OSGeo
Your Open Source Compass

OSGeo Journal Volume 8

February 2011

FOSS4G 2009 Conference Proceedings

OSGeo Community

- News & Announcements
- Case Studies
- Integration Examples

FOSS4G

DENVER 2011

SEPTEMBER 12-16

The Annual International

FREE & OPEN SOURCE
SOFTWARE FOR GEOSPATIAL

Conference Event

2011.FOSS4G.ORG

OSGeo

Your Open Source Compass

Volume 8 Contents

From the Editor

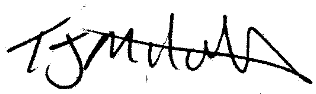
OSGeo has just past its 5th birthday, along with this 8th volume of the OSGeo Journal! With this edition we bring a few news headlines from the past couple months, a few general articles and, most significantly, several top papers from the **FOSS4G 2009** conference event held in Sydney, Australia.

The Journal has become a diverse platform for several groups and growth in each area is expected to continue. The key groups that read and contribute to the Journal include software developers sharing information about their projects or communities, power users showing off their solutions, academia seeking to publish their research and observations in a peer-reviewed, open source friendly medium. OSGeo also uses the Journal to share community updates and the annual reports of the organisation.

Welcome to those of you who are new to the OSGeo Journal. Our Journal team and volunteer reviewers and editors hope you enjoy this volume. We also invite you to submit your own articles to any of our various sections. To submit an article, register as an "author" and sign in at http://osgeo.org/ojs. Then when you log in you will see an option to submit an article.[1]

We look forward to working with, and for, you in the upcoming year. It's sure to be an interesting year as we see OSGeo, Open Source in general and all our relate communities continue to grow. Nowhere else is this growth more apparent than at our annual conference: **FOSS4G 2011 Denver**, September, 2011.[2] Keep an eye on your OSGeo mailing lists, blogs and other feeds to follow the latest FOSS4G announcements, including the invitation to submit presentation proposals.[3] It will be as competitive as ever to get a speaking slot, so be sure to make your title and abstract really stand out.

Wishing you the best for 2011 and hoping to see you in Denver!



Tyler Mitchell
tmitchell@osgeo.org
Editor in chief, OSGeo Journal
Executive Director, OSGeo

[1] The direct URL for article submission is: https://www.osgeo.org/ojs/index.php/journal/author/submit
[2] FOSS4G 2011 Denver: http://2011.foss4g.org
[3] FOSS4G 2011 Abstract Submission: http://2011.foss4g.org/program

FOSS4G 2009 Conference Proceedings

From the Academic Track Chair

Prof. Thierry Badard

The FOSS4G 2009 academic track aimed to bring together researchers, developers, users and practitioners – all who were carrying out research and development in the free and open source geospatial fields and who were willing to share original, recent developments and experiences.

The primary goal was to promote cooperative research between OSGeo developers and academia, but the academic track has also acted as an inventory of current research topics. This track was the right forum to highlight the most important research challenges and trends in the domain and let them become the basis for an informal OSGeo research agenda. It has fostered interdisciplinary discussions in all aspects of the free and open source geospatial domains. It was organized to promote networking between the participants, to initiate and favour discussions regarding cutting-edge technologies in the field, to exchange research ideas and to promote international collaboration.

In addition to the OSGeo Foundation[23], the ICA (International Cartographic Association) working group on open source geospatial technologies[24]) was proud to support the organisation of the track.

The coordinators sought to gather paper submissions globally that addressed theoretical, technical, and practical topics related to the free and open source geospatial domain. Suggested topics included, but were not limited to, the following:

- State of the art developments in Open Source GIS
- Open Source GIS in Education
- Interoperability and standards - OGC, ISO/TC 211, Metadata
- Spatial Data Infrastructures and Service Oriented Architectures
- Free and open source Web Mapping, Web GIS and Web processing services
- Cartography and advanced styling
- Earth Observation and remote sensing
- Spatial and Spatio-temporal data, analysis and integration
- Free and Open Source GIS application use cases in Government, Participatory GIS, Location based services, Health, Energy, Water, Urban and Environmental Planning, Climate change, etc.

In response to the call for papers, 25 articles were submitted to the academic track. The submissions were highly diversified, and came from USA, Canada, Thailand, Japan, South Korea, Sri Lanka, Australia, New Zealand, Italy, Denmark, France, Germany, Switzerland, Romania and Turkey. Selection of submissions were based on the full papers received. All submissions were thoroughly peer reviewed by two to three members of the international scientific committee and refereed for their quality, originality and relevance. The scientific committee selected 12 papers (48% acceptance rate) for presentation at the FOSS4G 2009 conference. From those, 6 papers were accepted for presentation in the proceedings of the academic track, which are published in this volume of the OSGeo Journal. They correspond to the 6 best papers assessed by the international scientific committee.

The accepted and published papers covered a wide

[23]OSGeo: Open Source Geospatial Foundation: http://osgeo.org
[24]ICA open source working group: http://ica-opensource.scg.ulaval.ca/

range of cutting-edge research topics and novel applications on Free and Open Source Geospatial technologies. I am particularly proud and happy to see some very high quality scientific contributions published in the OSGeo Journal. This will undoubtedly encourage more interesting research to be published in this volume, as our OSGeo journal is an open access journal. In addition, it helps draw attention to this important project of the OSGeo Foundation. I hope the publication of these proceedings in the OSGeo journal will encourage future scientists, researchers and members of academia to consider the OSGeo Journal as an increasingly valuable place to publish their research works and case studies.

As a concluding note, I would like to take the opportunity to thank the individuals and institutions that made the FOSS4G 2009 academic track possible. First, I would like to thank the international scientific committee members and external reviewers for evaluating the assigned papers in a timely and professional manner. Next, I would like to recognize the tremendous efforts put forward by members of the local organising committee of FOSS4G 2009 for accommodating and supporting the academic track. Finally, I want to thank the authors for their contributions, efforts, patience and support that made this academic track a huge success.

January, 2011
Prof. Thierry Badard
Laval University, Canada
Chair, FOSS4G 2009 Academic Track
Co-chair, ICA Working Group on Open Source Geospatial Technologies

A Data System for Visualizing 4-D Atmospheric CO2 Models and Data

Tyler A. Erickson, Anna M. Michalak, John C. Lin

Abstract

This paper describes a geospatial data system that produces KML representations of complex spatio-temporal datasets related to modeling the atmospheric carbon cycle. KML is an open standard language for transferring annotated geospatial data that can be used by many modern geospatial software packages, particularly virtual globe applications. The server component of the data system is built using a variety of open source software packages, which provide flexibility for creating custom geospatial representations of the datasets. The paper shows examples of how KML representations of atmospheric CO2 datasets and model outputs can be visualized with virtual globe client applications, allowing a diverse group of users to explore the complex scientific datasets that are central to the discussion of climate change and global warming.

Introduction

The general population's awareness of, and interest in, climate change has increased dramatically in recent years. The consensus among climate scientists is that climate change is occurring, and that there is "very high confidence (9 out of 10 chance of being correct) that the global average net effect of human activities since 1750 has been one of warming" (19). Although the need for further work on specific scientific aspects remains, the discussion has largely progressed from "is climate change occurring?" to "how will climate change progress in the future?" and "how can human society mitigate or adapt to climate change?"

Several factors affect the energy balance of the climate system, including greenhouse gases, aerosols, and land surface properties. Of all the components, the increase in the atmospheric concentration of carbon dioxide (CO2) has been responsible for the largest increase in radiative forcing, or tendency to warm the Earth's surface. In 2007, the International Panel on Climate Change (IPCC) published a summary report stating that "carbon dioxide is the most important anthropogenic greenhouse gas that contributes to climate change" (19).

CO2 is continuously exchanged between the atmosphere and the Earth's surface, including land and oceans. The rate of exchange, or flux, is spatially and temporally variable, with this variability itself changing across scales. Overall, approximately half of current anthropogenic emissions of CO2 are taken up by land and oceans, which act as natural carbon "sinks." However, there is a lack of understanding of where these sinks occur, how they vary in time, and how they are affected by climate variability and other processes. This, in turn, limits the skill of existing models in predicting future changes in net carbon balance (12), and, therefore, the future atmospheric concentrations of CO2. Because CO2 flux can only be measured directly at relatively small spatial scales and at a limited number of sites (e.g. (2)), the estimation of CO2 fluxes on regional to continental scales relies heavily on models and indirect measurements.

One approach that carbon cycle scientists use to characterize the spatial and temporal variability of CO2 fluxes is to relate concentrations of atmospheric CO2 measured in the atmosphere to fluxes occurring in upwind regions, through a process called inverse modeling (e.g. (10)). This approach couples atmospheric CO2 observations with numerical models describing winds and weather patterns, and any additional information relevant to estimating carbon exchange, in order to trace fluctuations in atmospheric concentration measurements of CO2 backwards in space and time to the sources and sinks. This process allows scientists to characterize variability in upwind carbon exchange between the Earth's surface and the atmosphere.

Modeling and understanding the sensitivity of available CO2 observations to upwind fluxes is a key component of the inverse modeling framework. One approach, described in this paper, involves the use of a Lagrangian model, which simulates large numbers of particle trajectories backwards in time, starting at the location and time of the measurement, to identify the regions (in space and time) that influence the measured concentration. This modeling approach produces a diverse set of spatial and temporally varying datasets, with concentration measurement locations that are fixed in 3-D space and variable in time, particle trajectories that are variable in 3-D space and time, and sensitivity maps of measurements to surface fluxes that are variable in 2-D space and time.

The focus of this paper is the development of an approach for sharing these complex spatial and temporal datasets with a diverse set of users, ranging from the general public to carbon cycle scientists and decision makers. These spatial and temporal datasets can be difficult to explore and visualize, due to the high dimensionality of the data. An ideal tool for exploring these datasets would:

- possess capability for displaying 3-D spatially and temporally referenced data;

- have an easy-to-use interactive user interface, that allows the user to navigate the data in both space and time, and to query attributes of the data;
- have support for overlaying other georeferenced datasets; and
- be freely available and be compatible with commonly used software.

While many past approaches for visualizing complex geospatial data have some of these characteristics, all of the characteristics are important for communicating the data to a diverse set of users.

Fortunately, recent advances in geospatial software have reduced the barriers for visualizing 3-D and 4-D datasets. Virtual globes, which are interactive client applications, can be used to present custom 4-D datasets over a richly detailed reference model of the Earth (3). Accessing this type of visualization no longer requires advanced training in geospatial information systems or computer science, but rather is accessible by typical computer users. Google Earth, a popular virtual globe with a simple user interface, has been downloaded over 350 million times (31).

This paper presents a client-server geospatial information system that allows users to visualize several complex datasets used for understanding terrestrial carbon fluxes. The geospatial data server is built using free and open source software (FOSS) components that store, process, and format spatial and temporal datasets so that they can be easily visualized, using modern virtual globe software packages. The system described in this paper incorporates the desired characteristics listed above, and is built on a flexible platform that can be easily enhanced in the future.

Background

This section presents an overview of the scientific background for the models used to generate datasets included in the visualization and of open source software development.

Atmospheric CO2 Data and Modeling

Regular atmospheric measurements of CO2 began in 1958 with observations taken at Mauna Loa, Hawaii (19). Since that time, global atmospheric CO2 monitoring networks have expanded significantly. The National Oceanic and Atmospheric Administration (NOAA) Earth System Research Laboratory (ESRL) Global Monitoring Division (GMD) maintains the NOAA-ESRL Cooperative Air Sampling Network, which currently includes over 150 sites globally (28). More recently, NOAA-ESRL-GMD has developed a Tall Tower Network of sites with continuous observations of CO2 and related gases (29). This network focuses on the continental United States, and currently includes eight tall tower sites. Continuous measurements are particularly useful for inverse modeling studies, because they allow individual sampling locations to "see" large regions, as a function of changing wind directions and weather patterns.

Data from two of the Tall Tower Network sites are used for illustration purposes in the presented application. The first site is the LEF tower in Park Falls, Wisconsin, a 396m tower that has been operating since 1994. The second site is the 107m AMT tower in Argyle, Maine, which has been operating since 2003. In the presented analysis, the concentration measurements from these towers were averaged to 3-hour intervals over the time period of June 1 to July 8, 2004. Meteorological information derived from the WRF model and atmospheric trajectories from the STILT model (described later in this section) were used to calculate the influence of atmospheric CO2 arriving from outside the examined domain on the available observations, and this impact was pre-subtracted from observations. As a result, the CO2 concentration variations examined here are influenced only by carbon sources and sinks within the examined North American domain (15).

As discussed in Section 1, one method used by carbon cycle scientists to quantify the sensitivity of atmospheric measurements to surface fluxes involves simulating collections of particles backwards in time from the measurement location, while modeling the turbulent dispersion as a stochastic process (30). The Stochastic Time-Inverted Lagrangian Transport Model (STILT) is one such model that estimates subgrid particle movement by interpolating gridded meteorological fields to the location of the particle and parameterizing the turbulent motions as functions of these meteorological variables (23). The STILT model is an adaptation of the HYSPLIT trajectory model (7) but incorporates four key improvements:

- a modified turbulence scheme that ensures adherence to the "well-mixed criterion", a manifestation of the 2nd Law of Thermodynamics (49);
- close coupling to atmospheric models to minimize deviations from mass conservation (27);
- capability to account for errors in the meteorological fields using a Monte Carlo method (22);
- a revised method for estimating the height of the planetary boundary layer (PBL) that generalizes to unstable, neutral, and stable conditions (50). The PBL is the lower portion of the atmosphere in which trace gas concentrations are most sensitive to surface fluxes (47).

The STILT model runs can utilize gridded meteorological datasets from a variety of sources, including high-resolution limited area models like the Regional Atmospheric Modeling System (RAMS) (5) and the Weather Research and Forecasting (WRF) model (45). For this study, WRF v2.2 was used to generate meteorological fields used by STILT (27). A 3-level nested grid domain was used, with base grid of 40-km spanning 10°N to 70°N, and 170°W to 50°W with higher resolution grids of 10-km resolution over the Eastern half of the United States, and 2-km resolution grids surrounding three tall towers, including the two examined in the current study. 10 day back-trajectories of 500 particles per hour from each receptor (tall tower location) were simulated. From these trajectories, a temporal grid was produced, representing the sensitivity of atmospheric concentrations to upwind surface fluxes on a 3-hour time interval and a 1°×1° grid. The sensitivities were derived using times and locations where the particles were below the planetary boundary layer, indicating that the air parcel is sensitive to fluxes occurring at that location. These sensitivity grids, a.k.a. *footprints*, provide the linkage between locations measuring atmospheric concentrations with upwind fluxes.

Biospheric fluxes play a critical role in the carbon cycle and atmospheric CO2 concentrations, imposing marked diurnal and seasonal cycles on CO2 variations. The terrestrial biosphere absorbs CO2 through photosynthesis during the daytime of the growing season and releases CO2 back to the atmosphere through respiration during the nighttime and the winter season (43). To simulate these biospheric processes, a wide variety of biospheric models have been developed over the past several decades (44).

In this project, CO2 surface fluxes generated using the CASA terrestrial carbon cycle model (39) were used, which simulates ecosystem processes and is driven by satellite observations and meteorology. The surface fluxes, which were presented in Olson and Randerson (32), were mapped every 3 hours and at 1°×1° resolution, matching the sensitivity datasets.

The biospheric processes controlling carbon dioxide fluxes are complex functions of a large number of variables, including solar radiation, temperature, vegetation type, nutrient availability, disturbance history, and soil moisture, among other factors. The interactions between these variables result in a heterogeneous distribution of fluxes that vary both in space and time.

Geovisualization

Geovisualization is an emerging field that includes approaches from a variety of disciplines, including cartography, scientific visualization, exploratory data analysis, and geographic information science, which provide tools for the visual exploration, analysis, synthesis and presentation of data that contain geographic information (9, 24). Geovisualization systems may include support for temporal information, which is often lacking in traditional geographic information systems (26) but may be of critical importance for understanding temporally-variable environmental datasets.

Many geovisualization systems allow for significant levels of interaction, which allows users to explore data, synthesize, confirm and communicate ideas through guided discovery (8). In recent years the general public has become familiar with interactive visualization in several forms such as online maps used for driving directions and virtual globes used by television news programs to give spatial context to remote events.

The scientific community has begun to utilize freely available virtual globe applications as geovisualization tools to communicate scientific results (3) such as meteorological data (46), disease transmission observations (18), and vertical profile data obtained from satellite sensors (4). Many virtual globes can display custom user content that is spatially (and may also be temporally) referenced, provided the data is in a standard format.

The Keyhole Markup Language (KML) is an open standard XML-based language for exchanging georeferenced feature data, styling, and annotation. KML was originally created by Keyhole, Inc. and further expanded by Google after it acquired Keyhole, Inc. in 2004. The KML 2.2 specification was submitted to the Open Geospatial Consortium (OGC) standards organization, and became an official OGC standard (the OpenGIS KML Encoding Standard) on April 14, 2008 (34). The adoption of the standard by the OGC should encourage the development of visualization clients and server software applications that use KML to exchange spatial and temporal data.

One import feature of the KML language is the ability to access additional KML-formatted data at a specified URL using the Network Link functionality. This allows KML viewers to hierarchically link to large external datasets stored on or generated by remote servers. This paper describes a data system, built with open source software components, that produces KML formatted data with spatial and temporal attributes. Open source is a software development method that allows wide accessibility to the software's code base for use, improvement, and redistribution in modified or unmodified forms.

Open source software has been adopted in many areas of academic research, such as the R language for statistical analysis (16). Rey (42) describes past interactions between academic geospatial researcher and the open source communities, as well potential opportunities for future cross-collaboration. In many ways, the open source software development process is similar to the scientific process of knowledge development (21) in that it promotes peer review by external developers (37) and the open source licenses allow for continuous enhancements and improvements to a body of knowledge.

Data System

A flexible three-tier data system architecture was designed to enable visualization of the carbon cycle datasets (Figure 1). The database and application tiers store, manipulate, and format the carbon cycle datasets as requested by client applications via a HTTP connection, enabling access by any client computer that has access to the application server. The database and application tier are created using open source geospatial components, and communicate with the client applications using KML, an open standard language for transferring geospatial data.

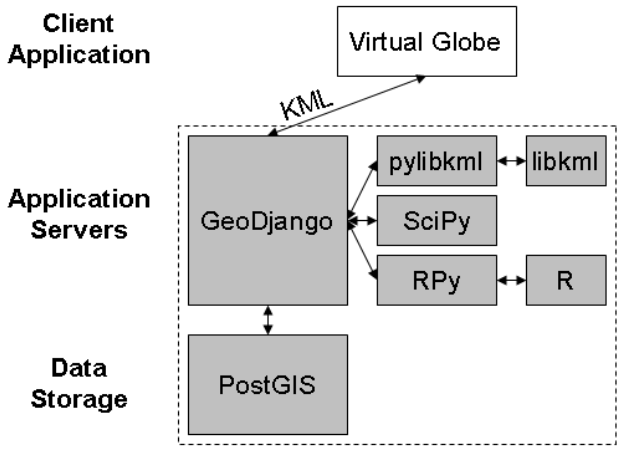


Figure 1: Overview of the open source data system components (shaded grey) and their relationship to the Google Earth virtual globe client application. The core of the application server, GeoDjango, uses the functionality of several open source libraries for formatting and manipulating data.

Data Storage

The foundation tier of the data system is PostGIS (41), a spatial extension to PostgreSQL (38), a client-server relational database. The PostGIS extension adds geographic data types and spatial operators to PostgreSQL, which enables the database to store spatial, temporal and attribute information as records. PostGIS was selected because it is a widely used, free and open source, and follows the *Simple Feature Access Specification for SQL* (35), which is an open international standard for storing and accessing geographic features.

Datafile Format	Packages Used	Dataset Examples
text file	SciPy (1)	land-water mask
MATLAB (48) data file	SciPy	sensitivity maps, biospheric flux maps
R formatted data file	RPy (25) & R (16)	particle locations

Table 1: Summary of the open source packages used to import datafiles.

Application Server

The GeoDjango web framework is the core of the data system. GeoDjango is an integration of the Django (6) web framework with several open source geospatial libraries (GEOS, proj.4, GDAL) that includes support for spatial databases and provides spatial processing functionality. GeoDjango exposes the geometric data types and operators provided by PostGIS at the database level. GeoDjango is written in Python (40), a general-purpose object-oriented programming language that can be used for many kinds of software development and is known for its code readability and its ability to integrate with other languages and tools. GeoDjango is a web server that shares content using the Hypertext Transfer Protocol (HTTP). While the content is typically a web page, for this data system we have customized GeoDjango to serve KML documents.

The functionality of GeoDjango is enhanced by integrating with several open source libraries, as shown in Figure 1. The libkml C++ library (20) provides a structured way of authoring valid KML documents. The pylibkml library (11) is a Python wrapper for libkml, which simplifies the use of the libkml objects from within Python code. Key elements of the KML language that are used by this data system are: (1) features that have temporal attributes to denote a specific time or interval of time, (2) network link elements that allow for accessing remote datasets, and (3) model elements that can incorporate 3-D models in the virtual globes to symbolize attributes.

Measurement and model data used by the data system originate from a variety of sources and occur in a variety of formats. Several open source packages, used by the data system, are summarized in Table 1.

Client Application

The client application's role is to present spatially and temporally referenced data to the user. Although any application that implements the OGC KML standard could be used to view the data, the visualizations shown in this paper were produced using the Google Earth virtual globe (version 5.0) (14).

Application: Visualizing Atmospheric CO2 Data

This section describes how the data system was configured to manage the atmospheric CO2 datasets.

Model Representation

The atmospheric transport and biospheric models produce datafiles with numerous attributes. Data models were created in the GeoDjango framework to describe data objects, their attributes, and relationships between

Conceptual Object	Description	Spatial & Temporal Attributes
Sensor	A tall tower measurement location, which includes the static 3D location of the sensor.	3D point
Particle	A single simulated particle that represents a parcel of air at a specific time that will later arrive at the sensor location	time instant
Location	A simulated 3D point location of a particle at a specified time that corresponds to a specified concentration measurement. Also includes the height of the planetary boundary layer.	3D point; time; corresponding to a specific time interval (measurement)
Sensor Measurement	Average CO2 concentration measured over a time interval	time interval (measurement)
Sensitivity	The sensitivity of a measurement to a surface flux	time interval (measurement) & time interval (surface flux)
Surface Region	A discretized portion of the Earth's surface	2D polygon
Surface Region Flux	A modeled surface flux value	time interval (surface flux)

Table 2: Summary of the GeoDjango data model objects used to manage the measured and modeled datasets.

objects. Table 2 gives a summary of the conceptual objects that are used in this visualization and highlights the spatial and temporal attributes of each object, while Figure 2 shows how the conceptual data objects are interrelated.

In conjunction with the imported data, GeoDjango uses the data model objects to create, populate, and save instances of the data records to the PostGIS database. Similarly, in order to access the data for the visualizations, GeoDjango uses the data model objects to query data stored in the PostGIS database.

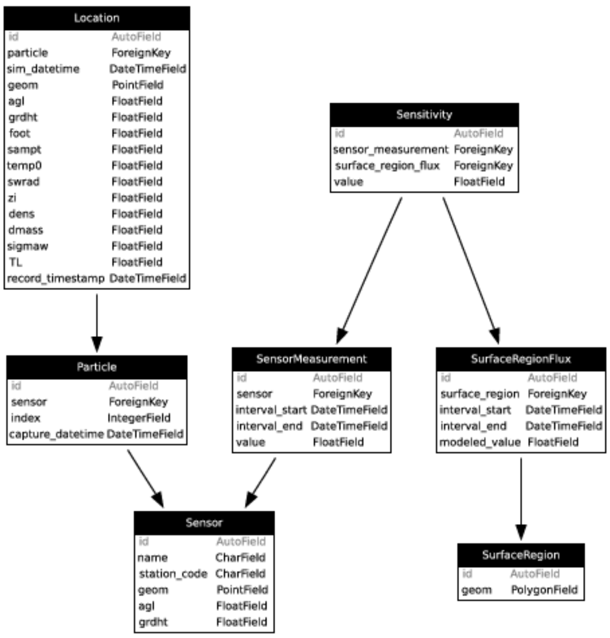


Figure 2: Overview of the GeoDjango data models and their relationships that are used to model the CO2 measurement and model output datasets.

Atmospheric Measurements of CO2

The data for the atmospheric CO2 measurements, described in the background section, was obtained as MATLAB data files (one per measurement sensor). A Python script is used to parse the data files and import records into the PostGIS data tables, using the SensorMeasurement object of the GeoDjango data models (Figure 2). KML-formatted representations of the CO2 concentrations are obtained by submitting a URL request to the GeoDjango server application. For example, a request for a KML representation of a measurement data series for the LEF Tall Tower sensor between June 1, 2004 and August 8, 2004 would be:

`http://localhost/measurement/station=LEF/start=2004-06-01T00:00:00Z/end=2004-07-08T00:00:00Z/series.kml`

An example rendering of the CO2 concentration relative to background measured at the LEF Tall Tower is shown in Figure 3(a), and an example KML representation is shown in Table 3. The KML model representation uses the <*Location*> element to set the 3-D position and the <*Link*> element to reference an external COLLADA model of a unit-sized, colored sphere (green_sphere.dae). The volume of the sphere is set to be proportional to the absolute value of the difference between the measured and background concentrations using the <*Scale*> element. CO2 concentrations that are greater (less) than the background concentration are symbolized in green (blue). Although KML is itself an OGC standard, this encoding of the sensor data does not conform to any of the current OGC Sensor Web Enablement specifications (33).

When a user navigates time using the time slider control in Google Earth, the color and size of the sphere change according to the selected time, conveying the temporal variability of the concentrations.

Table 3: Example excerpt of KML for displaying a CO2 measurement observation.

```
<Placemark>
 <name>2004-06-0101:00</name>
   ...
 <TimeSpan>
     <begin>2004-05-31T23:30:00Z</begin>
     <end>2004-06-01T02:30:00Z</end>
 </TimeSpan>
 <Modelid="model_1366">
     <altitudeMode>relativeToGround</altitudeMode>
     <Location>
        <longitude>-90.273157</longitude>
        <latitude>45.945048</latitude>
        <altitude>396</altitude>
     </Location>
     <Scale>
        <x>28.1536</x>
        <y>28.1536</y>
        <z>28.1536</z>
     </Scale>
     <Link>
        <href>green_sphere.dae</href>
     </Link>
 </Model>
</Placemark>
```

Table 4: Portion of the KML document that defines a particle location for a specified time and particle paths between consecutive locations. Although the location is valid for an instant in time for a moving particle, the KML <TimeSpan> tag for the particle location specifies a short interval to enhance the visualization. Paths between consecutive locations are extended to the ground surface.

```
<Document>
   <Styleid="style_particle">
      <IconStyle>
         <Icon>
            <href>http://maps.google.com/mapfiles/kml/shapes/shaded_dot.png</href>
         </Icon>
      </IconStyle>
   </Style>
   <Style id="style_particle_path_above_bnd">
      <LineStyle>
         <color>ff0000ff</color>
         <width>4</width>
      </LineStyle>
      <PolyStyle>
         <color>400000ff</color>
      </PolyStyle>
   </Style>
   <Style id="style_particle_path_below_bnd">
      <LineStyle>
         <color>ff00ff00</color>
         <width>4</width>
      </LineStyle>
      <PolyStyle>
         <color>4000ff00</color>
      </PolyStyle>
   </Style>
   <Folder>
      <name>locations</name>
      <visibility>0</visibility>
      <open>1</open>
      <Placemark>
         <visibility>0</visibility>
         <TimeSpan>
            <begin>2004-06-03T04:30:00Z</begin>
            <end>2004-06-03T04:50:00Z</end>
         </TimeSpan>
         <styleUrl>#style_particle</styleUrl>
         <Point>
            <extrude>1</extrude>
            <altitudeMode>relativeToGround</altitudeMode>
            <coordinates> -83.0984,77.8721,4240.77</coordinates>
         </Point>
      </Placemark>
   </Folder>
      <Folder>
      <name>path</name>
      <Placemark>
         <visibility>0</visibility>
         <TimeSpan>
            <begin>2004-06-03T04:40:00Z</begin>
            <end>2004-06-10T19:00:00Z</end>
         </TimeSpan>
         <styleUrl>#style_particle_path_above_bnd</styleUrl>
         <LineString>
            <extrude>1</extrude>
            <tessellate>1</tessellate>
            <altitudeMode>relativeToGround</altitudeMode>
            <coordinates>  -83.0984,77.8721,4240.77  -83.1489,77.8293,4255.398
                                                                       </coordinates>
         </LineString>
      </Placemark>
      <Placemark>
         ...
   </Folder>
</Document>
```

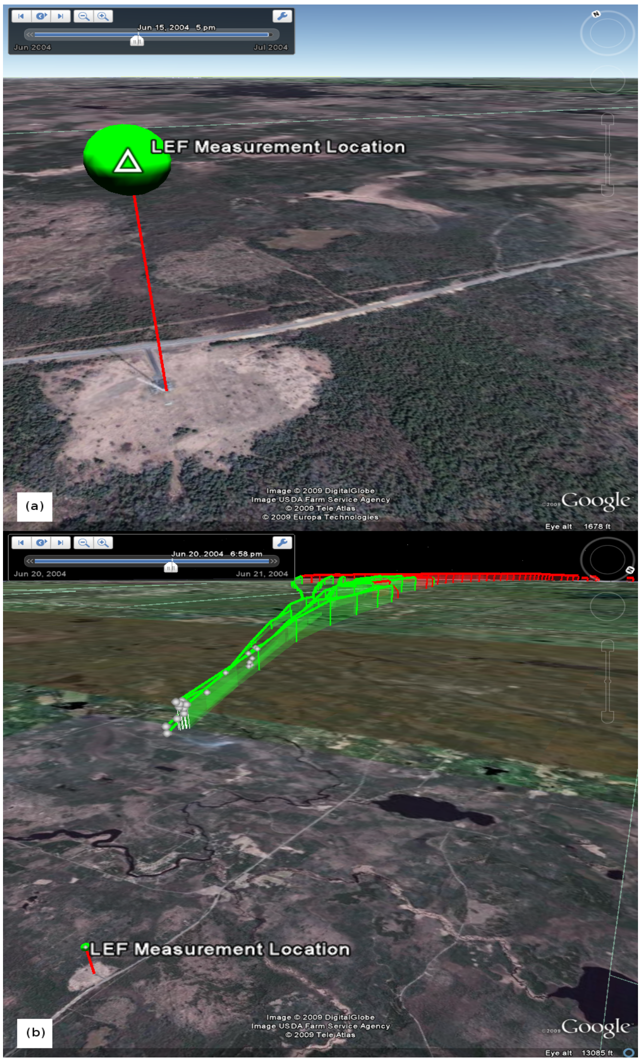


Figure 3: Screenshots of Google Earth renderings of the KML generated by the data system. (a) CO2 concentrations measured at the LEF Tall Tower. The volume of the sphere is proportional to the difference between the measured and the background concentrations. The sphere color (green/blue) is used to denote whether the measured concentrations are below/above the background concentration. (b) Simulated particle locations (grey) representing air parcels that are sampled by the LEF Tall Tower. The paths between simulated locations are colored green/red to indicate whether the particle is below/above the atmospheric boundary layer.

Air Parcel Simulation

The source of the particle location data are R-formatted datafiles produced by the STILT atmospheric transport model. A Python script is used to parse the data files and import records into the PostGIS data tables, using the GeoDjango data models.

KML-formatted representations of the particle data are obtained by submitting a URL request to the GeoDjango server application. For example, a request for single particle trajectory captured by the LEF Tall Tower measurements on 2004-06-10 at 19:00 (UTC) is:

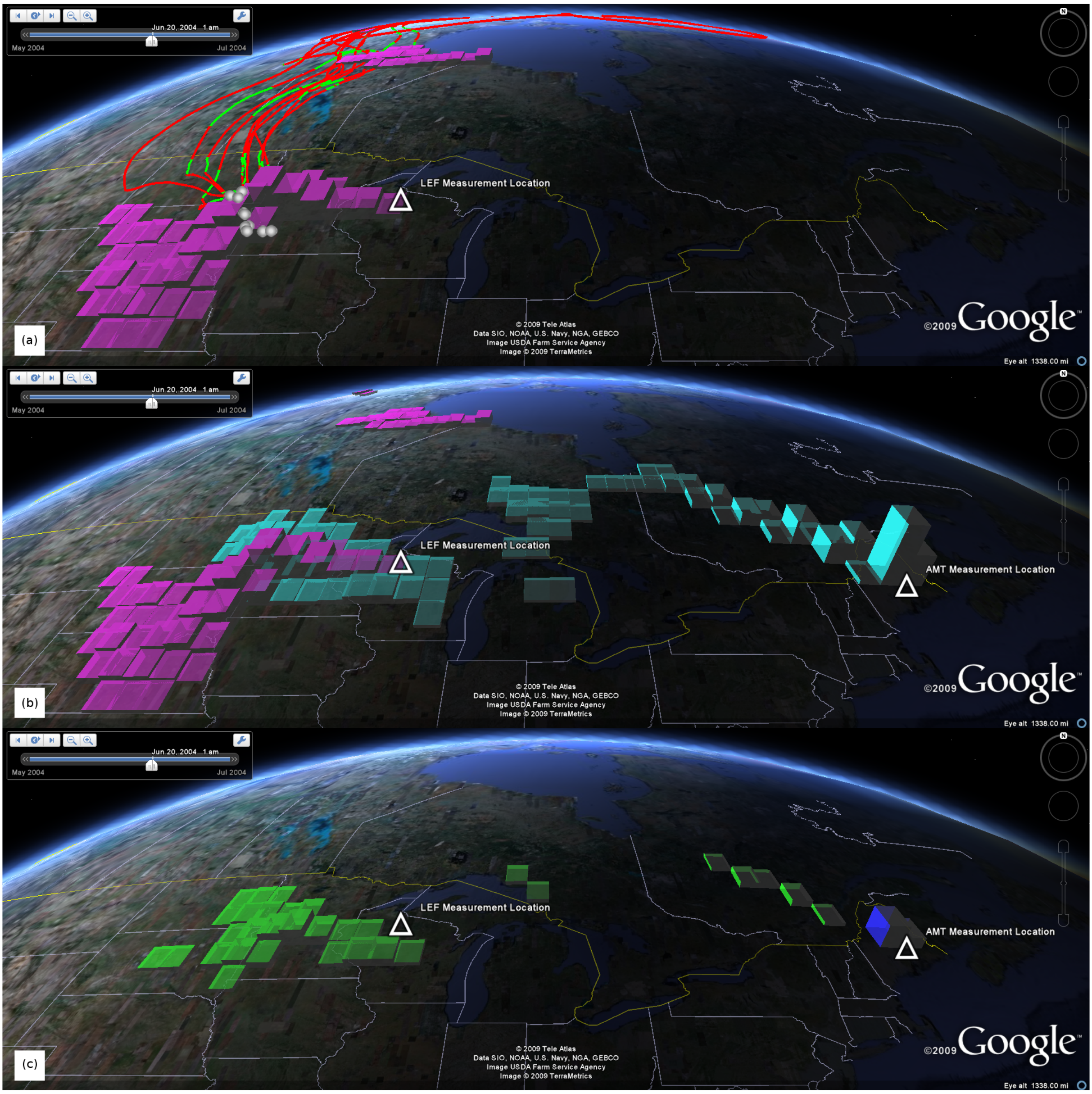


Figure 4: Screenshots of Google Earth renderings of the KML generated by the data system. The screenshots all represent the conditions at the same time: 6/20/2004 at 1:00 (UTC). (a) Particle locations (grey) and trajectories (red and green) for air parcels that are sampled by the LEF Tall Tower, along with the estimated sensitivity (purple) of the LEF tower measurements to surface fluxes. The height of the column is proportional to the sensitivity. (b) The overlaid sensitivity maps for the LEF and AMT tall towers. The sensitivity of CO2 measurements taken at the LEF/AMT tall tower to surface flux are symbolized in purple/cyan. (c) The sensitivity of CO2 measurements to the modeled biospheric flux of CO2 for measurements taken at the LEF and AMT tall towers. The height of the columns is proportional to the sensitivity multiplied by the model biospheric flux. Fluxes from the ground surface to atmosphere (respiration) are symbolized blue, while fluxes from the atmosphere to the ground surface (photosynthesis) are symbolized green.

`http://localhost/particle/station=LEF/capture=2004-06-10T019:00:00Z/index=1/track.kml`

A user may also request a series of particle trajectories. For example, a request for a series of 100 particle trajectories is:

`http://localhost/particle_tracks/station=LEF/capture=2004-06-10T19:00:00Z/index_start=1/index_end=100/track.kml`

Sample KML representations of the particle locations and paths between consecutive locations are shown in Table 4. An example Google Earth rendering of the particle tracks KML corresponding to measurements collected at the LEF Tall Tower is shown in Figure 3(b) and a screenshot of several paths is shown in Figure 4(a). The particle paths are symbolized, using the <*StyleUrl*> and <*Style*> elements, according to whether the particle location is above (red) or below (green) the atmospheric boundary layer. Particle locations that are below the atmospheric boundary layer contribute to the sensitivity of the sensor measurements to surface flux. Similarly, the path between consecutive particle locations is constructed with posts extending from each measurement location to the ground surface.

Sensitivity Maps

Spatially discretized sensitivity maps are produced by aggregating particle locations that are below the atmospheric boundary layer for a specified time interval (flux time), and that correspond to a specified measurement time interval (measurement time).

Table 5: Portion of the KML document that symbolizes the sensitivity of CO2 measurements made at the LEF Tall Tower to a surface flux for a 1°×1° ground region and 3-hour time interval.

```
<Document>
  <Style id="style_positive">
    <PolyStyle>
      <color>bfff00ff</color>
      <outline>0</outline>
    </PolyStyle>
  </Style>
  <Folder>
    <name>LEF sensitivity</name>
    <Placemark>
      <TimeSpan>
        <begin>2004-06</begin>
        <end>2004-06-01T03:00:00Z</end>
      </TimeSpan>
      <styleUrl>#style_positive</styleUrl>
      <MultiGeometry>
        <Polygon>
          <extrude>1</extrude>
          <tessellate>1</tessellate>
          <altitudeMode>relativeToGround</altitudeMode>
          <outerBoundaryIs>
            <LinearRing>
        <coordinates> -96,44,15036.9 -95,44,15036.9 -95,45,15036.9 -96,45,15036.9
                -96,44,15036.9 </coordinates>
            </LinearRing>
          </outerBoundaryIs>
        </Polygon>
      </MultiGeometry>
    </Placemark>
    <Placemark>
      ...
</Document>
```

A sample KML representation of a sensitivity map element is shown in Table 5. The TimeSpan element refers to the time interval of the surface flux, and the symbolized map elements correspond to aggregate sensitivity of the all measurements taken 0-10 days after the surface flux time interval.

Examples of Google Earth screenshots rendering the sensitivity map KML files are shown in Figure 4(a) and 4(b). The extruded height of the sensitivity map elements is proportional to the sensitivity of the CO2 measurements to the surface flux, and the elements are uniquely colored to correspond to the measurement sensor (magenta – LEF Tall Tower; cyan – AMT Tall Tower). As can be seen by the overlapping regions shown on the center section of Figure 4(b), more than one measurement sensor may be sensitive to surface fluxes occurring for a particular region.

While the sensitivity maps describe regions that may have affected the concentration measurements, a more informative variable to visualize is the sensitivity value multiplied by a modeled CO2 flux. The bottom section of Figure 4(c) shows a Google Earth rendering of this variable using biospheric fluxes of CO2 estimated by the CASA biospheric model. The extruded height of the map elements is proportional to the absolute value of the sensitivity multiplied by the biospheric flux. The elements are colored according to the direction of the flux, with green indicating transfer of CO2 from the atmosphere to the land surface (photosynthesis), and blue indicating release from the land surface to the atmosphere (respiration).

Discussion

CO2 Visualizations

The use of a user-friendly virtual globe application, such as Google Earth, makes the datasets accessible to a wide group of users. One use of the data system has been to create data layers that can introduce carbon cycle science to non-specialists such as educators. A KML document that included many of the datasets discussed in the paper was selected as a winner of Google For Educators 2009 KML in Research competition because "it represented a novel and compelling representation of science using Google Earth and the KML language." [39] The ease of use of virtual globe applications enables other non-specialist groups, such as the general public and decision makers whom are neither familiar with carbon cycle science or geospatial information tools, to access and explore the datasets.

Although carbon cycle scientists have long had other tools such as mapping applications for understanding the datasets, this group of advanced users can also benefit from having access to visualization tools for viewing their data. Virtual globe applications can be used for exploratory data analysis, helping scientists to identify issues with their data that are difficult to detect using traditional tools. For example, the virtual globe interface allows users to easily change perspective to view both the 'forest' (carbon sensitivity variations across North America) and the 'trees' (particle tracks that the sensitivity values are based on). This has been used to identify potential issues with the simulation of individual particle tracks which are not apparent when viewing the sensitivity footprints at the continental scale. Also, due to the three-dimensional nature of trajectories and the spatially and temporally varying footprints, the visualization software described in this

[39] Ryan Falor (Google), personal communication, 2 March 2009

paper provides a valuable tool to probe changes due to dynamic changes in the wind patterns. The tool enables the user to investigate the dispersion pattern of trajectories and quickly reveals the land areas whose emissions are sampled by the trajectories, whenever they dip within the PBL.

A key advantage of using a virtual globe to visualize spatio-temporal data is the ability to interactively navigate the temporal aspect of the datasets. The ability to select a specific time and to play forward and backward in time allows users to explore the temporal variability in the 'footprints' of each concentration measurement location. This conveys to users the effect that constantly changing meteorological fields have on the potential information can be extracted from the concentration measurements, a concept that is difficult to fully convey with a non-temporal representation that can only contain data for a single pre-specified time interval. While past work has used geovisualization tools to create movies to present changes over time (13), the non-interactive nature of movies does not facilitate exploration of the data. The sensitivity dataset actually has two temporal dimensions, which cannot be directly represented in KML because KML objects can only contain a single primitive element for time.

The sensitivity dataset relates a concentration measured over a time interval (time 1) to a surface flux that occurred over a previous time interval (time 2). The approach taken in this paper is to generate a KML representation of sensitivity integrated over the entire time that concentrations were measured (time 1), so that the temporal primitive element in the KML refers to the time of the surface flux (time 2). This allows the user to use the Google Earth time slider to see variations in surface flux sensitivity. A complementary and equally useful approach would be to create a second KML representation that integrates the sensitivity data over the surface flux time (time 2), so that users could use the time slider to see variations in regions that a particular concentration measurement is sensitive to, regardless of when the surface flux occurred.

Using Open Source Software for Research

The data system described in this paper was designed to manage a variety of spatially and temporally referenced datasets, which is a typical need for scientists that monitor the Earth's environment. The data system was constructed using a variety of open source software components because of the numerous advantages that open source software has over proprietary components in terms of flexibility, maintainability, simplicity, and the developer community.

Because the source code is available, modifications can be made to extend the functionality provided by the component. If these modifications are contributed back to the open source project, they may be incorporated into the core product. This can be advantageous to the author of the modification, because future enhancements to the core product will be compatible with the modifications. This is particularly important for code developed as part of academic research projects, which generally have a finite project length and do not support long-term maintenance of software.

In the lead author's experience, the process of designing prototype geospatial applications for research with open source software is quite different than with closed source proprietary systems. When an issue is encountered with closed source software, a user is restricted in their options for resolving the issue because they are prevented from inspecting and modifying the source code. When developing with open source software, there is always a way forward. If the user has sufficient technical skills they can debug and fix the issue themselves, or if not they can hire someone to fix the issue. Online forums and chat groups for both types of systems provide support to programmers, but for open source projects there is less separation between the developers and the users of the software, resulting in quicker and more relevant answers to questions.

Google Earth, a freely available proprietary closed source application, was primarily chosen as the visualization client because of its support of the full KML specification and its availability on multiple operating systems (Linux, Mac, Windows). Other beneficial features are the easy-to-use interface, the direct access to detailed vector data layers and high-resolution imagery that provide spatial reference, and the availability of rendering effects such as atmosphere and sun options that enhance the user's perception of spatial and temporal change, However the KML documents created by this data system can be rendered by any virtual globe application that supports the full OGC-KML specification (36).

Conclusions

This paper has described a prototype data system for producing visualizations of datasets related to atmospheric CO2 modeling. The intent has been to present an example of using open source geospatial software to manage complex spatial and temporal data, and to produce datasets in an open standard format that can be viewed in virtual globe applications as well as used by other geospatial software. While this paper focused on datasets related to modeling the atmospheric CO2 cycle, the data management and visualization techniques are appropriate for other regional to global-scale spatial-temporal datasets.

Providing a means of visualizing spatial-temporal datasets is important step toward increasing the non-specialists' knowledge of the complex processes that cause climate change. Geovisualizations, such as those presented in this paper, can be used to familiarize the general public, decision makers, and future researchers with an understanding the current state of knowledge and challenges in modeling Earth's systems and predicting future responses.

The data system presented in this paper is a work in progress, with numerous enhancements envisioned. Additional KML representations of the sensitivity datasets could be added to allow users to visualize according to the time of concentration measurement (in addition to the currently implemented time of surface flux). The current approach of storing discretized spatial variables (i.e. sensitivity maps or biospheric flux maps) as polygons could be improved by implementing raster data storage. Additional complex datasets used for inverse modeling, such as best estimate maps and covariance matrices, could be included in the visualization.

Acknowledgements

The authors gratefully acknowledges Adam Hirsch for discussions related to particle tracking as well as Sharon Gourdji, Kimberly Mueller, Vineet Yadav, and Deborah Huntzinger who provided the modeling datasets used in this visualization. The modeling datasets were created with support from the National Aeronautics and Space Administration under grant NNX06AE84G *Constraining North American Fluxes of Carbon Dioxide and Inferring Their Spatiotemporal Covariances through Assimilation of Remote Sensing and Atmospheric data in a Geostatistical Framework* issued through the ROSES A.6 North American Carbon Program to the University of Michigan.

Tyler A. Erickson
Michigan Tech Research Institute
Michigan Technological University
3600 Green Court, Suite 100
Ann Arbor, MI 48105
tylerickson@gmail.com

Anna M. Michalak
Department of Civil and Environmental Engineering, and
Department of Atmospheric, Oceanic and Space Sciences
The University of Michigan
183 EWRE Building
Ann Arbor, MI 48109-2125
anna.michalak@umich.edu

John C. Lin
Department of Earth and Environmental Sciences
University of Waterloo
200 University Avenue West
Waterloo, ON, Canada N2L 3G1
jcl@uwaterloo.ca

Bibliography

[1] Anonymous (2009) SciPy - Scientific Tools for Python. http://www.scipy.org/ Accessed 29 June 2009
[2] Baldocchi D, Finnigan J, Wilson K, Paw U KT, Falge E (2000) On Measuring Net Ecosystem Carbon Exchange Over Tall Vegetation on Complex Terrain. Boundary-Layer Meteorology. 96(1):257-291
[3] Butler D (2006) Virtual globes: The web-wide world. Nature. 439(7078):776-778
[4] Chen A, Leptoukh G, Kempler S, et al. (2009) Visualization of A-Train vertical profiles using Google Earth. Computers & Geosciences. 35(2):419-427
[5] Cotton WR, Pielke Sr. RA, Walko RL, et al. (2003) RAMS 2001: Current status and future directions. Meteorology and Atmospheric Physics. 82(1-4):5-29
[6] Django Software Foundation (2009) Django | The Web framework for perfectionists with deadlines. http://www.djangoproject.com/ Accessed 28 June 2009
[7] Draxler RR, Hess GD (1998) An overview of the HYSPLIT_4 modelling system for trajectories, dispersion and deposition. Aust. Meteor. Mag. 47(4):295-308
[8] Dykes J (2005) Facilitiating Interaction for Geovisualization. In: Exploring geovisualization. Kidlington, Oxford, UK: Elsevier.
[9] Dykes J, MacEachren AM, Kraak MJ (2005) Exploring geovisualization. In: Exploring geovisualization. Kidlington, Oxford, UK: Elsevier.
[10] Enting IG (2002) Inverse problems in atmospheric constituent transport. Cambridge University Press
[11] Erickson TA, Kemker R (2009) pylibkml - a python wrapper for the libkml library. http://code.google.com/p/pylibkml/ Accessed 29 June 2009
[12] Friedlingstein P, Cox P, Betts R, et al. (2006) Climate Carbon Cycle Feedback Analysis: Results from the C4MIP Model Intercomparison. Journal of Climate. 19:3337
[13] Gardiner N (2006) High Definition Geovisualization: Earth and Biodiversity Sciences for Informal Audiences. In: Geographic Hypermedia. http://dx.doi.org/10.1007/978-3-540-34238-0_24 Accessed 26 June 2009
[14] Google Earth. http://earth.google.com/ Accessed 29 June 2009
[15] Gourdji S, Hirsch A, Mueller K, et al. (2009) Regional-scale geostatistical inverse modeling of North American CO2 fluxes - A synthetic data study. Atmospheric Chemistry and Physics, 9, 22407-22458.
[16] Ihaka R, Gentleman R (1996) R: A Language for Data Analysis and Graphics. Journal of Computational and Graphical Statistics. 5(3):299-314
[17] IPCC (2007) Summary for Policymakers. In: Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press.
[18] Janies D, Hill AW, Guralnick R, et al. (2007) Genomic Analysis and Geographic Visualization of the Spread of Avian Influenza (H5N1). Syst Biol. 56(2):321-329
[19] Keeling CD, Bacastow RB, Bain-Bridge AE, et al. (1976) Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii. Tellus. 28:538-551
[20] kml.mashbridge, kml.bent (2009) libkml - a KML library written in C++ with bindings to other languages. http://code.google.com/p/libkml/ Accessed 29 June 2009
[21] von Krogh G, Spaeth S (2007) The open source software phenomenon: Characteristics that promote research. Journal of Strategic Information Systems. 16(3):236-253
[22] Lin JC, Gerbig C (2005) Accounting for the effect of transport errors on tracer inversions. Geophys. Res. Lett. 32(1)
[23] Lin JC, Gerbig C, Wofsy SC, et al. (2003) A near-field tool for simulating the upstream influence of atmospheric observations: The Stochastic Time-Inverted Lagrangian Transport (STILT) model. Journal of Geophysical Research (Atmospheres). 108(D16):ACH 2-1
[24] MacEachren AM, Kraak MJ (1997) Exploratory cartographic vi-

sualization: advancing the agenda. Computers and Geosciences. 23(4):335-343

[25] Moreira W, Warnes GR (2009) RPy - A simple and efficient access to R from Python. http://rpy.sourceforge.net/ Accessed 29 June 2009

[26] Mountain D (2005) Visualizing, Querying and Summarizing Individual Spatio-Temporal Behavior. In: Exploring geovisualization. Kidlington, Oxford, UK: Elsevier.

[27] Nehrkorn T, Eluszkiewicz J, Wofsy S, et al. (2010) Coupled Weather Research and Forecast-Stochastic Time-Inverted Lagrangian Transport (WRF-STILT) Model. Meteorology and Atmospheric Physics, 107, 51-64.

[28] NOAA ESRL GMD (2009) Observation Sites >> Listing by Project. ESRL Global Monitoring Division. http://www.esrl.noaa.gov/gmd/dv/site/site_table.html Accessed 28 June 2009

[29] NOAA ESRL GMD (2009) NOAA ESRL GMD Tall Tower Network. ESRL Global Monitoring Division - Carbon Cycle Group. http://www.esrl.noaa.gov/gmd/ccgg/towers/ Accessed 26 June 2009

[30] Obukhov AM (1959) Description of Turbulence in Terms of Lagrangian Variables. Advances in Geophysics. 6:113

[31] Ohazama C (2008) Truly global. Google Lat Long Blog. http://google-latlong.blogspot.com/2008/02/truly-global.html Accessed 26 June 2009

[32] Olsen SC, Randerson JT (2004) Differences between surface and column atmospheric CO2 and implications for carbon cycle research. J. Geophys. Res. 109:D02301

[33] Open Geospatial Consortium, Inc. (2009) Sensor Web Enablement WG | OGC®. http://www.opengeospatial.org/projects/groups/sensorweb Accessed 29 July 2009

[34] Open Geospatial Consortium, Inc. (2008) OGC® Approves KML as Open Standard | OGC®. OGC Website. http://www.opengeospatial.org/pressroom/pressreleases/857 Accessed 26 June 2009

[35] Open Geospatial Consortium, Inc. (2006) Simple Feature Access - Part 2: SQL Option | OGC®. http://www.opengeospatial.org/standards/sfs Accessed 28 June 2009

[36] Open Geospatial Consortium, Inc. KML | OGC®. http://www.opengeospatial.org/standards/kml Accessed 29 July 2009

[37] Open Source Initiative (2007) Home | Open Source Initiative. Open Source Initiative website. http://www.opensource.org/ Accessed 26 June 2009

[38] PostgreSQL Global Development Group (2009) PostgreSQL: The world's most advanced open source database. http://www.postgresql.org/ Accessed 28 June 2009

[39] Potter CS, Randerson JT, Field CB, et al. (1993) Terrestrial Ecosystem Production: a Process Model Based on Global Satellite and Surface Data. Global Biogeochem. Cycles. 7:811-841

[40] Python Software Foundation (2009) Python Programming Language – Official Website. http://www.python.org/ Accessed 28 June 2009

[41] Refractions Research (2009) PostGIS : Home. http://postgis.refractions.net/ Accessed 28 June 2009

[42] Rey S (2009) Show me the code: spatial analysis and open source. Journal of Geographical Systems. 11(2):191-207

[43] Schlesinger W (1997) Biogeochemistry: An analysis of global change. San Diego: Academic Press

[44] Sellers PJ, Dickinson RE, Randall DA, et al. (1997) Modeling the Exchanges of Energy, Water, and Carbon Between Continents and the Atmosphere. Science. 275(5299):502-509

[45] Skamarock WC, Klemp JB, Dudhia J, et al. (2005) A Description of the Advanced Research WRF Version 2. Boulder, Colorado: National Center for Atmospheric Research Boulder, Colorado, Mesoscale and Microscale Meteorology Division

[46] Smith TM, Lakshmanan V (2005) Utilizing Google Earth as a GIS platform for weather applications. In: 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology. Atlanta, Georgia, USA: American Meteorological Society. http://ams.confex.com/ams/pdfpapers/104847.pdf

[47] Stull RB (1988) An introduction to boundary layer meteorology. Dordrecht, The Netherlands: Kluwer http://adsabs.harvard.edu/abs/1988aitb.book.....S Accessed 28 June 2009

[48] The Mathworks (2009) MATLAB - The Language Of Technical Computing. http://www.mathworks.com/products/matlab/ Accessed 29 June 2009

[49] Thomson DJ (1987) Criteria for the Selection of Stochastic Models of Particle Trajectories in Turbulent Flows. Journal of Fluid Mechanics Digital Archive. 180(-1):529-556

[50] Vogelezang D, Holtslag A (1996) Evaluation and model impacts of alternative boundary-layer height formulations. Boundary-Layer Meteorology. 81(3):245-269

Imprint

Acknowledgements
Daniel Holt, LaTeX magic & layout support
Various reviewers & writers

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the LaTeX 2_ε style source has been kindly provided by the GRASS and R News editorial boards.

Journal online: http://www.osgeo.org/journal

OSGeo Homepage: http://www.osgeo.org

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8

9 771994 189002 0 0

ISSN 1994-1897

OSGeo

Your Open Source Compass

osgeo.org