Integration of Google Maps/Earth with microscale meteorology models and data visualization

Yansen Wang *, Giap Huynh, Chatt Williamson

US Army Research Laboratory, Adelphi, MD 20783, USA

ARTICLE INFO

Article history:
Received 3 May 2013
Received in revised form 15 July 2013
Accepted 17 July 2013
Available online 27 July 2013

Keywords:
Model result visualization
Model Google GIS integration
Microscale meteorology

ABSTRACT

The Google Maps/Earth GIS has been integrated with a microscale meteorological model to improve the system's functionality and ease of use. Almost all the components of the model system, including the terrain data processing, morphological data generation, meteorological data gathering and initialization, and displaying/visualizing the model results, have been improved by using this approach. Different from the traditional stand-alone model system, this novel system takes advantages of enormous resources in map and image data retrieving/handling, four-dimensional (space and time) data visualization, overlaying, and many other advanced GIS features that the Google Maps/Earth platform has to offer. We have developed modular components for all of the model system controls and data processing programs which are glued together with the JavaScript language and KML/XML data. We have also developed small modular software using the Google application program interface to convert the model results and intermediate data for visualizations and animations. Capabilities such as high-resolution image, street view, and 3D buildings in the Google Earth/Map are also used to quickly generate small-scale vegetation and building morphology data that are required for the microscale meteorological models. This system has also been applied to visualize the data from other instruments such as Doppler wind lidars. Because of the tight integration of the internet based GIS and a microscale meteorology model, the model system is more versatile, intuitive, and user-friendly than a stand-alone system we had developed before. This kind of system will enhance the user experience and also help researchers to explore new phenomena in fine-scale meteorology.

1. Introduction

It is important for a meteorological model system to be easy to learn and to use from an application point of view. The system should hide the complexity of computer software and expose only the necessary interfaces in an intuitive graphical form between the users and the software. The system should also be rich in functionalities so that a user should be able finish the modeling tasks using the components in the system without using the separated software packages.

For our application in microscale meteorological models, it is a significant advantage for the system to have detailed geo-referenced 3D terrain, buildings, vegetation, soil etc. in graphical or image forms for the users, since microscale meteorological phenomena are forced mechanically and thermally by those fine scale ground surface features in addition to the influence of the larger scale atmospheric conditions above the atmospheric boundary layer (ABL). The easiness to fetch, process, and display those required fine morphological data in timely fashion is also a very important requirement. Traditionally, the meteorological modeling community has been using separate packages, including model input data pre-processing, main model, and post-processing software. These separated steps in a model application introduce some unnecessary complexity and can often cause confusions for the users. The graphics display packages such as GrADS and ViS5D (Doty 1995; Hibbard and Santek, 1989) were used to visualize the results, or results from intermediate steps. These graphics packages were specifically designed for displaying large scale model results in which fine-scale surface morphology and other characteristics are not resolved. For example, it would be very difficult to use these traditional meteorological graphics packages to display high resolution (1–10 m) ground surface images or 3D terrain for fine-scale models. The recent development of the GIS system over the internet has produced a great potential for the advancement of microscale meteorological model systems. The Google Maps/Earth GIS has the advantages of cross-platform, internet based, and rich in high resolution data for the entire globe. The frequently updated high resolution images that are in the Google Maps/Earth is another benefit for the
meteorology modeling since many interesting meteorological phenomena are driven by the variations on the land surface. The Google system also has a well designed application program interface (Google Map/Earth programming API) which can be easily applied by meteorology model system developers. Another advantage of the internet based system is that it is based on standards, such as data exchange XML and KML formats, and the JavaScript language. Those standards make the implementation of the system to be much easier and enable it to be more extensible in the future.

The Google Earth has been used in the visualization tools for seismic tomographic data (Yamagishi et al., 2010) and meteorological satellite data (Chen et al., 2009). Those web based GIS systems have been proven to be very versatile tools and have changed the way in which scientists and the general public interact with three-dimensional geospatial data in a virtual environment. We have developed a prototype system for microscale meteorological models that connects the Google GIS to a meteorological model and users. This integrated system is initially developed for ARL's 3DWF (three-dimensional wind field) model (Wang et al., 2005; 2010; Hanna et al., 2011), and can be easily extended to our ABLE (Atmospheric Boundary Layer Environment) model (Wang et al., 2012). This system is user-centric, rich in functionality, efficient for a model user to do the morphological and meteorological data preprocessing, run the model, and display the model results. Different from the traditional stand-alone meteorological modeling system, this is a hybrid system which uses computing and data resources from the local client and remote Google server over the internet. Because of the extensive application the Google Maps/Earth for its GIS functions and data resources, the visualization components of the system are relatively light-weight software and are extendable for different customers or users. Some of examples are given in the paper to demonstrate the system capabilities. This paper first describes the architecture of the system and the data flow in the components. The geospatial data transformation and processing on which the entire system is based are illustrated through the terrain processing. The morphological data creation tool, the visualization tools for the Doppler wind lidar data, and model results are finally presented.

2. The integrated system architecture

Fig. 1 is a display of the microscale meteorological model architecture, where the sharp-cornered rectangles represent the data sets; the round-cornered rectangles represent the computer programs that process those data sets; the black arrow lines denote the data flows in the system; and the red double-headed arrows depict the graphical user interface (GUI) controls and interaction between the microscale meteorological model components and a user. The model system consists of several modular programs which are presented in bold font: the terrain processing, morphological data processing, model initialization, microscale meteorological model, model result post-processing, and model result display. The graphical user interface is a very light-weight subsystem that only collects the user simple inputs/or commands to control data flow into and out of these modules, and graphical displays from each of the model components. The data flow logic and the data processing in the modules are programmed in the modules' data interface. An example of the 3DWF model system is shown with top level menus in Fig. 2. Each button controls a model component, its data flow, and graphical display behavior. The natural order for running the model is from left button to right in that order when a user uses the 3DWF model system. The architecture of this system is modularized according to specific applications. A top-level control button can be added or deleted according to the requirement of a specific application. Sub-menu controls can be also added under the top control menus according to their function categories.

The connection between the model components and the GUI is achieved using the Microsoft Active X mechanism, i.e. each of the modules in the meteorological model system is an Active X component. The system modules are glued together using a popular and secure web language, the JavaScript. The modules in the 3DWF model system are coded using standard Fortran 90/95 and executed in a local client computer. The individual module can be coded by other computer languages depending on the developers' preference. Currently this GUI system is configured to run on a local computer with internet connection with Google Maps/Earth available in the web browser. Extension to web-service architecture is a simple task if a remote running capability is desired. Submenus can be added in each top menu to control different components of the system. For example, submenus in the terrain processing menu are used to set-up the model domain over the globe, retrieve the high-resolution terrain data and to process the terrain data into the model grid points. Different from traditional terrain preprocessing, the common GIS functions of Google Maps/Earth, such as map, image overlay, shaded relief terrain, street view, or the hybrid of those functions are preserved. A user can directly use those GIS functions that the Google Maps offers. This system is also a distributed system; that is, global high-resolution images and terrain data associated with Google Maps/Earth are served by the remote server of Google Inc. Those GIS
functions by Google Maps/Earth offer greater advantage to this system in comparison to the stand alone GUI.

3. Geospatial data transformation

Google Maps/Earth uses WGS84 (World Geodetic System 1984) coordinate. Any data to be visualized and processed using the Google GIS system has to be transferred between the local coordinate and the WGS84 coordinate system. Therefore, this transformation is the cornerstone for the entire system. The terrain data processing, the morphological data creation tool, and the visualization of the observational data and microscale meteorology model results are all based on this transformation.

In this section, we use the terrain data processing as an example to describe the geodetic data transformation method. Since Google Maps/Earth server only gives the terrain data in graphical display, we have to get terrain digital data from other sources. The fine-resolution (3° or finer) terrain data bases such as SRTM (Shuttle Radar Topographical Mission) or USGS’s DEM (Digital Elevation Model) data are necessary. Moreover, since 3DWF is used to model the fine-scale (meters up to 100 m) atmospheric flow, it needs fine resolution terrain data. In this project, we use the terrain elevation data set from SRTM (Farr et al. 2007) with 3-arcsecond (~90 m resolution at the equator) resolution. The data covers the land area, nearly global from 56S to 60N latitudes. We use the processed version 4 SRTM data set as described in Gamache (2005) in which some of the missing data holes were filled. The original data is organized in WGS84 (World Geodetic System 84) geodetic coordinate system. When the data are applied to the 3DWF model, they are transformed to the local East, North and Up (ENU) coordinate (see Fig. 3). Since the 3DWF is a fine scale wind model and its entire model domain is not intended to be larger than 20 x 20 km, this Cartesian coordinate system is a good choice with very little distortion due to the curvature of the Earth’s surface. The transformation from the WGS84 data to the ENU coordinate is performed as follows (Fukushima, 2006; Featherstone and Caessens, 2008).

Transformation from WGS84 involves following two steps: (1) from WGS84 to ECEF (Earth Centered Earth Fixed) coordinate (X, Y, Z):

\[
\begin{align*}
X &= (a/\chi + h) \cos \phi \cos \lambda \\
Y &= (a/\chi + h) \cos \phi \sin \lambda \\
Z &= [a(1-e^2) / \chi + h] \sin \phi
\end{align*}
\]

where \(\phi, \lambda, h\) are the geodetic latitude, longitude and height respectively; \(a\) and \(e\) are the Earth’s semi-major axis and square of first numerical eccentricity of the model ellipsoid respectively; and \(\chi = (1-e^2 \sin \phi)^{1/2}\). and (2) from ECEF to ENU coordinate \((x, y, z)\). In this transform from ECEF to the local ENU coordinate, a local reference point \((X_r, Y_r, Z_r, 3DWF\) center) is required. For a point \((x, y, z)\) in a
ordinate for given 3DWF ENU coordinate. When the 3DWF domain
where $x_1$ and $y_1$ are the terrain height for the 3DWF domain are saved in one
algorithm.

A schematic diagram for the WGS84, ECEF, and ENU coordinate systems for
Fig. 3. The data that is used
to overlay the contour map of the processed terrain data over the
shaded relief (Google Maps) or over the three-dimensional terrain
(Google Earth).

4. Morphological data generation and processing

The fine-scale morphological data, such as buildings and forest
patches are critical for the microscale meteorological model.
Different from larger scale (meso to global) models, the objective
for a microscale meteorological model is to resolve the meteor-
ological variable with grid resolution from meters to hundred
meters. In addition to the large scale weather force, the mechan-
ical and thermal effects of morphological features play an impor-
tant role in local variations. It is natural for the 3DWF to require a
microscale meteorology model system that allows the user to
interactively input data or to generate model output, and to
display the fine-scale morphological data.

Our intention in this part of the model system is not to generate
a very large morphological data set such as for a big city with
hundreds or thousands of buildings, but to generate a small
morphological data set with tens of buildings. The morphological
data generator creates the data set for building geometrical
shapes, dimensions, building heights, forest canopy type, and
canopy height using a digitization and rasterization techniques
(Huyhn et al., 2010). A data object is created for each building or a
forest patch so that the model can use these data. Each object has
information about the latitude/longitude/height of the building (or
vegetation patch), and the corresponding 3DWF model grid points
of a building and forest patch with similar dimensions/properties.
The coding applied the most popular rasterizing approach based on
line sweeping of the polygon decomposed into resulting triangles
(Foley et al., 1990). The detailed description about the
morphological data generator is found in Huyhn et al. (2010).

Fig. 4 shows examples of a building and vegetation morphology
data generator. Before processing the building data, it is necessary
to pull the processed and gridded terrain data using the terrain process-
ing described in Section 3. The 3D building’s height and shapes are
added on top of the terrain data. For the 3DWF building wake
parameterization, where the building intersects the ground surface
and the building top point at the vertical grids are also generated from
this process. The generation of building data using this method is only
for a model domain with a small number of buildings. The building
height can be estimated by either real outdoor measurement or
visually from the image using the building shadows. Google’s driving
through in street view is also a good way to estimate the building
height if it is available in the model domain. The forest canopy data
generation is done in a similar way. The forest type, height, and the
vegetation area index should be created by either outdoor measure-
ment or shadow estimate. Google’s driving through function is also a
good way to estimate the forest canopy height if it is available in the
model domain.

For a major urban with a large number of buildings, the user
should try to get the building data other ways and import the data
using processing of components since the data generation is very
labor intensive. The morphology data can be imported from other
sources if they are available. In some situations, available data
from public or private sectors could be the only source for our
purpose. For an example, in a major metropolitan area, there are
too many buildings in a domain and it is very hard to generate the

ECEF coordinate, the ENU $(x,y,z)$ coordinate is

$$
\begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix} = 
\begin{bmatrix}
-x \sin \phi - \cos \lambda \cos \phi r \\
-\sin \phi \cos \lambda - \sin \phi \sin \lambda \cos \phi r \\
\cos \phi \cos \lambda - \cos \phi \sin \lambda \sin \phi r \\
\end{bmatrix}
\begin{bmatrix}
X_p - X_f \\
Y_p - Y_f \\
Z_p - Z_f \\
\end{bmatrix}
$$

(2)

The inverses of two equations are used to find the WGS84 co-
ordinate for given 3DWF ENU coordinate. When the 3DWF domain
is set up, the domain center’s latitude and longitude are given by a
user. It is taken as a reference point, the latitudes and longitudes of
other 3DWF grid points in WGS84 are computed first. The elevation $(E
(x,y))$ of the terrain is then interpolated using a bilinear interpo-
alation algorithm.

$$
E(x,y) = \frac{E(x_1,y_1)(x_2-y_1)(y_2-y_1) + E(x_2,y_1)(x_2-x_1)(y_2-y_1) + E(x_1,y_2)(x_2-x_1)(y_2-y_2) + E(x_2,y_2)(x_2-x_1)(y_2-y_1)}{(x_2-x_1)(y_2-y_1)}
$$

(3)

where $x_1, x_2$ and $y_1, y_2$ are the ENU coordinates of the SRTM terrain
data points, and on x, y point is a model grid point which
has 4 neighbors $(x_1,y_1; x_2,y_2; x_2,y_2; x_2,y_2)$. The latitude, longitude,
and terrain height for the 3DWF domain are saved in one file that is used
in model initialization, morphology data generation, model numerical
computation, and visualization.

The SRTM data set is a large global data set, consisting of 5 by
5 degree tiles with 3-arcs second increment in east and north
directions. One of the tasks in the terrain processing software is
automatically searching the data set and finding the tiles that
contain the 3DWF domain according to the user’s center latitude
and longitude, the total number of grid points, and the model
resolution. Sometimes, two or four tiles of the SRTM terrain data
are required if the 3DWF is over the border lines of the terrain data
tiles. This task is implemented by computing the latitude and
longitude of the four corners of the 3DWF domain first. The data
tiles are searched using the latitude and longitude information of
the four corners.

During the model domain set up, several parameters, including
domain center latitude, longitude, grid spatial resolution $dx, dy, dz,$
and the numbers of grid points at each spatial directions ($nx, ny,$
and $nz$) should be set up by the user. Error checks and possible
error messages always accompany these user inputs. The center
latitude and longitude can be set up by dragging and clicking the
mouse cursor on the Google Earth. The domain set up is an
interactive process for a user to see exactly where the model
domain edges are located. A user can go back to the domain set-up
function to adjust the domain parameters so that the domain
covers a desired region. It is also important to visually check the
correctness of the interpolated terrain data. The way to do that is
to overlay the contour map of the processed terrain data over the
shaded relief (Google Maps) or over the three-dimensional terrain
(Google Earth).

Fig. 3. A schematic diagram for the WGS84, ECEF, and ENU coordinate systems for
the Earth and their transformation relationships (PM line is the Prime Meridian; $\phi$ and $\lambda$ are
latitude and longitude in WGS84; X,Y,Z for ECEF; and E,N,U for ENU).
building data manually as described above. There is a database of buildings in most U.S. cities generated by lidar altimeter flights (Priestnall et al., 2000; Yu et al., 2010). This data set is generally in vector form such as a shape file. This shape file can be rasterized to make into the microscale meteorological required form. We have used a DTRA (Defense Threat Reduction Agency) supplied shape file for Oklahoma City for a research project (Wang et al., 2005).

5. Integration of microscale meteorological observations with the Google Earth

In this GIS based system, the observation points can be easily drawn on the map or image showing the location, and the values of the observations. For a real-time application, it also can display the sensor reading in real time if the GIS based system is connected with a data collection system. The initial fields of wind, temperature, moisture and concentration based on the objective analysis can be displayed altogether on the map.

Fig. 5 shows an example of real time Doppler radial wind velocity displays for a range-height-indicator (RHI) scanning over a complex terrain. The data from the Doppler wind lidar was first transformed from local lidar coordinates to WGS84 coordinates, the radial wind velocity values were drawn on each polygon formed by the lidar spatial scanning beams and the range gate. The geo-referenced display is especially useful for data visualizations over complex terrain. In some cases, the local thermal circulation of the upslope/downslope flows, the wave production by the terrains, and the interactions between the valley and slope flows, are very intuitive when the lidar image is overlaid on the Google Earth terrain. In this example, the down-slope flow (negative wind velocity) is shown near the mountain surface, while the upper level wind velocity (positive) is in totally different direction. Before applying the Doppler wind lidar data to the model, the wind vector components data has to be retrieved from the radial wind velocity. This topic is outside of the scope of the current paper. However we have put instrument scanning and data collection as a subcomponent of the data pre-processing for the modeling system.

After the data collection and pre-processing, the model is run on a local client computer, followed by error checking. The error checking routines were built into every sub-component of the modeling system. The system alerts the user to check anomalous data to see if it is a real result or an error.

Fig. 5. Doppler wind lidar radial wind velocity ($V_r$) is displayed using Google Earth.
Fig. 6. Visualization of the wind vectors: Top panel, wind vector field 20 m above the ground surface over a complex terrain. Bottom panel, vertical wind profiles over specific locations.
Fig. 7. Top panel: contour plots of total wind speed without direction (a scalar) in a vertical cross-section and a terrain following surface at $z=100$ about ground level. Bottom panel: isosurface plot a simple Gaussian puff model of an air pollutant plume with threshold values of 0.005 μg/m$^3$ (bright orange), 0.003 μg/m$^3$ (orange), and 0.001 μg/m$^3$ (yellow). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
6. Display and visualization of the model results

Model results visualization is an integral part of the model system. It would be very difficult to check the results and communicate them without some forms of graphic display since the model results contain a tremendous amount of data. It is also much easier to connect the model results with its surrounding environments and to make error detection, unknown phenomena discovery more effective. The advanced GIS tools such as Google Maps/Earth offer many aspects of the data visualizations that are very hard to develop with a stand-alone GUI. The multi-layers of 3D terrain, images, and geo-referenced visualization over an entire globe require a great amount of well-processed data which would be very difficult to create and manage in a stand-alone system. The well-implemented map and graphics functions in the Google Maps/Earth such as places, road, borders and labels, zoom/panning/fly by, rotation, lighting, and rendering bring the user to an intuitive and immersive virtual environment. That Google GIS also frequently updates their image is another benefit for the microscale meteorological models since many fine-scale phenomena are produced by the land surface features. Moreover, the well-designed Google API makes the coding of visualizations very easy. It is definitely a great step forward in meteorological data visualization to use the advanced functionalities of the Google GIS.

The following are some examples to show our application of the Google Earth for visualization of the 3DWF model. Again, the model results are first transformed from the local model ENU coordinate to the WGS84 coordinate for the data visualization on the Google Earth. Fig. 6 shows wind vector visualization of a 3DWF output for complex terrain. The top panel draws all the 3D wind vectors on a terrain following surface 20m about the ground surface for a strong wind case. The accelerated wind over the peaks of the mountain and forced flow turning in valleys are evident in this figure. The bottom panel displays several vertical profiles of the wind vectors. This kind of profile is often useful for a user to visualize the vertical variation of the wind over a specific location. Of course, the wind field can be visualized from different perspectives by rotation, zoom/panning.

The scalar variables such as temperature, pressure, moisture, pollutant concentration, and total wind speed (without direction) can also be visualized and can draw the contours for a 2D slice of those variables over the Google Earth. Fig. 7 shows both vertical slicing and terrain following surface of total wind speed over the same complex terrain. An arbitrary slice is also possible by interpolating the grid values of the scalar value to the slice. In some applications, the isosurface of a scalar is necessary to figure out the 3D structure and its evolution. The bottom panel of Fig. 7 shows an isosurface of a pollutant plume. The popular “marching cubes” method is used in generation of the isosurface (Dürst, 1988; Newman and Yi, 2006). Detail implementation on our isosurface generation is referred to Huynh et al. (2012; 2013).

In addition to the vector and scalar displays shown above, we also developed a more dynamic visualization using the Google Earth’s animation capability (not shown in this paper). In this animation, the frames are synchronized with the time slider. Fly-by animation also was developed for simulation of what an aircraft would encounter or simply to track parcel of air in a Lagrangian frame.

7. Summary and conclusion

We have developed an integrated microscale meteorological model system based on an advanced GIS system, the Google Maps/Earth. The system’s user interface and the model run on a local client computer and the functionalities of the GIS are retrieved from Google Maps/Earth servers via internet. Different from the traditional meteorological model system, this novel system takes advantages of enormous resources in map and image data retrieving/handling, four-dimensional (space and time) data visualization, overlaying, and many other advanced GIS features that Google Maps/Earth has offered. In this paper, we have demonstrated how to process the data for initialization and for visualization. We have developed modular components for all of the model system controls and data processing programs which are glued together with the JavaScript language and KML/XML data structure. We have also developed small modular software to convert the model results and intermediate data for visualization and animation. Capabilities such as high-resolution image, street view, and 3D buildings in the Google Earth/Map are also used to quickly generate small-scale vegetation and building morphology data that are required for the microscale meteorological model.

The microscale meteorology model was developed to resolve the atmospheric phenomena near the Earth. Besides the large weather conditions, the fine-scale topographic features such as mountains, building, forest are critical elements that affect the microscale weather. To have this fine-scale background in image and 3D forms definitely improve the model capability. Since older graphics packages for large-scale weather are specifically designed for displaying large scale model results in which fine-scale surface morphology and other characteristics are not resolved, a microscale meteorological system integrated with a modern GIS such as Google Maps/Earth via internet is definitely a good step forward.

Acknowledgment

The authors would like to thank Dr. James Cogan for his careful review of this paper. Google Maps and Google Earth are trademarks of Google, Inc.

References


Google Map/Earth programming API. See the references from web site [http://code.google.com/apis/Earth/].


