**NASA DEVELOP National Program**



NASA Langley Research Center

*Summer 2015*

CALIPSO Cross-Cutting

Interfacing CALISPO Data through a Graphical User Interface

 **Technical Report**

Final Draft – July 29, 2015

Grant Mercer (Project Lead)

Nathan Qian

Jeff Ely, NASA DEVELOP National Program (Science Advisor)

Dr. Kenton Ross, NASA DEVELOP National Program (Science Advisor)

Previous Contributors:

Jordan Vaa

Courtney Duquette

Ashna Aggarwal

# I. Abstract

The Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observation (CALIPSO) satellite is a NASA Earth observation that analyzes aerosol particles suspended in the Earth’s atmosphere. Researchers use visualized CALIPSO data to track the global distribution, dispersion, and source of aerosols. However, the current visualization tool for displaying CALIPSO data does not support needed features for tracking aerosols such as selecting areas of data and sharing those selected sections, making tracking specific airborne objects difficult for researchers. Adding these necessary features to the current CALIPSO visualization tool is difficult, as the tool is written in Interactive Data Language (IDL), a proprietary and obscure language, and writing additional features for the tool would require a specialized development team. For the 2015 summer term, our team is focused on the development of the *Visualization of CALIPSO* (VOCAL) Python program. VOCAL will serve as the successor to the current visualization tool for CALIPSO data. We will be building off the previous 2015 spring team’s work to implement a number of new features and offer new functionalities for Earth scientists to more easily identify the sources of aerosols and their impact on Earth’s climate.

**Keywords**

CALIPSO, CALIOP, LiDAR, Atmospheric Aerosols

# II. Introduction

**Background Information:**

Earth’s climate is a complex system that involves many variables and factors. One such factor is aerosols. Aerosols can come from anthropogenic (pollution or biomass burning) or natural (dust, sea salt, and volcanic) sources (Omar et al. 2009). Once in the atmosphere, aerosols can affect cloud formation and cloud radiative properties. Additionally, aerosols can suppress and enhance precipitation, simultaneously affecting the water cycle as well as radiative heating (Winker et al, 2010). Besides environmental impacts, aerosols can negatively affect human health in heavily populated regions (Winker et al, 2003). Unlike greenhouse gasses, aerosols are a major source of uncertainty in climate models due to their variable distribution and properties (Winker et al, 2003). Because of these effects, Earth scientists track and identify aerosols to analyze the aerosols’ impact on human health and Earth’s climate.

In April 2006, the Cloud-Aerosol LiDAR Infrared Pathfinder Satellite Observation (CALIPSO) was launched to provide measurements of clouds and aerosols and their interactions and roles in the climate system (Winker et al, 2010). CALIPSO carries three instruments for aerosol measurement. The primary instrument is the Cloud-Aerosol LiDAR with Orthogonal Polarization (CALIOP), a new-nadir viewing two-wavelength polarization-sensitive LiDAR sensor (Winker et al, 2009). The remaining two are passive sensors: a wide field camera and an infrared imaging radiometer (Winker et al, 2009). CALIPSO can determine the aerosol’s type through an algorithm by utilizing backscatter and volume depolarization ratio measurements (Omar et al. 2009). The algorithm takes altitude, integrated attenuated backscatter, location, surface type, and volume depolarization ratio as parameters (Omar et al 2009). With the addition of CALIPSO, researchers have access to more aerosol data and can better analyze aerosols’ effect on climate.

Prior to the spring of 2015, the CALIPSO science team used an Interactive Data Language (IDL) program written in 2007 to display CALIPSO data. Since 2007, the program has been maintained and updated with improvements. However, since IDL is a proprietary language, users outside of NASA cannot easily use nor customize the CALISPO visualization tool. In 2010, a Python program, called CloudSat and CALIPSO plotting tool (ccplot), was written to mimic the IDL program to display CALIPSO data and was used by the CALIPSO science team. Unlike IDL, Python is an open source language, which allows users to more easily use and change programs written in Python. Despite the conversion to Python, ccplot proved to be obtrusive as CALIPSO data is displayed as an unadaptable image and can only be manipulated through a command line. In the spring of 2015, a team at NASA DEVELOP started development of a new CALIPSO visualization tool called Visualization of CALIPSO (VOCAL) that will address the issues of the previous CALIPSO visualization programs. The spring term provided the back end implementations that displayed the initial GUI and the three different CALIPSO plot types.

**Project Objectives:**

For the summer term, our team implemented a number of new features and started the process of releasing VOCAL. The new features added allow users to draw shapes that mask specific aerosols, and share these shapes through a database developed for VOCAL. Specifically, a user can trace a shape or multiple shapes in the backscattered or depolarized plots from CALIPSO data that represents a specific aerosol. The user can then tag the shapes with relevant data, such type of aerosol, the aerosol’s source, or the aerosol’s composition. When the user exports the shapes to the database, the shapes are saved in a JavaScript Object Notation (JSON) file with additional metadata, such as the time of creation and the corresponding CALIPSO data product file. When the shapes are saved in the database, the user can retrieve them using queries based on the CALIPSO data product or user-generated tags. The code base left over from the spring term already displays level one and level two CALIPSO data products. However, our team made additional improvements, such as bug fixes and code documentation, to make VOCAL more reliable and user friendly. With these additions, researchers can better collaborate on aerosol tracking and identification.

**Study Area:**

The scope of this project is global, as VOCAL can read any CALIPSO curtain plot data.

**Study Period:**

The timeframe of this project ranges from 2006, when CALIPSO was launched, to the present day.

**National Application:**

Our end-product addressed the Health and Air Quality application area. VOCAL will aid researchers in identifying, tracking, and documenting aerosols and their effect on the environment and air quality.

**Project Partners:**

Our end-user is the CALIPSO science team. Our main point of contact with our users were Dr. Charles Trepte and Dr. Amber Soja, who provided guidelines on the VOCAL features and use cases. The shape drawing and database functionality of VOCAL tool will help the CALIPSO science team to better track and identify aerosols and let them more easily share aerosol research with each other.

# III. User Manual

**Installation**

VOCAL requires several dependencies to run. The installation process for these modules will vary depending on the user’s computer operating system. This installation guide will assume the user is running on a Windows operating system. For a more in depth installation guide or instructions for other operations systems, refer to the VOCAL documentation website [here](http://syntaf.github.io/vocal/index.html).

First download and install Python Anaconda 2.7. Afterwards, open a terminal and navigate to the Anaconda installation directory. Type “conda install numpy” into the terminal and type “y” when prompted. After installing numpy, install basemap by typing “conda install basemap” and type “y” again when prompted. Lastly, install PIL by typing “conda install pil” and confirm installation by typing “y” when prompted. PIL is an old library and may run into errors. Refer to the documentation website if VOCAL produces the error message “Import Error: cannot import name \_imagingtk”.

The next step is to install ccplot. First, download the executable from the ccplot website. Afterwards, run the executable to install ccplot. When prompted, install ccplot in Anaconda’s install directory “\Anaconda\Lit\site-packages\”.

After installing all the dependencies, the user may elect to install an integrated development environment (IDE). IDEs are optional, but they provide additional convenience on modifying the codebase. All Python distributions are installed with the IDLE IDE. The documentation website provides instructions on installing PyDev on Eclipse. The user should follow the IDE’s website for instructions on installing the desired IDE. Recommended IDEs include PyCharm and Vim. Emacs are acceptable, though discouraged.

**User Guide**

When the user starts the program, VOCAL will present the user with the CALIPSO program image. Figure 1 shows how the VOCAL appears during start up on a Windows system. The top of the screen at the window bar displays the name of the program, CALIPSO Visualization Tool (VOCAL). The window bar also contains the standard minimize, maximize, and close button. Below the window bar on Window machines is the menu bar. The menu bar contains the “File”, “Polygon”, and “Help” menus. For OS X systems, the menu bar will be at the top of the screen. The rest of the user guide will assume VOCAL is running on a Windows machine.

The “File” menu contains four commands: “Import file”, “Save all shapes”, “Save as shapes”, and “Exit”. The “Import file” command loads a CALIPSO .hdf file to display on VOCAL’s canvas. VOCAL will generate a file dialog where the user can explore his or her local file repository and select the desired .hdf file to display. Below “Import file” are “Save all shapes” and “Save as shapes”. “Save all shapes” will save all shapes drawn on the various plots, while “Save as shapes” will only save shapes from the displayed plot. Lastly, “Exit” will close out of the program. It shares the same functionality as the close button at the window bar. Upon closing, VOCAL will check if there are any unsaved shapes in any of the plots. If there are, VOCAL will ask the user if he or she would like to save them. The user has the option to save and close, close without saving, or cancelling and resume using VOCAL.

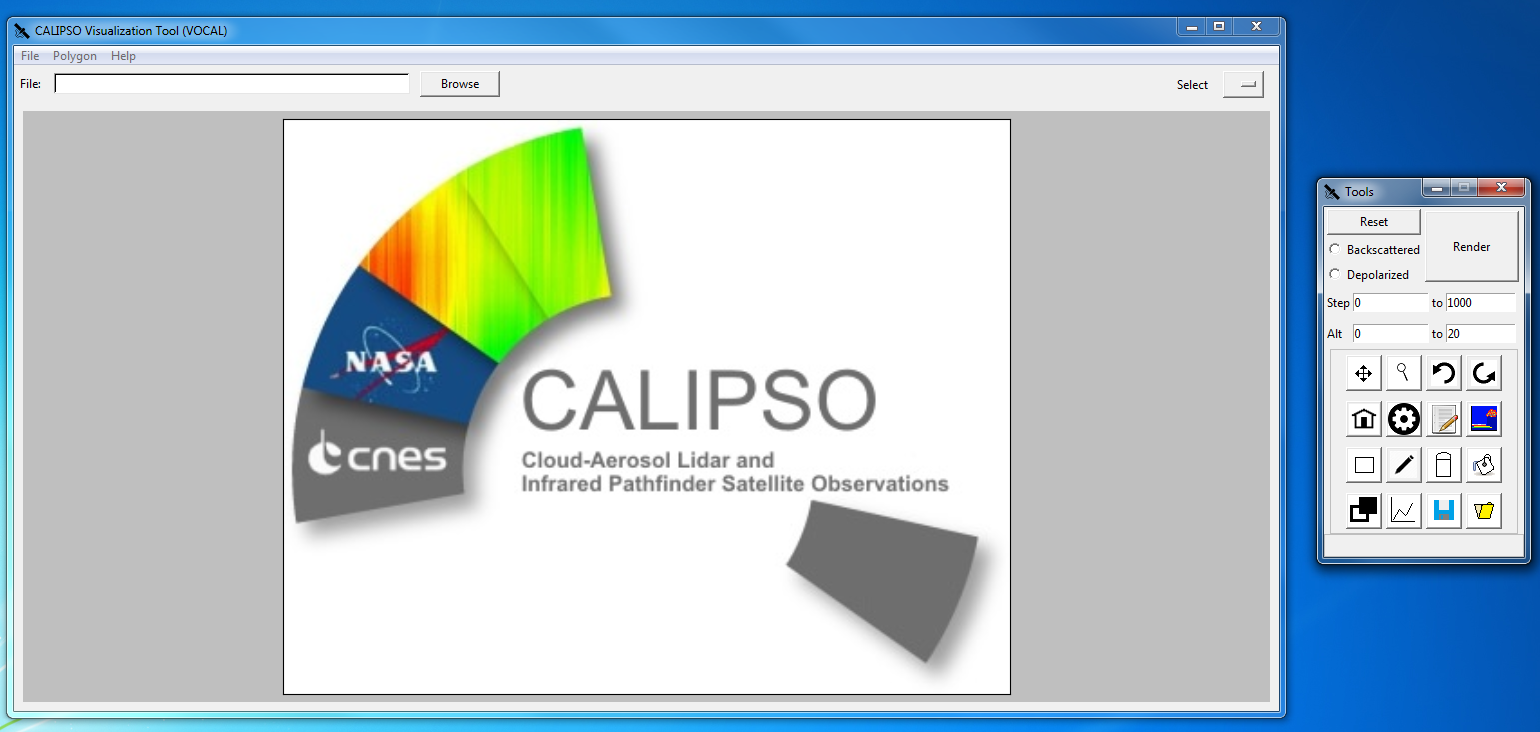


Figure 1. VOCAL's startup image

The “Polygon” menu holds four additional features. The first is “Import from Database.” Figure 2 displays the result of selecting this menu option. If the user selects this option, VOCAL will create a dialog box, displaying all saved shapes in the database. The user can query for specific shapes using the search bar at the top left of the database dialog. Additionally, the user can have the dialog box to display shapes drawn only for the displayed .hdf file by clicking on the “Filter for this file” checkbox. Lastly, the user can click on the “Advanced” button to have a more refined query. At the top right of the database dialog is the “Delete” button. After the user selected shapes form the database, the user can click this button to permanently delete the selected shapes from the database. This operation is irreversible. Lastly, at the bottom of the screen is the “Import” button. After the user makes a selection form the import dialog, the user can click this button, and VOCAL will draw and display these shapes on the .hdf file.

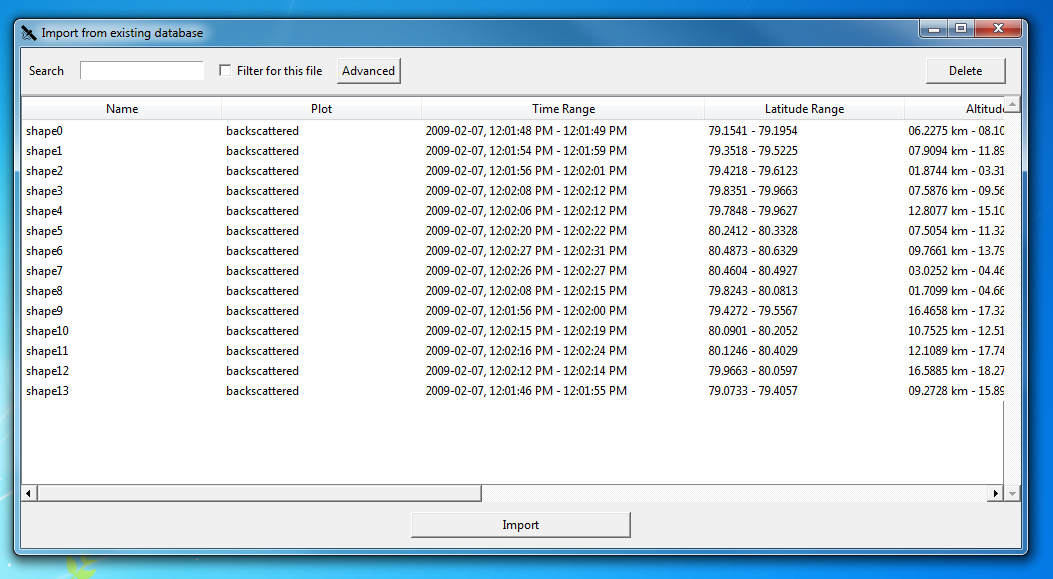


Figure 2. Database dialog

Below the “Import from “Database” option is the “Export to Database” feature. Clicking on this menu item will prompt VOCAL to save all shapes drawn on the .hdf file into the database. VOCAL will automatically let the user know if the operation is successful or not. The last two options in the “Polygon” menu are the “Import archives from database” and “Export database to archive”. Clicking of the former item will prompt the user with a file dialog that allows the user to select a zip archive containing shapes. VOCAL will access the selected compressed zip file and save all the shapes within the zip archive into the database. The latter option will select all the shapes in the database and save them into a zip archive. The user has the ability to specify where this zip folder will be saved.

The last menu is the “Help” menu. The “Help” menu only has two options, “Documentation” and “About”. Clicking on “Documentation” will open the documentation pages in the default web browser. The documentation pages is the main help page for the user and provides instructions on installing VOCAL, development support, program documentation, coding conventions, and an in depth tutorial and user guide. The user should default to this documentation website for any questions or troubleshooting. The “About” menu item displays all the DEVELOP participants who contributed to VOCAL’s development.

Below the menu bar is main screen’s button bar. The button bar contains three items: the file name text box, the “Browse” button, and the select dropdown menu. If an .hdf filed is loaded into VOCAL, the file name text box will display the name of the loaded .hdf file. Directly next to the file name text box is the “Browse” button. The “Browse” button is identical to the “Import file” command under the “File” menu. Lastly, the select drop down menu allows the user to select from all the shapes drawn on the current plot. Selecting a specific shape will change the shape’s solid outline to a broken outline to help the user identify the selected shape.

The last item on the main screen is the canvas. The canvas displays .hdf file and the shapes drawn on the file. When an .hdf file is displayed, the name of the plot will show up at the top of the canvas. Immediately below the title is the latitude scale and label. On the left is the altitude axis and label. At the bottom is the time scale and label. On the left is the color bar scale, identifying the values that correspond to the color of the individual pixels in the plot. Additionally, the user manipulates the plot and shapes drawn here through the use of the toolbar.

To the right of the main screen is the toolbar. Figure 3 shows the toolbar. The toolbar provides the user additionally functionality to manipulate CALIPSO data and shapes. The window bar of the toolbar contains the title of the toolbar, “Tools”, and helps the user identify the toolbar. Immediately below the window bar is the “Reset” button. The “Reset” button will set the zoom to its original scope and deletes all shapes drawn on the plot. If the shapes were not saved before pressing the “Reset” button, these shapes are permanently destroyed. Below the “Reset” button are the “Backscattered” and “Depolarized” radio buttons. These two radio buttons are mutually exclusive selections. After selecting either the “Backscattered” or “Depolarized” options, the user then presses the “Render” button to display these plots on the main screen’s canvas. Underneath the radio and “Render” buttons are the step and altitude entry boxes. Here, the user can specify how far along the time or latitude axes to display the plot. Likewise, the altitude entry boxes specify how far along the altitude axis to display. The entry boxes will automatically inform the user of invalid entry selections.

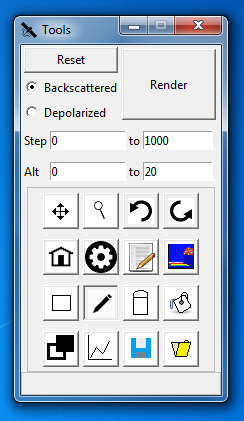


Figure 3. VOCAL toolbar

After the entry boxes are the buttons. The first row of buttons are for navigation. The first button in this row is the “pan” button. Clicking this button will allow the user to move the plot left or right. The second button is the “magnify” button, which allows the user to zoom in and out of the plot. To zoom in, left click and drag on an area to zoom in. To zoom out, right click and drag on an area to zoom out. The third button is the “undo” button. The “undo” button will undo the most previous navigation operation. The final button is the “redo” button. Clicking on this button will redo a previously undone navigation operation.

The second button row contains one additional navigation button and three shape manipulation buttons. The first button is the “home” button. The “home” button will reset the zoom scale. The second button is the “properties” button. This button allows the user to see a shape’s properties. To view a shape’s properties, simply click on it. The third button is the “edit” button, which allows the user to edit the attributes of a shape. To edit a shape, click on it. The fourth and final button is the “extract” button. Clicking a shape with this button selected with display the data bounded by the selected shape.

The third button row contains drawing operations. The first button is the “rectangle” button. The “rectangle” button allow the user to draw rectangles. To draw a rectangle, click and drag on the plot. The next button is the “free draw” button. With this button, the user can draw free form shapes. To draw polygonal shapes, click on points of the plot to create the vertices of the shape. After the first point, VOCAL will automatically draw a line between the points. As soon as two lines intersect, VOCAL will draw a shape based on these lines and points. The third button is the “erase” button, which allow the user to delete shapes by clicking on the desired shape. The fourth button in this row is the “paint” button. To recolor a shape, click on a shape. VOCAL will then prompt the user to select a color to repaint the shape.

The final button row holds shape view and saving and loading operations. The first button is the “focus” button. The “focus” button will toggle displaying the shapes outline. The second button is the “hide polygon” button. Clicking on this button will toggle displaying all the shapes on the plot. The third button is the “save” button, which is identical to the “Save as shapes” found in the “File” menu. The last button is the “load” button, which loads JSON files when selected. VOCAL will produce a file dialog, where the user can select the desired JSON file to display.

The last item in the toolbar is the coordinate display. When an .hdf file is displayed, the coordinate display the coordinates of the mouse cursor based on the cursor’s position on the plot. The first item in the coordinate display is the time coordinate. The next item is the altitude coordinate. The final item is the latitude coordinate.

# IV. Results & Discussion

VOCAL builds upon and expands on the old IDL tool. With the new features, primarily the shape drawing and the database sharing, Earth scientists can better understand how aerosols affect Earth’s climate and human health. With the old tool, our partners would identify aerosols of interest, estimate the boundaries of the aerosols, and write them to an Excel spreadsheet. The spreadsheet would also contain other characteristics of the aerosol, such whether it was smoke, clouds, or pollution, and any notes by the researcher. Since the IDL tool was incapable of sharing specific data features, if a scientist wanted to share an aerosol, he or she would have to share the Excel spreadsheet instead. The receiving scientist would then have to open both the Excel spreadsheet and the IDL program to view the aerosol. With VOCAL, scientists now have a centralized way of viewing and sharing aerosol data.

In VOCAL, if an Earth scientist wants to highlight a specific aerosol within CALIPSO data, all he or she needs to do is draw a shape that bounds the aerosol, instead of approximating the boundaries and writing them down in an Excel sheet. The shape makes the aerosol of interest immediately noticeable to the researcher and other users compared to reading coordinates from an Excel spreadsheet and searching the shape from the coordinates. The scientist can then add attributes to the shape. VOCAL has a preset set of standardized attributes that the scientist can label the shape, such as smoke, volcanic plume, or cloud. Since VOCAL is an open source program, Earth scientists can add, remove, or change the existing list of attributes to meet their needs. Additionally, the scientist can add a note to the shape to specify additional characteristics or features of the aerosol. Figure 4 demonstrates the shape drawing and figure 5 shows the attribute-labeling feature of VOCAL.

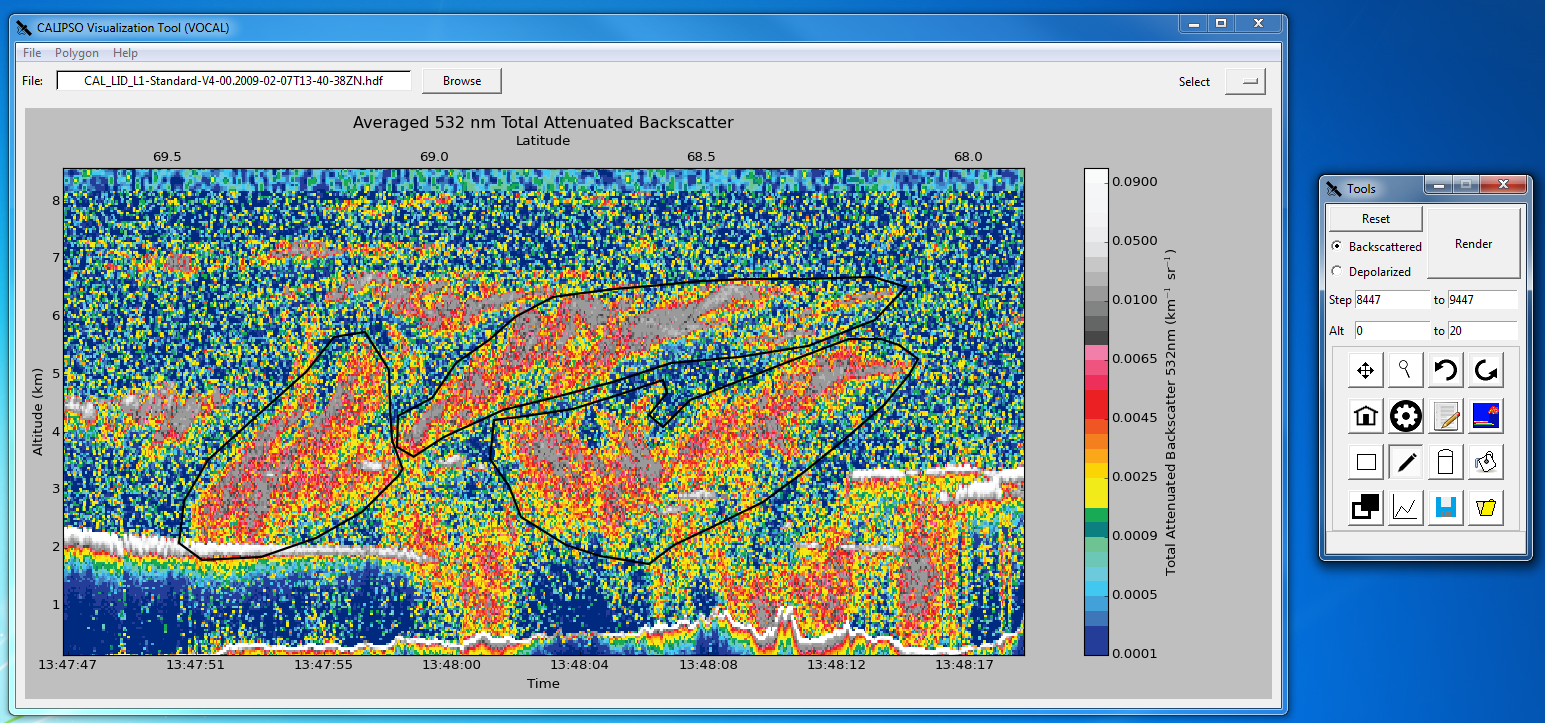


Figure 4. Sample of shapes

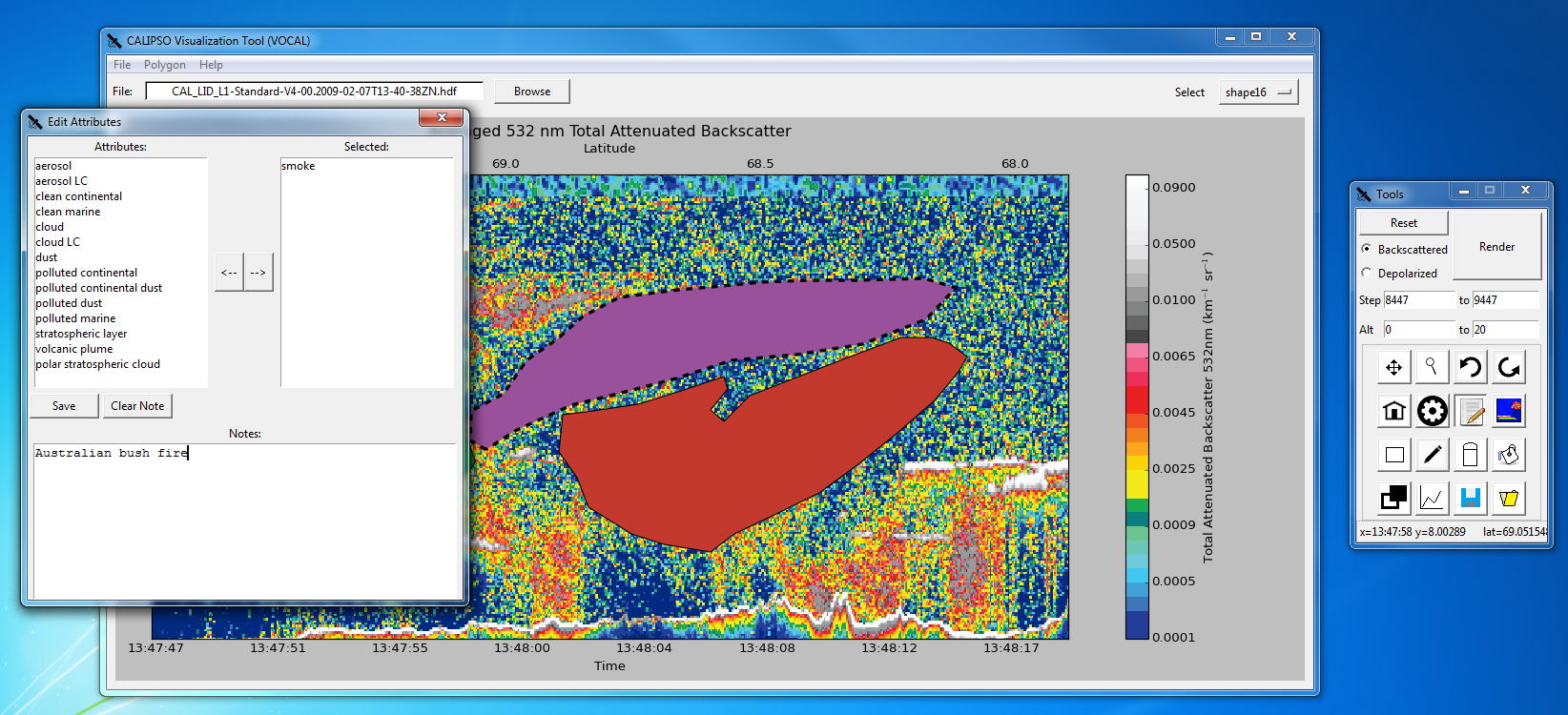


Figure 5. Sample of attributes

On its own, VOCAL’s shape drawing feature simply adds convenience to Earth scientists when identifying aerosols. However, the shape drawing functionality proves its merit in allowing researchers predict the trajectory of aerosols. A scientist can label an aerosol as a volcanic plume for one CALIPSO curtain plot. Then, for another CALIPSO plot at a different time and place, the scientist can label the same volcanic plume that has traversed the globe. After identifying all the volcanic plumes in all the curtain plots, the researcher can then export all these shapes into the database. Another user can view these shapes remotely and see all the volcanic plumes in the various CALIPSO data files. In the IDL program, the user would have to read off coordinates of the volcanic plumes for all the CALIPSO files and find the aerosols, whereas VOCAL makes the volcanic plume immediately noticeable with the drawn shapes. Additionally, the shapes display the volcanic plumes traversal graphic. As the user flips between the different CALIPSO .hdf files, the user can see in which the direction of the plumes travelling and can make predictions of the trajectory based on the shapes’ display. Based on the predicted trajectory, the user can analyze how the volcanic plume might affect local weather patterns, the global climate, and impacts on human health and air quality. The shape drawing and database sharing are not just simple quality of life additions; these new functionalities help Earth scientists identify and track aerosols within the atmosphere to better predict the aerosol’s role on global climate and health.

During the summer term, the team successfully implemented the shape drawing and database sharing features on time, the two main objectives of this project. Future development can refine and expand on these features. Future work may focus on minor bug fixes, optimizations, or other minor improvements. However, there are still areas on VOCAL in which the program can significantly be improved. Currently, VOCAL has functionality to extract data bounded by shapes, though the current implementation is not geared towards Earth scientists’ needs. Future work should focus on this data-extracting feature and tailor it to suit the researchers’ needs.

Another possible improvement is updating the ccplot library. Ccplot is integral for reading CALIPSO .hdf file and the original author of ccplot has discontinued support for the library. The biggest flaw with ccplot is that it only runs with 32 bit Python. This severely limits VOCAL’s computational power, as this forces VOCAL and all the library is uses to be in 32 bits as well. An ambitious improvement would be to reimplement ccplot on 64 bit Python.

In addition to the previously mentions changes, VOCAL’s pan feature could also be improved. Current implementation is constricted by the sheer amount of data to display on the screen. The typical CALIPSO plot contains 60,000 ticks of time data, each tick having a list of specific pixel values for each tick of altitude. Most useful data comes from the first 20 km, however with so many single points of time data a 32-bit python program simply does not have enough RAM to load all of the CALIPSO data. The current pan implementation divides the CALIPSO data into pages, the largest allowing only 15,000 ticks of time data to be displayed. When the user pans CALIPSO data, VOCAL loads the next page to display the new subsection of data. However, this operation is time consuming as VOCAL must load millions of data points. When the user pans, he or she sees a choppy animation as the plot moves horizontally. Because of this, the user has a hard time discerning the translation between dragging the plot and moving the plot horizontally. To improve panning, pages should be cached or preloaded so when the user pans the CALIPSO plot, the user sees smooth horizontal movement as the user drags the plot. This feature would require asynchronous operations and use of multiple threads to load the page into the cache.

The improvements listed may be accomplished in another term. However, since VOCAL is an open source project, anyone can contribute and improve VOCAL. The source code is hosted as a repository on GitHub, a popular version control system. Users can make changes and improvements to VOCAL, such as bug fixes or new features. To share these modifications, the user can send a pull request to the repository. Other users can comment and discuss whether to include the modifications into the source code and make it a standard feature within VOCAL. The owners of the repository can decide whether or not merge these changes. Figure 6 shows an example pull request.

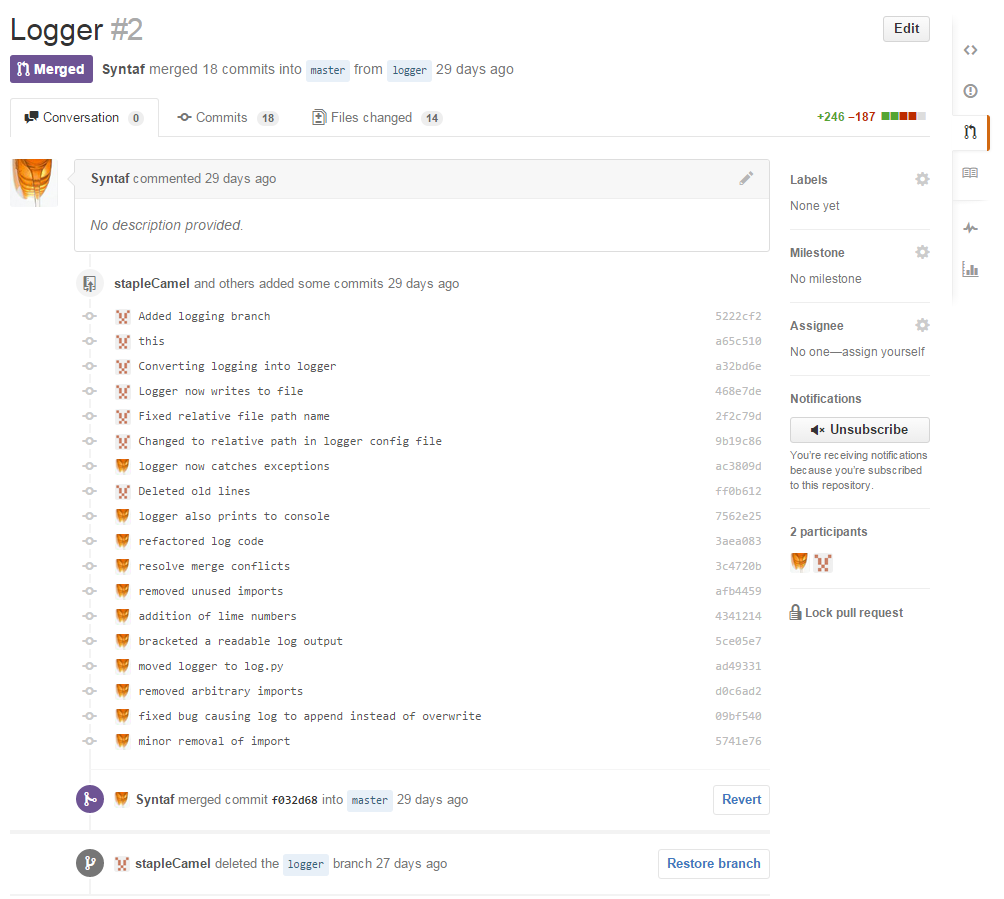


Figure 6. Example of a pull request

Additionally, GitHub provides convenient method of provided bug reports and requests. A scientist might find a bug in VOCAL, but may not have any programming knowledge to fix the bug. The scientist can post an issue on VOCAL’s GitHub page, where other users can see the issue, discuss it, and address it. VOCAL has a logger that reports on internal operations. If there is a crash or a bug, the scientist can post the log trace within the issue to help programmers and the owner fix the issue. Figure 7 demonstrates how issues work on GitHub. Issues posted on GitHub can also be requests for new features. If a scientist wants a new feature added into VOCAL and does not know how to implement it, he or she can post an issue on GitHub requesting this feature. Like any other issue, every user can view this issue and discuss whether or not to implement the request. The open source nature of VOCAL allows every user to collaborate on VOCAL’s development, regardless of programming knowledge. This allows scientists to easily adapt VOCAL to their needs and goals.

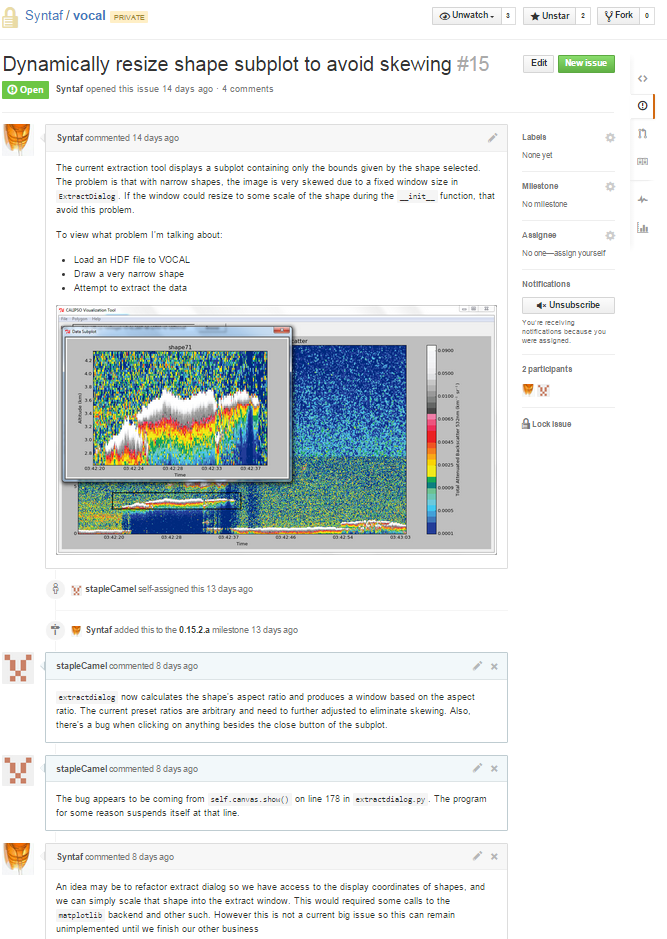


Figure 7. Example issue

# V. Conclusions

With the completion of VOCAL, Earth scientists will finally have a customizable and open source tool to help them analyze the impacts of aerosols on Earth’s climate and human health. Aerosols are a major factor in Earth’s climate; they can affect cloud formation and cloud radiative properties. The CALIPSO satellite, launched in May 2006, measures the global distribution of aerosols in Earth’s atmosphere and allows climate scientists to analyze aerosols and determine their potential impact on climate and human health. To view CALIPSO data, researchers used a tool written in IDL. However, this IDL tool is difficult to modify and lacked key functionality that would have aided researches in identifying aerosols’ role in climate and air quality.

VOCAL is a major improvement over the old IDL tool, and includes new features, such as shape drawing and database sharing. VOCAL, written in Python, implements some of the old IDL tool’s features and adds additional functionality. The shape-drawing feature will allow scientists to highlight specific aerosol features of interests. The shapes additionally provide scientists the ability to view the data bounded by the shapes, enabling scientists to extract subsections of CALIPSO data. Researchers can then use these shapes to predict the trajectory of these aerosols within the Earth’s atmosphere. After selecting all aerosol features, the scientists can then share these shapes on the database, so that other researchers can view these aerosols and collaborate remotely on projects. Additionally, since VOCAL is an open source software, Earth scientists can change and modify the tool to meet their needs, another major improvement over the old IDL tool.

# VI. Acknowledgments

CALIPSO Health and Air Quality LaRC Spring 2015

* Jordan Vaa (Team Lead)
* Ashna Aggarwal
* Courtney Duquette

CALIPSO Science Team

* Dr. Charles Trepte
* Dr. Amber Soja

NASA DEVELOP National Program

* Jeffry Ely
* Dr. Kenton Ross

This material is based upon work supported by NASA through contract NNL11AA00B and cooperative agreement NNX14AB60A.

# VII. References

Omar, Ali H., David M. Winker, Mark A. Vaughan, Yongxiang Hu, Charles R. Trepte, Richard A. Ferrare, Kam-Pui Lee, Chris A. Hostetler, Chieko Kittaka, Raymond R. Rogers, Ralph E. Kuehn, and Zhaoyan Liu. "The CALIPSO Automated Aerosol Classification and Lidar Ratio Selection Algorithm." *Journal of Atmospheric and Oceanic Technology J. Atmos. Oceanic Technol.* 26.10 (2009): 1994-2014. *Web of Science*. Web. 17 June 2015.

Winker, D. M., J. Pelon, J. A. Coakley, S. A. Ackerman, R. J. Charlson, P. R. Colarco, P. Flamant, Q. Fu, R. M. Hoff, C. Kittaka, T. L. Kubar, H. Le Treut, M. P. Mccormick, G. Mégie, L. Poole, K. Powell, C. Trepte, M. A. Vaughan, and B. A. Wielicki. "The CALIPSO Mission: A Global 3D View of Aerosols and Clouds." *Bulletin of the American Meteorological Society Bull. Amer. Meteor. Soc.* 91.9 (2010): 1211- 229. *Web of Science*. Web. 17 June 2015.

Winker, David M., Jacques Pelon, and Patrick McCormick. “The CALIPSO mission: Spaceborne Lidar for Observation of Aerosols and Clouds.” *SPIE Proceedings* 4893 (2003): 1-11. *SPIE*. 24 Mar. 2003. Web. 17 June 2015. <http://proceedings.spiedigitallibrary.org/>.

Winker, David M., Mark A. Vaughan, Ali Omar, Yongxiang Hu, Kathleen A. Powell, Zhaoyan Liu, William H. Hunt, and Stuart A. Young. "Overview of the CALIPSO Mission and CALIOP Data Processing Algorithms.” *Journal of Atmospheric and Oceanic Technology J. Atmos. Oceanic Technol.* 26.11 (2009): 2310-323. *Web of Science*. Web. 17 June 2015.

# VIII. Content Innovation

Database Linking Tool

Inline Supplementary Material containing some of VOCAL’s source code

[To be added]

# IV. Appendices

Insert here