National Institute for Applied Statistics Research Australia

The University of Wollongong

Working Paper

05-14

Visualizing massive spatial datasets using multi-resolution global grids

T. Stough, A. Braverman, N. Cressie, E. Kang, A.M. Michalak, H. Nguyen, and K. Sahr

National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong NSW 2522. Phone +61 2 4221 5435, Fax +61 2 4221 4845. Email: anica@uow.edu.au

Visualizing massive spatial datasets using multi-resolution global grids

T. Stough[1], A. Braverman[1], N. Cressie[2], E. Kang[3], A.M. Michalak[4], H. Nguyen[1], and K. Sahr[5]

[1]NASA Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA

[2]National Institute for Applied Statistics Research Australia, University of Wollongong, Austrailia

[3]Department of Mathematical Sciences, University of Cincinnati, Cincinnati, OH

[4]Department of Global Ecology, Carnegie Institution for Science, Stanford, CA

[5]Department of Computer Science, Southern Oregon University, Ashland, OR

Abstract

In this chapter, visualization is used to evaluate the performance of global-scale computational algorithms. We generate synthetic global data sets and input them into computational algorithms that have a visualization capability. The global visualization allows us to quickly and easily compare the output of the computational algorithm to the synthetic-data input. Visualization is key here because the algorithms we are evaluating must respect the spatial structure of the input. We modify, augment, and integrate four existing component technologies: statistical conditional simulation, Discrete Global Grids, array set addressing, and Google Earth, where the internal representation of the synthetic data to be visualized is mirrored by the structure of the statistical model used to generate it. Both are spatially nested, so that one can move up and down in resolution in a mutually consistent way. We provide an example of how our simulation-visualization system may be used, by evaluating a computational algorithm called Spatial Statistical Data Fusion that was developed for use on massive, remote sensing data sets.

Keywords: Discrete Global Grids; Array Set Addressing; Efficient Algorithms and "Flattening"; Hexagonal Image Processing; Massive Spatial Datasets; Remote Sensing; Multi-Resolution; Google Earth; Spatial Statistics; Conditional Simulation of Atmospheric CO_2; Climate Model Downscaling

1 Introduction

As satellite technologies for Earth observation have advanced over the past decades, the volume and complexity of geophysical data collected by space-based instruments has grown, and so have the challenges of interrogating these data and drawing quantitative conclusions from them. Large-scale computational algorithms that transform data through many stages, from raw bits to meaningful information, are required in order to realize an order-of-magnitude increase in scientific return. However, those algorithms necessarily incorporate modeling assumptions and computational approximations that may lead to artifacts that in turn may compromise scientific conclusions. Thus, it is essential to understand and quantify geophysical artifacts, and visualization plays a key role in that endeavor. The geolocational aspects of remote sensing data make them natural to visualize and interactively explore through maps.

The usual mechanism for evaluating computational algorithms is a simulation experiment (SE): simulated data with known properties generate synthetic input to the algorithm of interest, and the algorithm's outputs are compared to the corresponding "true" values obtained from the simulated data. Implementing SEs for algorithms that are designed to run on massive satellite data sets can be challenging for at least two reasons. First, many geophysical processes of interest vary continuously in space, requiring very-high-resolution simulations to realistically mimic them. Moreover, realism also requires that scientific knowledge of the underlying geophysics be brought to bear by enforcing some form of consistency between the simulated data and the output of a coarse-resolution, geophysical process model. This means that the simulations must be consistent across scales, not only with respect to mean structure but also with respect to spatial covariance. Second, observed data collected by satellite remote sensing instruments represent incomplete aggregates over different spatial supports, with measurement errors superimposed. The SE must recreate this by averaging the synthetic field over an instrument's ground footprints, or sampling from the field if the footprints are smaller than the resolution of the simulated field. These operations must be performed in a way that recreates the spatial sampling and error characteristics of real

satellite instruments.

Both these problems require that the simulated field exhibits reasonable spatial coherence and variability. One way to achieve this is through a fine-resolution spatial statistical model that respects the output of a coarse-resolution physical (deterministic) model. By this, we mean that the parameters of the statistical model are set in such a way that, when the simulated field is aggregated up to the coarse resolution of the physical model, it is guaranteed to reproduce the output of the physical model. We call this constrained-parameter-fitting procedure *calibration*, and we use *conditional simulation* (e.g [4], Ch. 3) to simulate from the calibrated model. Here, the computational algorithm we use to illustrate our approach is Spatial Statistical Data Fusion (SSDF; [12]), which ingests two or more massive, heterogeneous, remote sensing data sets and produces optimal estimates of the underlying field. The visualization challenge is to display the massive, fine-resolution conditional simulation and the equally massive output of SSDF so that they can be compared. Both data sets are global and are expected to reproduce large and small-scale spatial structures. The visualization system must be able to render these features without geometric distortion, and it must be capable of zooming in and out so that features and possible artifacts can be explored at a variety of scales.

A number of systems and software tools for multi-resolution geographic visualization already exist. Google Earth does display and allow for pyramid-based multi-resolution zoom, but it uses a cylindrical projection that causes distortions in both appearance and, crucially for us, in representation of spatial relationships. The cylindrical projection creates a non-uniform tiling of earth's surface, with tiles becoming smaller near the poles. This distorts the spatial statistical properties of fields whose units are "per-unit areas." The HEALPix (Hierarchical Equal Area isoLatitude Pixelization) [9] system represents data at multiple resolutions, with storage and computation on the sphere. However, it does not use hexagonal tessellations of the sphere, which are ideal for spatial statistical inference [13], nor does it provide a visualization capability by itself. Laderstadter [11] has developed a system for exploring large climate data sets using interactive visualization and simple statistical

tools. Like Google Earth, this system uses a cylindrical projection and does not perform computations on the sphere. Other tools designed for global data sets (e.g., The Global Climate Change Viewer [1]; Climate Wikience [17]) typically display data at resolutions that are too coarse for our purposes and use unequal-area latitude-longitude grids. While they often possess simple computational tools, they do not typically include downscaling to the finer resolutions, where our interest lies.

Our simulation-visualization system combines four key technologies: 1) a multi-resolution statistical process model calibrated to the output from a coarse-resolution deterministic model; 2) the Discrete Global Grids (DGG) software package for tessellating the globe with a hierarchy of nested hexagonal grids to provide a system of multi-resolution supports for prediction; 3) an enhanced indexing system for cells of spherical hexagonal grids and for mapping the cells onto a flat plane so that the statistical process model can be used without geometric distortion; and 4) Google Earth for multi-resolution, interactive visualization of the simulated field and the computational algorithm being evaluated. In Section 2, we describe these four technologies and how we adapted and integrated them for our purposes. Section 3 is a case study showing how we used our system to visualize a) simulated fine-resolution fields produced by conditional simulation, b) synthetic instrument observations constructed from the simulated field, and c) the output from SSDF. Finally, in Section 4 we offer some conclusions about efficacy of our system and a discussion of future work.

2 Algorithms and Methods

We have combined four component technologies to create a simulation-visualization system for massive geophysical data sets. In this section, we describe these components and how we have adapted them for our purposes. In Section 2.1, we briefly introduce conditional simulation. In our context, it uses a dimension-reducing, multi-resolution spatial statistical model that enables optimal spatial prediction at a variety of spatial resolutions. Those predictions are identified with the hexagonal cells of the DGG, which have certain desirable

properties (e.g., equal area) and are described in Section 2.2. To exploit DGG's downscaling and image-processing features, two things are required: a method for flattening spherical grids onto two-dimensional planes, and an efficient indexing system for the grid cells. In Section 2.3, we describe how we satisfy these two requirements. Finally, Google Earth is a ubiquitous and intuitive interactive visualization environment for multi-scale georeferenced data sets. In Section 2.4, we describe how we leverage this platform for the exploration of spatial predictions at multiple scales.

2.1 Conditional Simulation

Atmospheric processes are defined at every location on the sphere, which is our mathematical abstraction of Earth's surface. In practice, the surface of the sphere is discretized into a fine-resolution regular grid; we call a generic grid cell a Basic Areal Unit or BAU. In what is to follow, we let the BAUs be the hexagons of the DGG at the finest resolution (see Section 2.2) and identify each BAU by the latitude and longitude of its center. Let $\boldsymbol{s}$ denote the two-dimensional latitude-longitude center of a BAU. Then our model for the geophysical variable of interest, Y, at $\boldsymbol{s}$ is

$$Y(\boldsymbol{s}) = \mu(\boldsymbol{s}) + \nu(\boldsymbol{s}) + \xi(\boldsymbol{s}), \tag{2.1}$$

where $\boldsymbol{s}$ ranges over the sphere, $\mu(\boldsymbol{s})$ is the large-scale trend, $\nu(\boldsymbol{s})$ is smooth small-scale variation, and $\xi(\boldsymbol{s})$ represents the remaining micro-scale variation. The components of (2.1) are assumed to be statistically independent.

Suppose that the total number of BAUs over Earth's surface is N; then we can form N-dimsenional vectors for each of the terms in Equation (2.1) by simply stacking the terms corresponding to the N locations into column vectors. Thus we can compactly write the entire model as,

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\nu} + \boldsymbol{\xi}. \tag{2.2}$$

Cressie and Johannesson [5, 6] developed a flexible, nonstationary spatial statistical model they called the Spatial Random Effects model (SRE; see also [21]), and we use that model

here for $\boldsymbol{\nu}$ and $\boldsymbol{\xi}$. We assume that $\boldsymbol{\mu}$ describes the mean of $\boldsymbol{Y}$ and that $\boldsymbol{\nu}$ and $\boldsymbol{\xi}$ are independent, zero-mean, multivariate Gaussian distributions.

To simulate the entire field $\boldsymbol{Y}$, we use y-values defined on a coarse-scale grid that represent our scientific understanding of the geophysical processes of interest. These might be output from a finite-element approximation to a physical model. For instance, in Section 3, we use the output of the Parameterized Chemistry and Transport Model (PCTM) for CO_2 concentrations at the resolution of $1^\circ \times 1.25^\circ$ as our coarse-scale y-values; these "inform" the simulation on BAUs defined by the finer-resolution DGG resolution 8 hexagons (30 kilometers in diameter).

Let the number of coarse-scale grid cells be M, and let $\tilde{\mathbf{Y}}$ be the associated M-dimensional vector of y-values. We consider the coarse-scale process to be and integrated version of the underlying geophysical processes; namely,

$$\tilde{\mathbf{Y}} = \mathbf{A}\mathbf{Y},$$

where $\mathbf{A}$ is the $M \times N$ incidence matrix that describes the relationship between the BAUs and the coarse-scale grid.

Models for $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, and $\boldsymbol{\xi}$ result in models for $\tilde{\boldsymbol{\mu}} \equiv \mathbf{A}\boldsymbol{\mu}$, $\tilde{\boldsymbol{\nu}} \equiv \mathbf{A}\boldsymbol{\nu}$, and $\tilde{\boldsymbol{\xi}} \equiv \mathbf{A}\boldsymbol{\xi}$. Consequently, we can "calibrate" choices for $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, and $\boldsymbol{\xi}$ based on the empirical mean and empirical covariance of $\mathbf{Y}$.

Naturally, we would like the simulated values at BAUs to be "consistent" with the physical-model output. At the very least, we require that, when we aggregate the simulated field from the BAU-scale up to the coarse scale of the physical model, the simulated field agrees with the model output. To achieve this, instead of simulating $\mathbf{Y}$ from its joint distribution obtained from (2.2), we simulate from the conditional distribution of $\mathbf{Y}$, conditional on the physical-model output. That is, we generate an N-dimensional vector $\mathbf{Y}$ from the conditional distribution $\mathbf{Y}$ given $\mathbf{A}\mathbf{Y} = \tilde{\mathbf{Y}}$. In obvious notation,

$$\begin{aligned}&\mathbf{Y}|\mathbf{A}\mathbf{Y} = \tilde{\mathbf{Y}} \\ &\sim \text{Gau}(\boldsymbol{\mu} + \boldsymbol{\Sigma}\mathbf{A}'\left(\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}'\right)^{-1}\left(\tilde{\mathbf{Y}} - \mathbf{A}\boldsymbol{\mu}\right), \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\mathbf{A}'\left(\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}'\right)^{-1}\mathbf{A}\boldsymbol{\Sigma}),\end{aligned} \tag{2.3}$$

where $\mathbf{\Sigma} \equiv \text{var}(\mathbf{Y})$. This allows us to simulate finer-resolution y-values consistent with the coarse-resolution output. Note that the conditional simulation defined by (2.3) requires computation of $(\mathbf{A\Sigma A}')^{-1}$, the inverse of an $M \times M$ matrix. We take advantage of the variance-covariance structure resulting from the Spatial Random Effects model and use the Sherman-Morrision-Woodbury formula (e.g., [5, 6]) to invert the matrix, $(\mathbf{A\Sigma A}')^{-1}$ with computational complexity of only $O(M)$.

2.2 Discrete Global Grids

Discrete Global Grids (DGGs; [20]) provide an approach to uniformly tiling the sphere with equal-area hexagonal cells at multiple resolutions. Regular polygonal cells are defined on the faces of a regular polyhedron, and these cells are then projected to the sphere using an appropriately designed inverse equal-area projection. Since a base polyhedron has the same topology as the sphere, the topological singularities associated with whole-Earth cylindrical projections are avoided.

The ISEA4H (Icosahedral Snyder Equal Area aperture 4 Hexagonal) DGG was chosen for this study. This DGG is constructed by tiling an icosahedron with cells that are primarily regular hexagons. Hexagonal grid cells have numerous advantages over traditional square grid cells. Hexagons are the most compact regular polygon that tile the plane, and hexagonal cells exhibit unambiguous uniform adjacency. Rasters of hexagonal pixels are 13.4% more efficient at sampling circularly band-limited signals [14]. For kriging, hexagons have lowest average standard error, lowest maximum standard error, and maximum screen effect [13]. The reference [19] provides a survey of additional advantages of hexagonal grids. It should be noted that it is impossible to tile a polyhedron completely with hexagons; in the case of the icosahedron, the 12 cells centered on the vertices of the icosahedron are pentagons with exactly 5/6 the area of the hexagonal cells.

In the ISEA4H DGG, multiple grid resolutions are constructed by introducing, at each resolution, cells that are 1/4 the size of the cells at the next coarsest resolution. The icosahedral version of the Snyder equal area polyhedral projection [22] is used to inversely project

the cells from the icosahedral faces to the sphere, preserving equal area at the cost of distorting the shapes of the hexagonal cells. The DGG software provides us with nested grids at increasingly fine levels of resolution, ranging from twelve 7674-km cells at the root of this hierarchy, to 40,962 120-km cells at resolution 6, 655,362 30-km cells at resolution 8, and more than 671 million one-kilometer cells at resolution 13.

2.3 Flattening and Image Processing

The DGG provides a multi-resolution global grid that covers a sphere with equal-area hexagons, modulo twelve pentagons. However, indexing these grid cells in a way that allows efficient storage and computation is not simple. The first step is to flatten the global grid onto a two-dimensional plane that can be easily manipulated and stored. A key goal in flattening is to achieve an arrangement of grid cells in computer memory that maintains locality of reference. We unfold the icosahedron of the DGG onto a plane; see Figure 2.1 as described in [3]. It is then necessary to choose an indexing scheme that allows storage and addressing in the flattened grid.

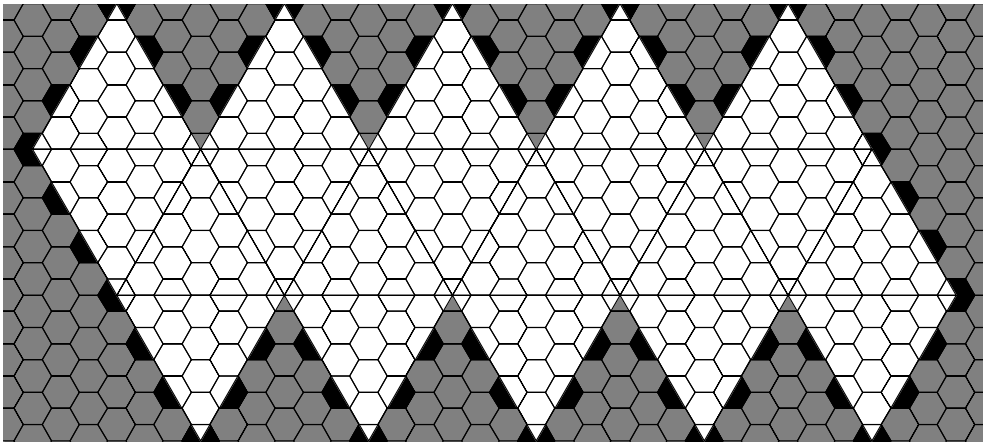

Figure 2.1: A flattened icosahedron

Array Set Addressing (ASA; [18]) provides a simple coordinate system with an efficient storage template for planar hexagonal grids. The ASA coordinate system has constant-time computation of nearest neighbors, distances, vectors, and routing on a hexagonal grid.

Convolution can be performed using optimized matrix operations on the arrays in memory, allowing fast downsampling, filtering, sampling, and other image-processing algorithms. ASA hexagonal grids are divided into two arrays, one for the even rows and one for the odd rows (see Figure 2.2). The ASA coordinate for any hexagonal cell is indexed by the triple (a, r, c), where $a \in [0, 1]$ specifies which of the two arrays and (r, c) specify the row and column, respectively.

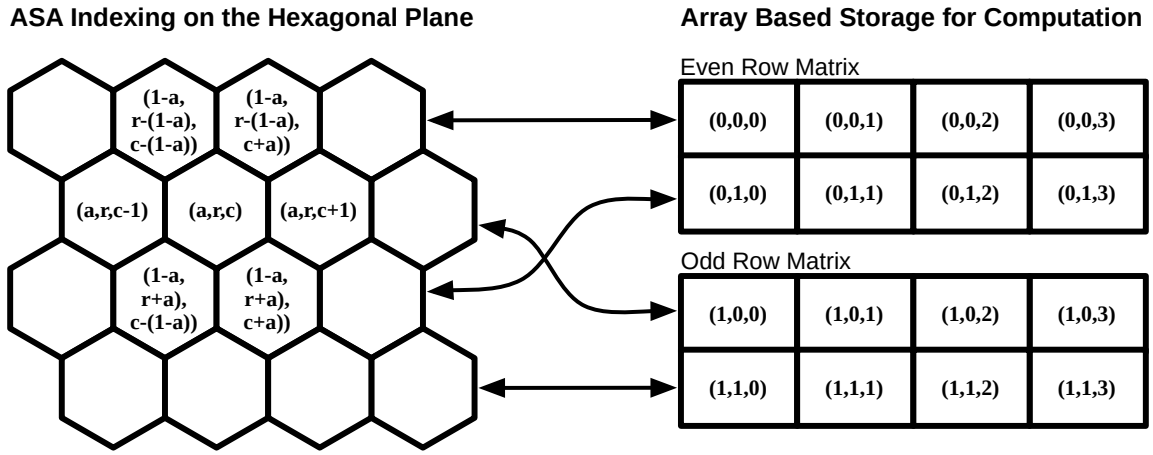

Figure 2.2: Hexagonal grid separated into two arrays and addressed using Array Set Addressing (ASA).

The flattened, unfolded DGG does not completely fill a plane with hexagonal cells. There are gores, or empty locations, in the planar representation of the globe, as well as padding at the edges of the planar image (see Figure 2.1). In order to compute efficiently on this plane, as if it were the sphere, we pad the gores and edges with the values that would be neighbors to those cells on the folded icosahedron. Unfortunately, this padding breaks down in the vicinity of the 12 pentagonal cells that are distributed around the sphere. In order to detect and deal with these, we pad the flattened plane with "NaN" in each of these 12 locations along the centerlines of the gores (see Figure 2.3). The result of this strategy assures that, when computing on the flattened plane, the NaN result will "poison" any computations that are close to the pentagons. This allows us to compute on the vast majority of the sphere via an efficient storage and indexing method, while easily detecting the 12 regions on the sphere

where special-case processing is required. In many cases, we can ignore these regions and compute them only when needed. Flattening, ASA indexing, and NaN poisoning provide

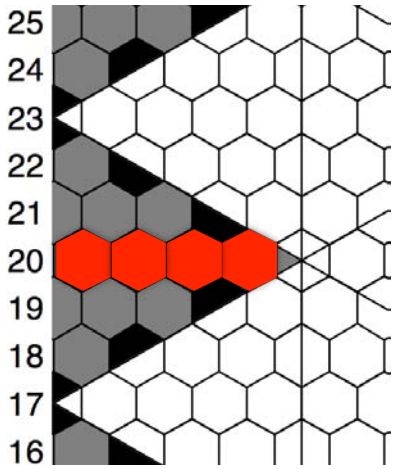

Figure 2.3: The red cells are "NaN poisoned" while the gray cells are filled with the values of the cells that they overlap with after folding.

a way to move data computed on a DGG into arrays in memory that can be operated on efficiently using standard image-processing techniques, with only small modifications.

2.4 Google Earth

Visualizing global data that have been computed on the sphere requires a globe upon which the mapping takes place. Although there are "digital globe" displays, Google Earth offers a virtual-globe platform that is ubiquitous, accessible, and free. It also supports the visualization of global data as the user spins the globe and zooms in and out.

Google Earth is designed to deal with cylindrically projected digital imagery data. For very fine-resolution imagery, it is possible to use a multi-resolution pyramided structure to speed visualization. Pyramiding allows the use of coarse-resolution images when zoomed out and full-resolution tiles when zoomed in. The tiles at intermediate resolutions are loaded as needed when the user zooms or moves around the globe. Unfortunately, using the image-pyramid approach requires working in a cylindrical projection. This would complicate our

display of globally gridded data, introducing the very distortion that we hope to avoid. Ideally, we would like to use pyramided arrays of hexagons, but Google Earth does not support that data type.

To circumvent this limitation, we use KML (Keyhole Markup Language), the file format used to create Google Earth visualizations, to represent the hexagonal cells of a set of DGGs directly, as a list of coordinates that define the boundaries of the hexagonal cells. We then shade those cells' interiors using a color pallette to display the magnitudes of data associated with those cells. Representing each grid cell as a polygon in KML has the advantage of accurately displaying grid-cell boundaries at any scale, but it does not allow for the use of built-in multi-resolution pyramids for quick display and memory management. We deal with this by rendering small regions at high resolution (finer polygons), and global data at lower resolution (coarser polygons). The multi-resolution nature of DGG allows us to easily group finer polygons and average them to create coarser polygons. We are also investigating how to render image pyramids for browsing, and then how to transition to polygons when zooming in.

2.5 Integrating Component Technologies

Our end-to-end simulation-visualization system is implemented with a python toolkit called DDGrid.py. This toolkit wraps the DGGRID software [19] and implements the data structures and algorithms required to store and generate KML; KML is used to visualize simulated fields, to extract synthetic instrument observations, and later to visualize the output of the computational algorithm being evaluated. DDGrid.py leverages existing optimized image-processing tools from Numpy and SciPy [15] for building multi-resolution pyramids. It is also used for computing simulated observations by averaging the BAU-level hexagons that coincide with the ground footprints of remote sensing instruments. Figure 2.4 is a data-flow diagram showing the main components of this system.

The DGGrid.py toolkit follows the principle of object-oriented design. The DGG is instantiated in objects that represent the entire grid as well as individual hexagons. This

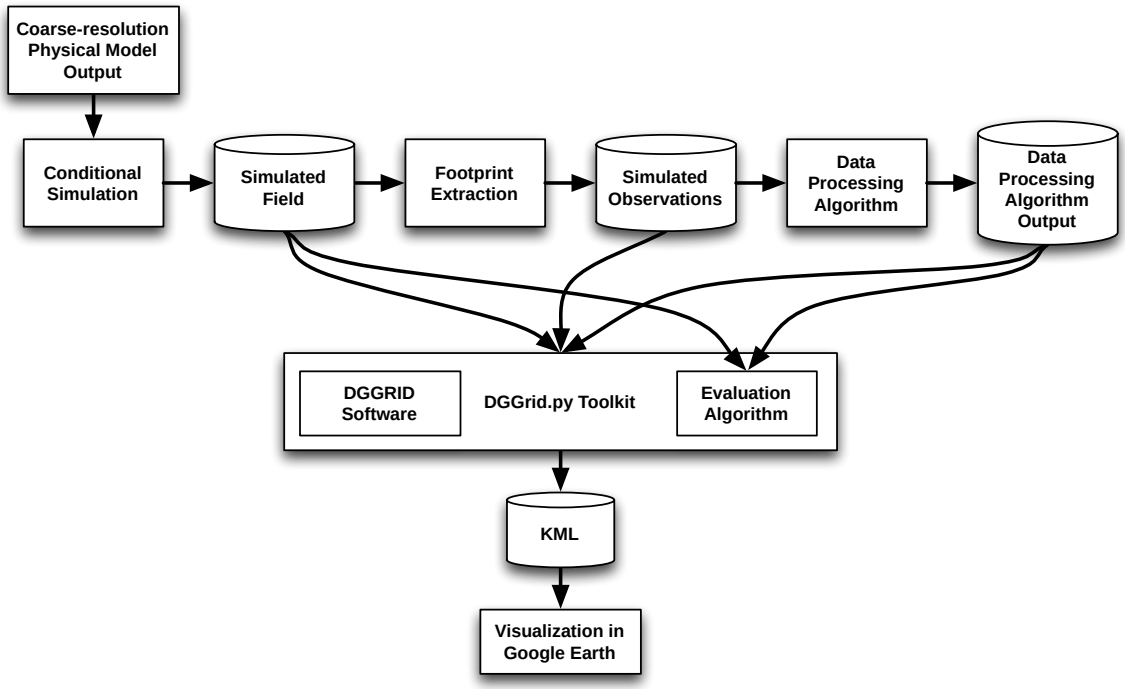

Figure 2.4: System diagram. The Evaluation Algorithm will calculate and display a fidelity metric for each hexagonal cell at the resolution of the visualization.

object-oriented structure allows us to support many features, like plug-in models for instrument footprints, different data types, and different visualization styles and evaluation functions. The toolkit integrates the global grid structure produced by DGGRID with the flattened, padded, NaN poisoned, and ASA addressable representations. The objects in DGGrid.py map to the topology and cells of the DGG, and each object is capable of producing KML to visualize itself. This allows us to subset the grid into any grouping that we like. We can also use the boundaries of any grid cell to create finer-resolution cells that make up the original grid cell. Together these features allow the production of easy-to-visualize, multi-resolution grids.

In addition to the object-oriented representation of the DGG, the toolkit implements utilities for extracting instrument footprints for simulation experiments. Our application requires aggregating hexagonal cells (BAUs) over regions commensurate with the ground footprint of a remote sensing instrument (see the discussion of the OCO-2 and AIRS footprints in Section 3). For our prototype system, we have implemented two types of footprint extraction: nearest DGG cell and average of cells within a given radius. For each footprint location and radius, we use ASA to compute neighborhoods of DGG cells associated with footprints and to extract corresponding averages. In the case of footprints smaller than the DGG cell, we extract the value of the nearest-neighbor DGG cell. The resulting synthetic instrument observations are stored and made available for algorithm testing. Footprint plug-ins will allow us to specify satellite-footprint shapes, response curves, and measurement-error behavior to simulate how the instruments measure.

The DGGrid.py toolkit is designed to support automated execution of simulation experiments. A single entry point allows sets of parameters to be defined and systematically processed. Hence, testing and visualization can be carried out for different parameters, both for the conditional simulation and for the data-processing algorithm being considered.

In the next section, we describe how we can use our approach to assess the performance of the Spatial Statistical Data Fusion (SSDF) algorithm. We shall eventually incorporate the ability to compute and display quantitative performance metrics from inside DDGrid.py,

but here we focus on what can be learned by visually comparing the synthetic input and the SSDF algorithm output.

3 Evaluating SSDF Global Estimates of CO2

This section describes the specific implementation of our simulation-visualization system for evaluating the SSDF algorithm. SSDF produces optimal estimates of geophysical fields from two or more massive, heterogeneous, remote sensing data sets. The methodology is similar to kriging and allows for input observations with different sampling characteristics and spatial supports. SSDF models and subsequently leverages spatial correlation in the data to produce optimal (minimum mean squared prediction error, unbiased) estimates of the underlying true fields; importantly, it also produces uncertainty measures (root mean squared prediction errors) of these estimates.

Here, we study the performance of the SSDF algorithm as it will be applied to data from two NASA instruments that measure carbon dioxide in the atmosphere: the Atmospheric Infrared Sounder (AIRS) and the Orbiting Carbon Observatory (OCO-2). The AIRS instrument has been in orbit since mid-2002, and it observes mid-tropospheric CO_2 concentration on circular footprints that are 90 kilometers in diameter and are contiguous [2]. The OCO-2 instrument will be launched in July 2014, and it will observe total column CO_2 concentration on contiguous trapezoidal footprints roughly 2 kilometers in diameter [7]. Both instruments fly on satellites that are in polar orbit, observing swaths of Earth along their respective tracks from pole to pole. The AIRS field of view across-track is about 1500 kilometers, so its swaths are wide and the entire world is seen once every three days. The OCO-2 field of view across-track is only about 10 kilometers, so its swaths are very narrow; the OCO-2 instrument never observes the whole world but repeats the same 233 globally distributed orbital paths every 16 days. Both instruments' data are subject to high degrees of "missingness" because neither can observe CO_2 in the presence of clouds.

To evaluate the performance of SSDF, we perform a simulation experiment using

DDGrid.py. First, we generate a synthetic CO_2 field at fine spatial resolution using conditional-simulation technology (Section 2.1). The simulation was performed at DGG resolution-8 in which the BAU hexagons are 30 kilometers in diameter.

The conditional simulation is calibrated to a coarser simulated atmospheric CO_2 field. The CO_2 concentrations were simulated using the output of PCTM [10] driven by analyzed meteorological fields from NASA's Goddard Earth Observation System, version 4 (GEOS-4). The prescribed net surface fluxes of CO_2 were taken from the Carnegie Ames Stanford Approach (CASA; [16]) model for biospheric fluxes, from Takahashi et al. [23] for the monthly mean climatology for air-sea CO_2 exchange, from Erickson et al. [8] for anthropogenic CO_2 emissions, and from the Global Fire Emission Database version 2 (GFED2; [24]) for wildfire and biomass burning emissions. This model is herein referred to as PCTM for simplicity. The model has a horizontal resolution of $1^\circ \times 1.25^\circ$ with 25 vertical levels. We use the simulated fields from level 8 (approximately 5-km elevation, meant to represent the mid-troposphere) at 1800 GMT on April 15, 2006, in the analysis presented here.

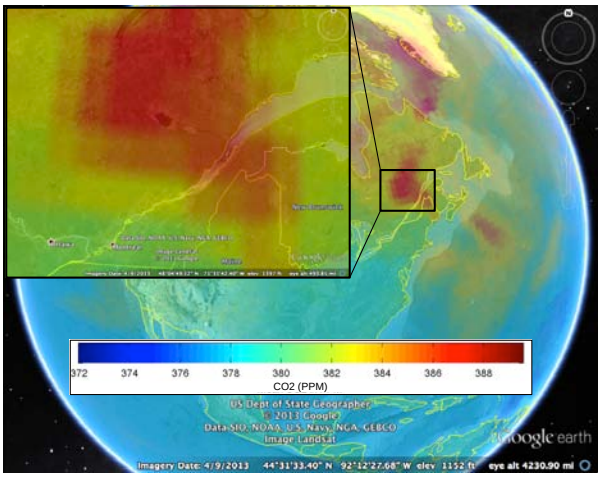

Figure 3.1a: PCTM CO_2.

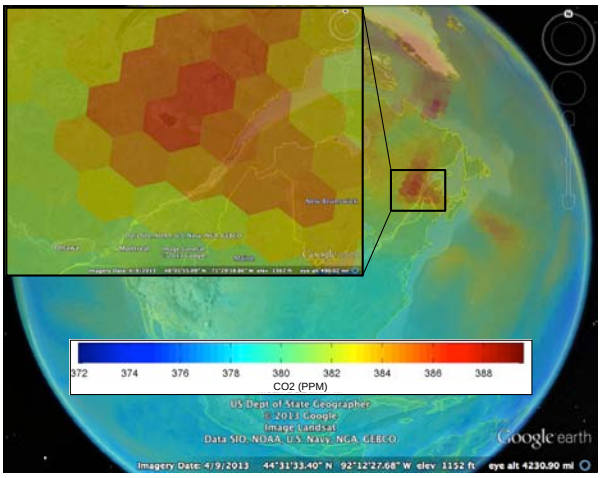

Figure 3.1b: Simulated CO_2, resolution-6.

Figure 3.1a shows the coarse-resolution PCTM model output for the mid-troposphere. The PCTM resolution is approximately DGG-resolution-6 near the equator. This coarse resolution shows up as blockiness in Figure 3.1a. Figure 3.1b is a global visualization of our simulation output. Although produced at DGG resolution-8, we have displayed the

simulation output at the coarser DGG resolution-6 (cells are 120 kilometers in diameter) because Google Earth's memory limitations prevent global display at finer resolutions.

Here we leveraged an important feature of our system. If we conditionally simulated at resolution-8 and aggregated to resolution-7 or to resolution-6, etc., we would obtain a process with the same statistical properties as if we conditionally simulated directly at the respective resolution. The visualizations in Figures 3.1a and 3.1b show nearly identical features, as they should, given the simulation's constraint that the simulated field at all resolutions must aggregate to reproduce the values on the PCTM grid.

In the second step, we sample the simulated field to create *synthetic observations* analogous to what AIRS and OCO-2 "see." We start with the centers of actual AIRS and OCO-2 footprints. For AIRS, we use the actual locations of non-missing footprints for a representative three-day period. To create synthetic AIRS observations, we average simulated values for all 30-kilometer hexagons (DGG-resolution-8) with centers falling within a 45-kilometer radius of the actual center of the AIRS footprint. In the case of OCO-2, whose footprint is smaller than the resolution-8 hexagon, we take the value of the simulated data for the hexagon with center nearest the center of the OCO-2 footprint. We use three representative days of simulated orbit tracks provided to us by the OCO-2 team at the Jet Propulsion Laboratory.

Figure 3.2a shows the simulated field at DGG resolution-8 for a wedge of Earth, with an inset that zooms in on eastern New England and Quebec, in order that the 30-kilometer hexagons are clearly visible. Synthetic observations for AIRS and OCO-2 are shown in Figure 3.2b. The main image shows the locations and values of AIRS observations for a subset of Earth's surface, color-coded according to their simulated values. Memory limitations of Google Earth prevent us from giving a full global display of AIRS footprints. The inset shows a better view of eastern New England and Quebec – the circles show the locations and sizes of AIRS observations. The thin, almost vertical, strip represents the OCO-2 orbit track, although there is a representation issue here because the strip is made up of 2-kilometer-diameter regions with values taken from the nearest 30-kilometer hexagon, whereas

the OCO-2 track is only about 10 kilometers wide. The size mismatch between AIRS and OCO-2 footprints would render the OCO-2 footprints invisible if we did not use the zoom in Figure 3.3f. The OCO-2 footprints are also color-coded according to their simulated values.

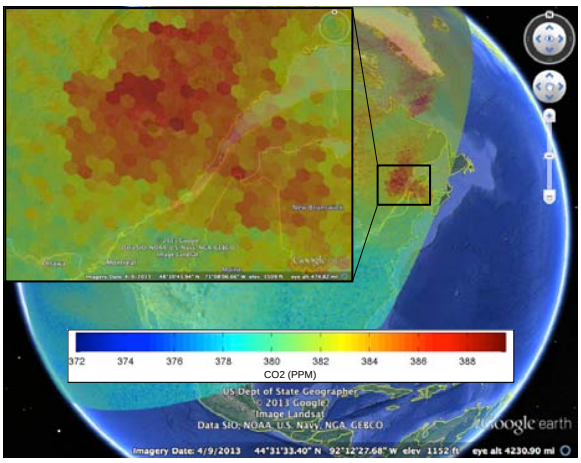

Figure 3.2a: Simulated CO_2, resolution-8.

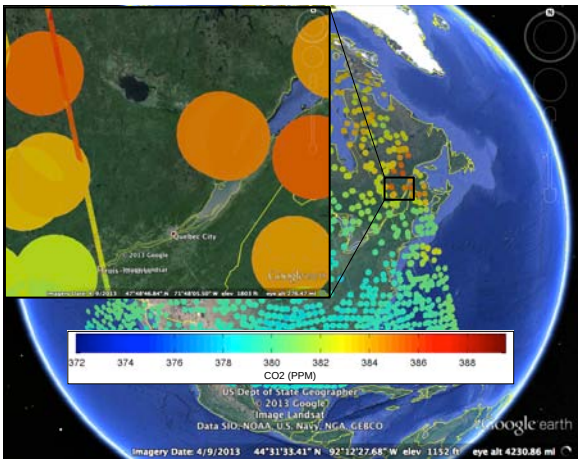

Figure 3.2b: Synthetic observations.

Finally, we apply SSDF to estimate a contiguous field of CO_2 concentrations given both synthetic AIRS and synthetic OCO-2 observations. Our estimates are produced at 30-kilometer spatial resolution (DGG resolution-8). Figures 3.3d and 3.3f show the fused estimates and corresponding standard errors at resolution-8 for the same wedge of Earth as in Figure 3.2a, and with high-resolution insets. Figures 3.3c and 3.3e show the corresponding global views produced by aggregating the resolution-8 SSDF results up to resolution-6. Figures 3.3a and 3.3b are duplicates of Figures 3.2a and 3.1b for easy comparison.

Exploratory evaluation of SSDF might include visually comparing Figure 3.3d to Figure 3.3a, and Figure 3.3c to Figure 3.3b. The former is a regional comparison at finer resolution, and the latter is a global comparison at coarser resolution. Both comparisons should be considered in light of the standard-error maps that correspond to the spatial statistically fused estimates. These are shown in Figures 3.3f and 3.3e, respectively.

One can make a number of observations about SSDF based on these visualizations. At the global scale, the SSDF estimates of CO_2 in Figure 3.3c give a smoother impression than the simulated CO_2 process given in Figure 3.3b. The standard errors in Figure 3.3e show

features that do not appear to correspond to features in the estimates themselves, but they do have some similarities to the simulated output in Figure 3.3b. Recall that the input to SSDF is made up of sparse synthetic footprints like those shown in Figure 3.2b. This accounts for the smoothing in the fused estimates and will also influence the geographic patterns of the standard errors. At the finer spatial scale (Figure 3.3d), the smoothing is even more pronounced, and it is accompanied by similar smoothing in the standard-error map (Figure 3.3f). This is in sharp contrast to the spatial heterogeneity of the resolution-8 simulated field in Figure 3.3a and is due to the sparsity of the synthetic observations in the region of the inset. Our simulation-visualization experiments illustrate that SSDF estimates are likely to be more useful on global scales than on regional ones if the instrument data are geographically sparse. This is not surprising, and it could have been anticipated with some knowledge of how SSDF works (i.e., it is akin to kriging), but this visualization tool makes it possible to understand how problematic this is for specific regions of interest.

4 Conclusion

We have built an initial version of a simulation, analysis, and visualization infrastructure that ties the visualization environment to the representation of the underlying data in nested, discrete global grids. In our implementation, the underlying fine-resolution data are produced using a spatial statistical conditional-simulation methodology. The methodology constrains the simulation output to reproduce features of a physical model that was constructed from scientific knowledge about the structure of the true physical process.

We developed a python toolkit to implement instrument-like sampling of the simulated field, manage interfaces between component technologies, and augment them where necessary. We demonstrated how our system can be used to visualize and better understand the behavior of a global data processing algorithm, SSDF, over different spatial scales. This is possible because the simulated field obeys hierarchical aggregation consistency, so that coarse-resolution fields can be derived in a statistically controlled way from fine-resolution

fields. This mirrors the upscale-pyramiding capability with which we have augmented Google Earth. We are currently working on a downscale-pyramiding capability that would enable us to generate fine-resolution simulated fields for limited regions and display them in near-real time as Google Earth zooms in on the area. This infrastructure was implemented for the SSDF algorithm, but other computational algorithms whose performance depends on fine-resolution spatial structure can also be evaluated.

Acknowledgements

The authors would like to thank Abhishek Chatterjee for processing and providing inputs on the use of PCTM/GEOS-4 global model data. The authors would also like to thank Dr. Jon Bradley and Dr. Jonathan Hobbs for their contributions and comments. The work described in this book chapter was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. It is supported by NASA's Earth Science Technology Office through its Advanced Information Systems Technology program.

References

[1] Alder, J., Hostetler, S., and Williams, D. (2013). An interactive web application for visualizing climate data. *Eos, Transactions American Geophysical Union*, Volume 94, Issue 22, pages 2324–9250.

[2] Aumann, H. H., Chahine, M. T., Gautier, C., Goldberg, M. D., Kalnay, E., McMillin, L. M., Revercomb, H., Rosenkranz, P. W., Smith, W. L., Staelin, D. H., Strow, L. L., and Susskind, J. (2003). AIRS/AMSU/HSB on the Aqua mission: Design, science objectives, data products, and processing systems. *IEEE Transactions on Geoscience and Remote Sensing*, Volume 41, Number 2, pages 253–264.

[3] Carr, D., Kahn, R., Sahr, K., and Olsen, T. (1997). ISEA discrete global grids. *Statistical Computing & Statistical Graphics Newsletter*, Volume 8, Number 2/3, pages 31–39.

[4] Cressie, Noel A.C. (1993). *Statistics For Spatial Data*, rev. edn., John Wiley and Sons, New York.

[5] Cressie, N. and Johannesson, G. (2006). Spatial prediction for massive datasets. *Mastering the Data Explosion in the Earth and Environmental Sciences: Proceedings of the Australian Academy of Science Elizabeth and Frederick White Conference. Australian Academy of Science*, Pages 1–11.

[6] Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of The Royal Statistical Society, Series B*, Volume 70, Number 1, Pages 209–226.

[7] Eldering, A., Solish, B., Kahn, P., Boland, S., Crisp, D., and Gunson, M. (2012). High precision atmospheric CO_2 measurements from space: The design and implementation of OCO-2. *Proceedings of the 2012 IEEE Aerospace Conference*, Big Sky, Montana, USA, March 3-10.

[8] Erickson, D.J., Mills, R.T., Gregg, J., Blasing, T.J., Hoffman, F.M., Andres, R.J., Devries, M., Zhu, Z., Kawa, S.R. (2008). An estimate of monthly global emissions of anthropogenic CO_2: Impact on the seasonal cycle of atmospheric CO_2. *Journal Of Geophysical Research – Biogeosciences*, Volume 113, Issue G1, Article G01023.

[9] Górski, K. M., Hivon, E., Banday, A. J., Wandelt, B. D., Hansen, F. K., Reinecke, M., Bartelmann, M. (2007). HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, Volume 622, Issue 2, pages 759–771.

[10] Kawa, S. R., Erickson III, D. J., Pawson, S., and Zhu, Z. (2004). Global CO_2 transport simulations using meteorological data from the NASA data assimilation system. *Journal of Geophysical Research*, Volume 109, Article D18312.

[11] Ladstdter, F., and 4 coauthors (2010). Exploration of climate data using interactive visualization. *Journal of Atmospheric and Oceanic Technology*, Volume 27, pages 667–679.

[12] Nguyen, H., Cressie, N., and Braverman, A. (2012). Spatial statistical data fusion for remote-sensing applications. *Journal of the American Statistical Association*, Volume 107, pages 1004–1018.

[13] Olea, R.A., (1984). Sampling design optimization for spatial functions. *Mathematical Geology*, Volume 16, Number 4, pages 369–392.

[14] Petersen, D. P. and Middleton, D. (1962). Sampling and reconstruction of wave-number-limited functions in n-dimensional euclidean spaces. *Information Control*, Volume 5, pages 279–323.

[15] Peterson, P. (2007). Python for scientific computing. *Computing in Science & Engineering*, Volume 9, Number 90, pages 10–20.

[16] Randerson, J.T., Thomps, M.V., Conw, T.J, Fun, I.Y., and Field, C.B. (1997). The contribution of terrestrial sources and sinks to trends in the seasonal cycle of atmospheric carbon dioxide. *Global Biogeochem. Cycles*, Volume 11, Number 4, pages 535–560, doi:10.1029/97GB02268.

[17] Rodriges Zalipynis R.A., Zapletin E.A., and Averin G.V. (2011). The Wikience: Community data science. Concept and implementation. *Proceedings of the 7th International Scientific-Technical Conference "Informatics and Computer Technologies (ICT-2011)"*, November 22–23, 2011, Donetsk, Ukraine, Volume 1, pages 113–117.

[18] Rummelt, N.I. and Wilson, J.N. (2011). Array set addressing: Enabling technology for the efficient processing of hexagonally sampled imagery. *Journal of Electronic Imaging*, Volume 20, Issue 2, Article 023012.

[19] Sahr, K. (2011). Hexagonal discrete global grid systems for geospatial computing. *Archives of Photogrammetry, Cartography and Remote Sensing*, Volume 22, pages 363–376.

[20] Sahr, K., White, D., and Kimerling, A.J. (2003). Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, Volume 30, Issue 2, pages 121–134.

[21] Shi, T. and Cressie, N. (2007). Global statistical analysis of MISR aerosol data: A massive data product from NASA's Terra satellite. *Environmetrics*, Volume 18, pages 665–680.

[22] Snyder, J. P. (1992). An equal-area map projection for polyhedral globes. *Cartographica*, Volume 29, Number 1, pages 10–21.

[23] Takahashi, T., Sutherland, S.C., Sweeney, C., Poisson, A., Metzl, N., Tilbrook, B., Bates, N., Wanninkhof, R., Feely, R.A., Sabine, C., et. al. (2002). Global sea-air CO_2 flux based on climatological surface ocean pCO_2, and seasonal biological and temperature effects. *Deep-Sea Research Part II – Topical Studies In Oceanography*, Volume 49, Issue 9-10, pages 1601–1622.

[24] van der Werf, G.R., Randerson, J.T., Giglio, L., Collatz, G.J., Kasibhatla, P.S., Arellano, A.F. (2006). Interannual variability in global biomass burning emissions from 1997 to 2004. *Atmospheric Chemistry and Physics*, Volume 6, pages 3423–3441.

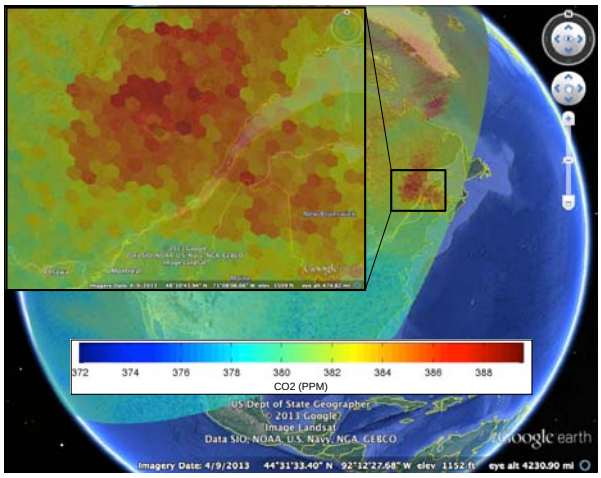

Figure 3.3a: Simulated CO_2, resolution-8.

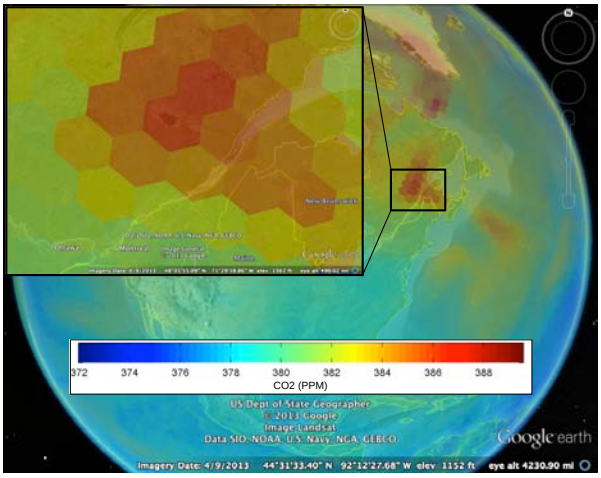

Figure 3.3b: Simulated CO_2, resolution-6.

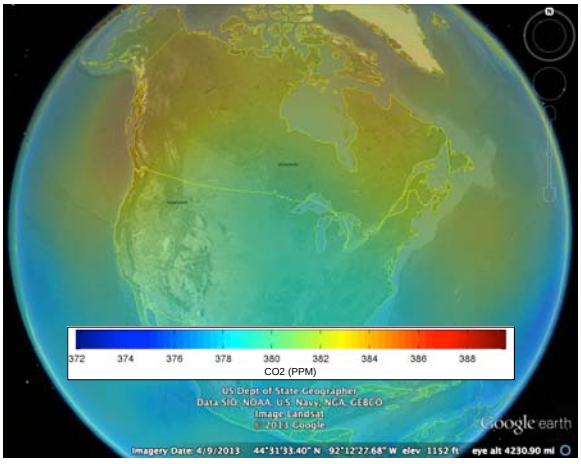

Figure 3.3c: SSDF estimates, resolution-6.

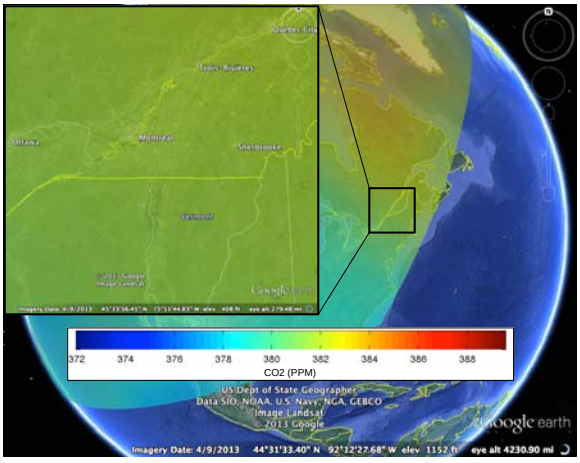

Figure 3.3d: SSDF estimates, resolution-8.

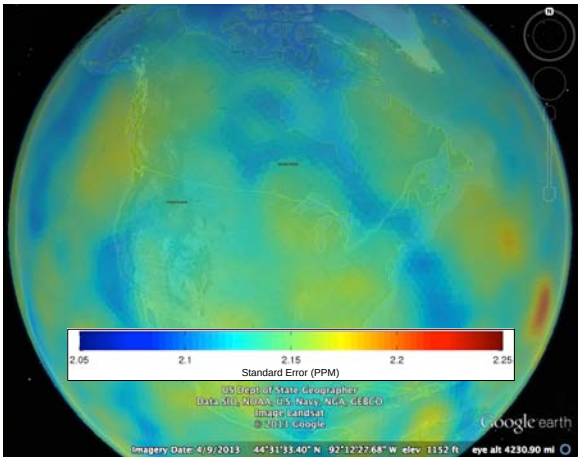

Figure 3.3e: Standard errors, resolution-6.

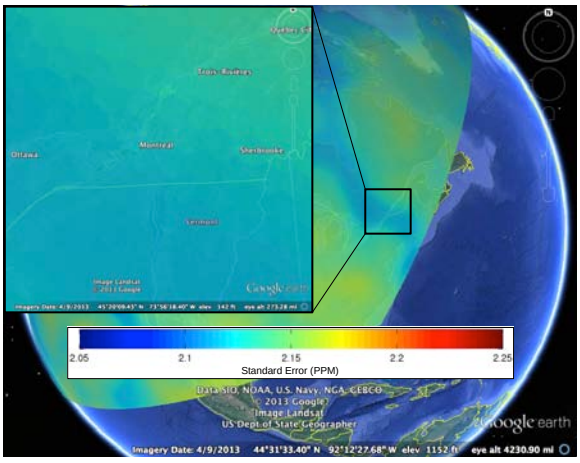

Figure 3.3f: Standard errors, resolution-8.