to this inaccuracy. In conclusion, the developer must choose a balance between precision and computational time.

The image renderer converts between multiple dissimilar coordinate systems, resulting in a final integer-based image grid. This constraint requires data points with non-integer coordinates to be rounded to a whole number. The translation between coordinate systems in the image renderer results in an insignificant round-off error. Consequently a minor distortion in the pixel grid develops.

REFERENCES

Google. *Google Earth*. Mountain View, California.

Google. *Google SketchUp: Release 6*. Mountain View, California.

Google KML Developer Support Group. KML Documentation. Google. Mountain View, California. code.google.com/apis/kml/documentation/ COLLADA.

ImageMagick Studio LLC. *ImageMagick: Release 6.3.4*.

Katsiavriades, Kryss and Talaat Qureshi. *Spherical Trigonometry*. The KryssTal Web Site: A United Kingdom Based Educational and Information Web Site. London, 2004. www.krysstal.com/sphertrig.html.

KHRONOS Group. COLLADA: 3D Asset Exchange Schema. Beaverton, Oregon. 2007. www.khronos.org/collada/.

Kitware Inc. *The Visualization ToolKit: Release 5*. Clifton Park, New York.

The HDF Group. Introduction to HDF5. hdf.ncsa.uiuc.edu/HDF5.

Winker, D., C. Hostelter, M. Vaughn, and A. Omar. CALIPSO algorithm theoretical basis document part 1: CALIOP instrument, and algorithms overview. Technical Report PC-SCI-202 Part 1, Release 2.0, NASA Langley Research Center, September 2006.