Usage, Usability, and Utility of 3D City Models, 02002 (2012)
DOI: 10.1051/3u3d/201202002

Advances in 3D visualization of air quality data

R. San José[1], J.L. Pérez[1] and R.M. González[2]

[1] *Environmental Software and Modeling Group, Computer Science School, Technical University of Madrid, Campus de Montegancedo, Boadilladel Monte, 28660 Madrid, Spain*
[2] *Department of Meteorology and Geophysics, Faculty of Physics, Ciudad Universitaria, 28040 Madrid, Spain*

Abstract. The air quality models produce a considerable amount of data, raw data can be hard to conceptualize, particularly when the size of the data sets can be terabytes, so to understand the atmospheric processes and consequences of air pollution it is necessary to analyse the results of the air pollution simulations. The basis of the development of the visualization is shaped by the requirements of the different group of users. We show different possibilities to represent 3D atmospheric data and geographic data. We present several examples developed with IDV software, which is a generic tool that can be used directly with the simulation results. The rest of solutions are specific applications developed by the authors which are the integration of different tools and technologies. In the case of the buildings has been necessary to make a 3D model from the buildings data using COLLADA standard format. In case of the Google Earth approach, for the atmospheric part we use Ferret software. In the case of gvSIG.-3D for the atmospheric visualization we have used different geometric figures available: “QuadPoints”, “Polylines”, “Spheres” and isosurfaces. The last one is also displayed following the VRML standard.

1. INTRODUCTION

Air pollution modelling is an attempt to describe the functional relation between emissions and occurring concentrations and deposition. Air pollution modelling can give an analysis of the causes which have led to these concentrations. Air pollution models play an important role in science, because of their capability to investigate the importance of the relevant processes. Air pollution models are the only method which can quantify the relation between emissions and concentrations, including the consequences of future scenarios and the determination of the effectively of abatement strategies. Considerable work has been done in the area of air quality modelling and computational science to develop high quality air quality modelling systems.

The air quality models produce a considerable amount of data, raw data can be hard to conceptualize, particularly when the size of the data sets can be terabytes, so to understand the atmospheric processes and consequences of air pollution it is necessary to analyse the results of the air pollution simulations. The visualization of diverse atmospheric data allows making easy to understand the air quality physical and chemical process. In order for this data to provide useful insight into the workings of the atmosphere it must be visualized in a form that users can readily interpret [1].Visualization has been proved to be an effective tool for presenting results of air pollution modelling [2]. Human visual perception offers excellent capabilities that facilitate knowledge construction and the analysis of spatial-temporal relations [3]. Geo-visualization is an effective tool to explore geographic data and communicate information [4].

Visualization of weather and air pollution data helps people to make decisions. 2D display of scientific data can no longer satisfy the increasing requirement on realistic of 3D air quality data sets produce by air quality information systems. The data must be presented in an interesting and easily understandable form; it would facilitate the communication of such to the end users.

The first visualization tools were designed as general visualization systems, to be used for everything from air quality modelling to the design of jet engines. Since they were designed as general purpose systems, only a small subset of the functionality was ever used for air quality modelling. Our study presents the most advanced tools which have been specifically developed to our air quality modelling systems, for the purpose of communicating information to the different audiences.This paper provides several solutions to the atmospheric scientists for the analysis of large amounts of data, which are generated by the atmospheric simulation systems.

2. SPECIAL REQUIREMENTS

The basis of the development of the visualization is shaped by the requirements of the users. There are three main user groups: experts or scientists, stakeholders and general public. The experts aim is to evaluate and analyse their research results with the help of analysis tools. The requirements of the stakeholder need to evaluate different scenarios which are the basis of their decisions. Finally the user group of general public want to know how many unhealthy will be the air quality in area close to them. A further challenge is to design the visualization in such a way that it could be applicable to all of its three stages: the explorative analysis, confirmative analyses and communication [5].

Large masses of spatial data have to be managed by the visualization system [8]. The data source can be different, so different file formats have to be supported [9], for example netcdf, grid, ascii, etc. A system for air pollution data exploration should ensure the support of all existing (three spatial, one temporal, multiple thematic and possible scenarios) dimensions [11]. The presentation and visual analysis of air pollution data cannot be done without geo-referenced coordinates.

Multiple visualization techniques should be included in a suitable software solution. For the purposes considered, the most common shading methods, texture mapping facilities, generation of isolines and isosurfaces, slicing functions, display of vector fields, flow visualization, color editing and reclassification functions, glyph representations, volume rendering and animation are needed. The appropriate use of visualization methods should be assisted by help tools. Different types of legends, scale bars, frames, north arrows, and textual explanations should be part of every system's functionality, but are seldom implemented. Another important feature, a function which assists the user in choosing a meaningful color or symbol legend compliant with the values that you are displaying.

A decisive criterion for the usability of visualization systems is often found in the implemented strategy for human computer interaction, above all concerning 3D-navigation [10]. During the mapping of 3D spaces to 2D screens, problems occur similar to those in traditional mapping [7]. Additionally, interaction and navigation play an important role. Navigation is conventionally done by mouse movement or keyboard input

3. IDV

The Integrated Data Viewer (IDV) from Unidata [12] is a Java tool based on VisAD [14]. It is a very sophisticated product addressed to scientific users. The IDV supports most of the requirements: 2D and 3D temporally varying gridded data [13]. The display capabilities include global and flat 2D and 3D displays, data probes, vertical profiles, time-height plots, topography, isosurfaces, skew-t plots, contours, colour filled contours, wind vectors, wind barbs, streamlines. The IDV also supports animation through space and time, subsetting, derived parameters, zooming, panning and rotating [15]. However, the IDV does not support include GIS data requirements. Fig. 1 is an example of visualization with IDV.

IDV is not ready to Internet outputs, but can be used to produce results over Internet in an operational way using his scripting language ISL, and following the flow chart show in Fig. 2. The air quality data can be displayed automatically with ISL instructions producing animated images, videos or Kmz files to Google Earth. All these formats can be showed through Internet.

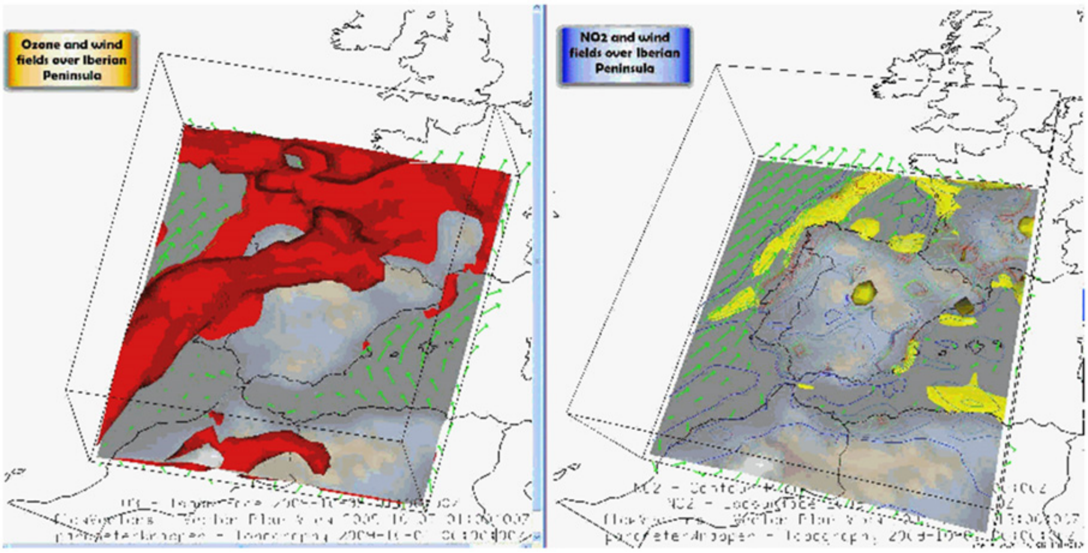

Figure 1. Visualization example with IDV. Air pollution and Wind vector over Iberian Peninsula. Data from MM5-CMAQ model run by UPM.

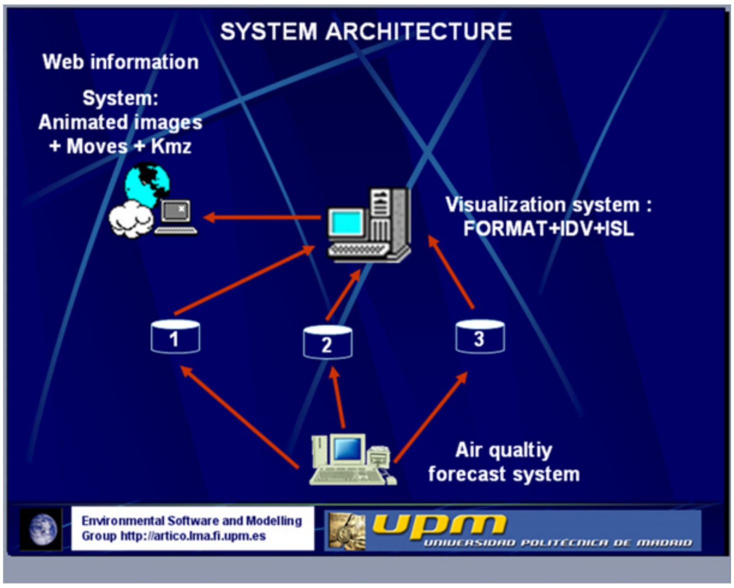

Figure 2. Flow chart to produce Internet information using IDV capabilities.

4. CITYGML

It is a standard that enables the creation and exchange of 3D data, applied to cities and the urban environment, as well as topography. CityGML [16] is realized as an open data model and XML-based format [21] for the storage and exchange of virtual 3D city models. CityGML has 5 levels of representation defined, starting at level 0, appropriate to represent large areas with little detail, and ending at level 4, recommended for use inside buildings and areas with lots of detail. Furthermore, these representations can be combined without conflict, thus highlighting those areas most important representation.

Using CityGML is possible to represent the most important elements of the cities, as buildings and roads, but also the urban material, such as trees. The buildings receive a special treatment; CityGML can incorporate elements as windows, doors, decorative items, etc. It is also possible to incorporate the field and its features, using a dot notation, triangles or curves, as well as bodies of water, such as rivers and seas. CityGML allows the inclusion of generic objects, i.e., objects no defined in the standard and must be specified by the designer. But CityGML not only has graphics features, also allows the inclusion of semantic information [17]. Unlike other standards for representing 3D, CityGML defined classes used to build the city. Thus, a building is not just a set of polygons, but a set of polygons defined as walls, ceiling, floor, windows, doors, etc. Each element of the city has its own class (and a generic class for elements not defined in the standard), so that the buildings are defined as buildings, the roads are defined as road, etc. But also, with CityGML can add features of each element, such as year of construction of each building for example.

We have used this standard for the generation of cities, including buildings (with doors and windows), the ground and trees. We have added the pollution data in the form of soil color, thus achieving a 2D representation [18]. Using the standard features, different representations are been included, each indicated the expected values at a given instant of time. This value is indicated by the color of the ground, and including the numerical value in soil characteristics. The final view can be viewed in Fig. 3.

5. GOOGLE EARTH: COLLADA-FERRET

Google Earth is a "virtual globe" program [19] that allows you to fly anywhere on Earth to view maps, terrain, 3D buildings. It requires files in kml (keyhole mark-up language [21]) or kmz (zipped kml) format. It is becoming extremely popular. It is free and easy to use, you can overlay data sets and it allows you to zoom in/out of our area of interest. Finally, it can display 3D datasets.

Kml files [22] may also include links to images or Collada files, which contains a 3D model. In this case, Google Earth reads and interprets the contents of the Collada file and sets it to the location defined in the Kml file. However, Google Earth defines their own animations, so you cannot show the animations included in the Collada file. These animations are based on a series of moments, which defines the elements that are appropriate at each moment in each point highlighted on the map.

In our case we have created a 3D model of the city of Madrid, using the Google SkechUp software to become Shapefile files (GIS polygonal data) to Collada file format, containing all the buildings in the area selected. The 3D model is show in the Fig. 4.

On the other hand have been obtained maps on the area at different points in time, representing the evolution of temperature over a day in the selected area. These images were obtained with Ferret visualization tool. Subsequently a Kml file was created, this Kml file includes a link to the Collada file from the city of Madrid, and it incorporates the coordinates where it is. Also defined the map to display at every moment of time and location, giving rise to the animation [20]. Finally, all these files have been compressed in a zip file, leading to a Kmz file. Google Earth is able to read this file, and using the coordinates located in Kml file, add the 3D model of Madrid and the first of the images. Google Earth incorporates the progress bar indicating the status of the animation and the dates indicated in the Kml file.

The different tools that have been integrated and the flow chart of the integration to display buildings and atmospheric information in Google Earth are showed in Fig. 5. The final product is showed in the Fig. 6.

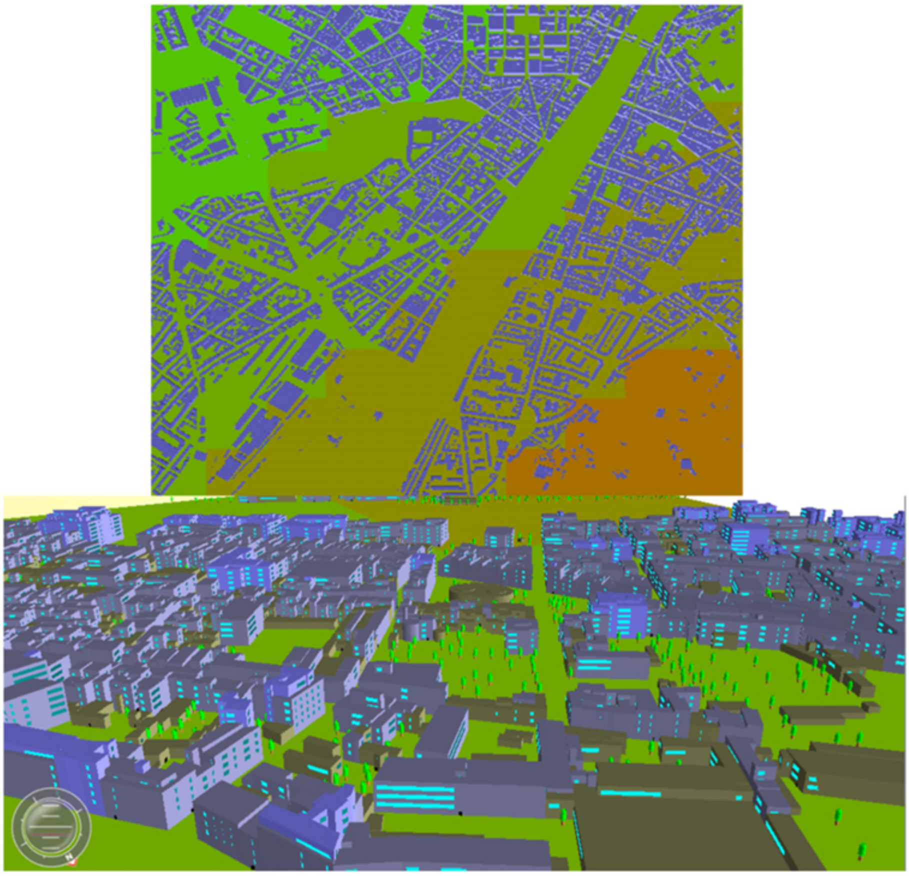

Figure 3. CityGML visualization including 3d buildings and 2d pollution information over Florence (Italy) developed by UPM. Top view (upper) and Zoom-in with LOD 4 level (bottom).

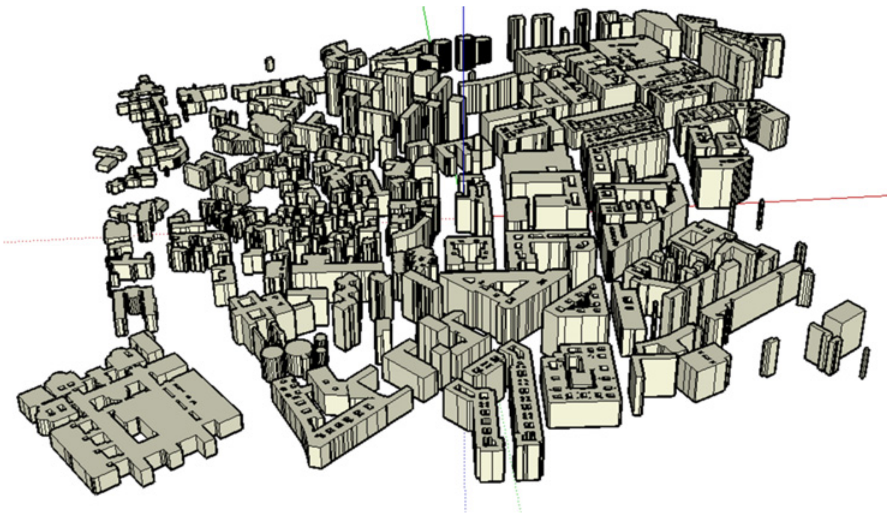

Figure 4. 3D buildings displayed using COLLADA. Madrid (Spain) area.

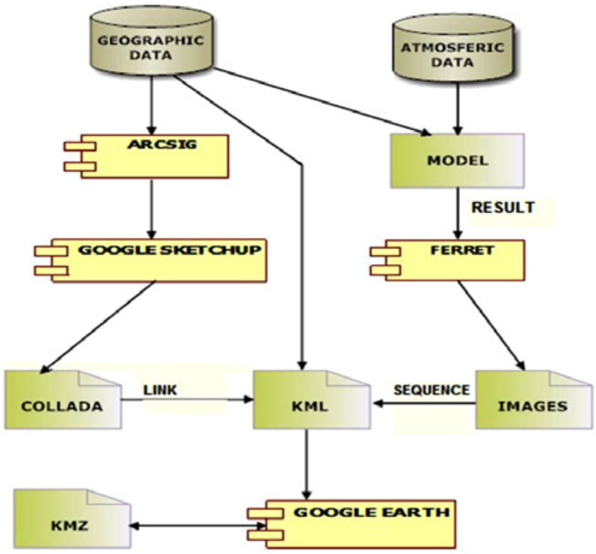

Figure 5. Flow chart to the integration of the different tools, ArcGIS, Google SketchUp and Ferret.

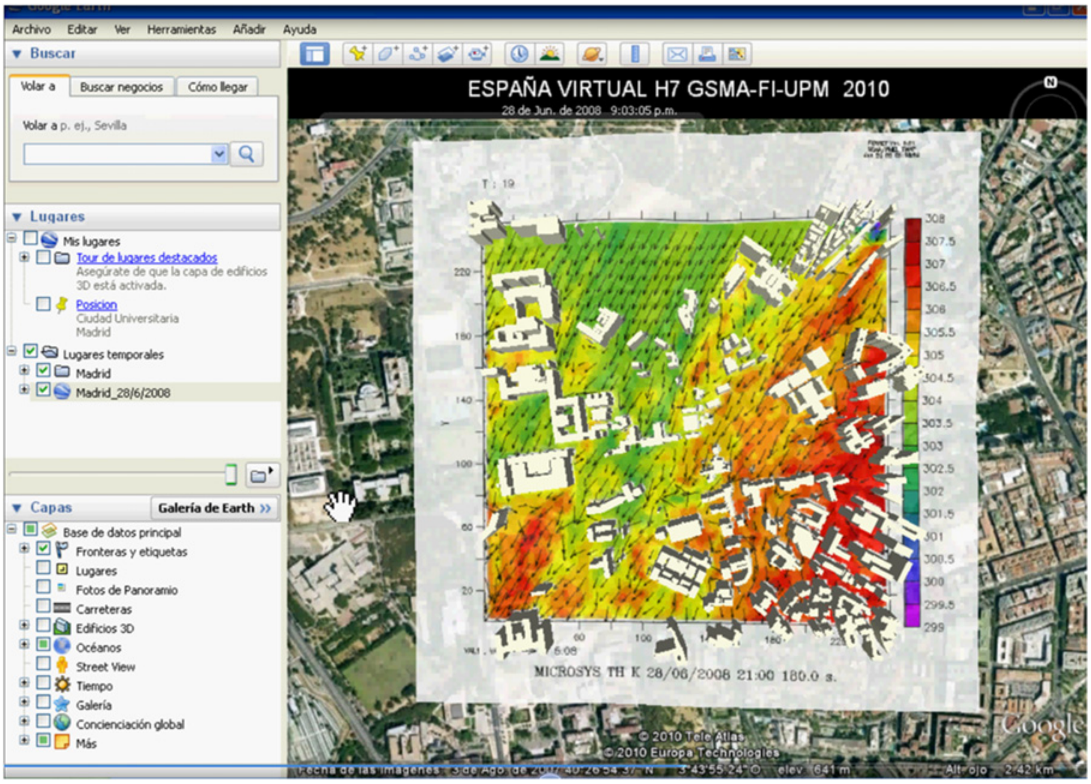

Figure 6. Google Earth visualization of temperature and winds plus 3D buildings developed combining FERRET maps and COLLADA 3D model.

6. GVSIG-3D

GvSIG is a Geographic Information System with a simple and friendly user interface [23]. It has capabilities of handling many raster and vectorial data formats. It is free and open source software, so each developer can implement different add-ons. In our case, we need a new functionality that allows to the visualization of atmospheric data. Our final objective is the visualization of 3D atmospheric data with the gvSIG plus its 3D extension. It allows combine 3D data from atmospheric models and buildings in a urban environment. For the atmospheric visualization we use different geometric figures available in gvSIG-3D and finally we have implemented the representation of 3D volumetric data using isosurfaces in gvSIG-3D. One important decision to take is how can be displayed the atmospheric information.

The main core of the 3D extension is the "Framework OsgVP (OpenSceneGraph Virtual Planets). OsgVP has a library called "Osgvfeatures". It allows draw vectorial data, text, points, lines, polygons and simple geometric figures, for example spheres. The developer can set a transparency value and add a colour.

City 3D models provide a more realistic representation and the aesthetic plays a very important role, since a more realistic representation will be easier to the end user the interpretation and treatment of the data. For it, in the virtual representation of objects, such as the buildings, require to equip to them with the greater possible realism, one of the used solutions more is to add textures to those objects. They are multitude of standards and formats of representation of 3D models. In this research we used COLLADA, format accepted by the gvSIG-3D tool. We have implemented a process to build the final 3D model in the standard COLLADA from the geographic data available. This process is equal than was applied with the Google Earth visualization. In this case, a new JAVA tool has been developed to add texture to building surfaces. In order to add textures it is necessary to analyse the structure of file COLLADA, identifying the different surfaces from each building. Different textures for the walls and the tile roofs have been used.

The visualization of the model will carry out with the 3D extension of the gvSIG program. In a new 3D view will be necessary to create a virtual layer, after that it is ready to import our COLLADA model. To improve the visual aspect of the 3D model, several textures have been added. The final result can be observed in Fig. 7 where the last buildings present a better and more realistic aspect.

Although there are different software tools for the visualization of atmospheric data, these have limitations to include the representation or urban elements. So our objective is to integrate the capabilities of geo-referencing and visualization of 3D elements in gvSIG.-3D with the visualization of the 3D atmospheric data.

In this work, we show how the atmospheric data can be displayed using "QuadPoints", "Polylines" and "Spheres". May be the spheres are the most appropriate for the atmospheric data, due to it is a 3D figure [24]. Each grid cell value is modelled by a sphere with a colour. The colour is based on a linear colour scale. Also a transparent degree is added to be able to visualize the urban buildings.

In Fig. 8 we can see four examples of the representation of atmospheric data with spheres. The results are temperature data. Red colour means highest values and green colour means lowest values.

Other available representation elements from OsgVP have been used to represent the atmospheric data. Although these elements do not have 3D aspect, the results obtained for the visualization of grid cell 3D data are quite good and can to be another form to represent the results of the models. The Fig. 9 shows the representation that can be obtained with the same 3D data used with spheres, but now using "Polylines" and "Quadpoints".

Other possibility is the use the isosurfaces. The isosurfaces have been generated using the "marching cubes" algorithm [6]. It is often used for building a polygonal mesh from 3D scalar data, as temperature or concentrations of pollutants. The modelling domain is spited into cubes with the corresponding value in each vertex. The algorithm runs through all cubes and if one or more corners of the cube have values less than the isovalue selected, them this cube is part of the isosurface. Having determined the sides of the cube are intersected by the isosurface, the algorithm creates triangular polygons which divide the

Figure 7. Visualization of Madrid 3D buildings with gvSIG.-3D with textures and location.

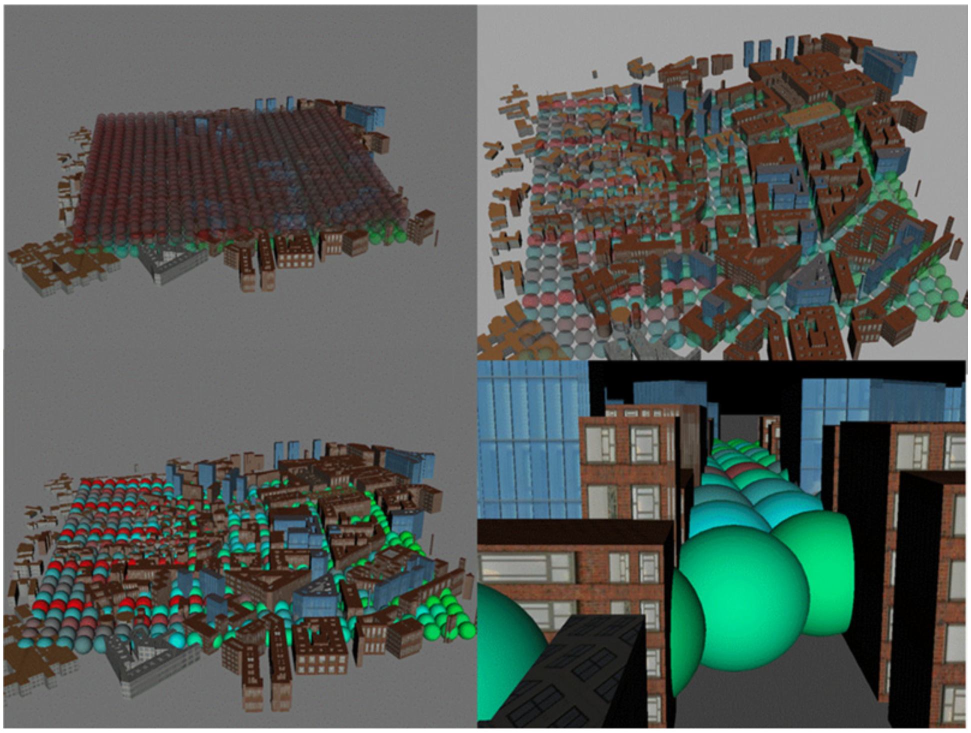

Figure 8. 3D Visualization of temperature data and buildings in gvSIG.-3D with 3 vertical layers (upper-left), surface layer (upper-right), without transparency (bottom-left) and zoom-in over a street (bottom-right).

cube into regions that are inside the isosurface and others are not. For the graphical representation of the isosurfaces in gvSIG.-3D we have used the class "Polygon" to represent the triangles which form the isosurface. The colour of the isosurface depends of the isosurface value. Different isosurfaces can be displayed at the same time.

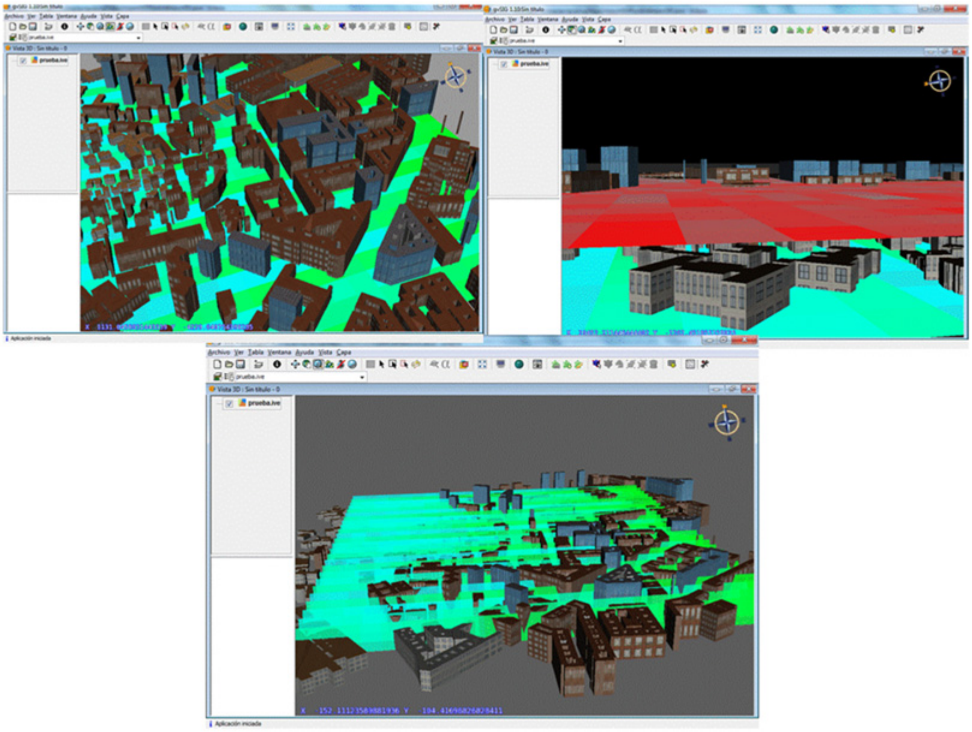

Figure 9. Representation of 3D temperature data in gvSIG.-3D. The green colours represent zones with lowest values of temperature, blues mead values and red colours are for highest values of temperature .One vertical level with “Polylines” (bottom), “Quadpoints” (upper-left) and “Quadpoints” with 2 vertical levels (upper-right).

In the Fig. 10, we show several examples of visualization of isosurfaces with the buildings in the system gvSIG-3D. Finally to make notice the high number of polygons to represent to visualize an isosuperficie, in the previous examples we are around 100000 triangles, in the realised performance tests from 200000 triangles, the system gvSIG-3D begins to have problems of memory at the time of handling such amount of polygons. The degree of quality of the isosuperficie is going to depend as large as enmeshes used.

7. VRML

The Virtual Reality Modeling Language (VRML) [25] is a file format for describing 3D interactive worlds and objects. It may be used in conjunction with the World Wide Web [26]. It may be used to create three-dimensional representations of complex scenes such as illustrations, product definition and virtual reality presentations. VRML is capable of representing static and animated objects and it can have hyperlinks to other media such as sound, movies, and image. Interpreters (browsers) for VRML are widely available for many different platforms as well as authoring tools for the creation VRML files. VRML supports an extensibility model that allows new objects to be defined and a registration process to allow application communities to develop interoperable extensions to the base standard [29]. There is a mapping between VRML elements and commonly used 3D application programmer interface (API) features [27].

To represent the city buildings in VRML se have used “IndexedFaceSet” nodes. This type of nodes is a collection of faces that allow building any type of geometric figures as buildings. One modelled all sides of the buildings using the node described above.

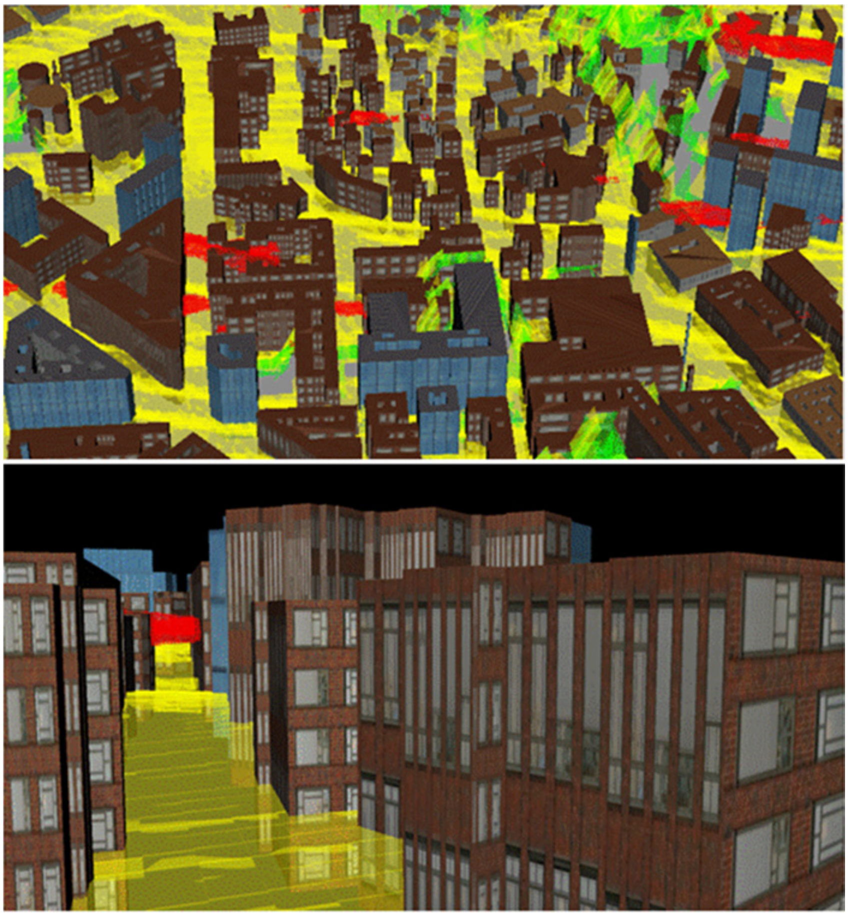

Figure 10. (Upper) Top view of 3 isosurface that represent low values of contamination in colour green, middle values in yellow colour and high values in red. (Bottom) Surfing through the streets to identify "hot spots".

The isosurfaces have been generating using the "marching cubes" algorithm [5] as in the gvSIG-3D approach. In this phase, we have 2 components, buildings and isosurfaces, now the final step is the integration of buildings and isosurfaces to generate the final product, which can be viewed by the end users with the corresponding VRML viewer [28]. An example of the final results is shown in the Fig. 11. The user can navigate through the streets and see the distribution of the atmospheric data between the buildings as the Fig. 12

8. CONCLUSIONS

In this paper we described out present state of developing a visualization techniques that integrates diverse sources of data. Our contribution presents new tools, which are being developed by software integration. This paper provides several solutions to the atmospheric scientists for the analysis of large amounts of data, which are generated by the atmospheric simulation systems. Instead of terabytes of raw number they can utilize the 3D interaction and visualization capabilities provided for the different tools.

The first one IDV is a generic tool that can be used directly with the simulation results. The second one tools are specific applications developed by the authors, which are the integration of different tools

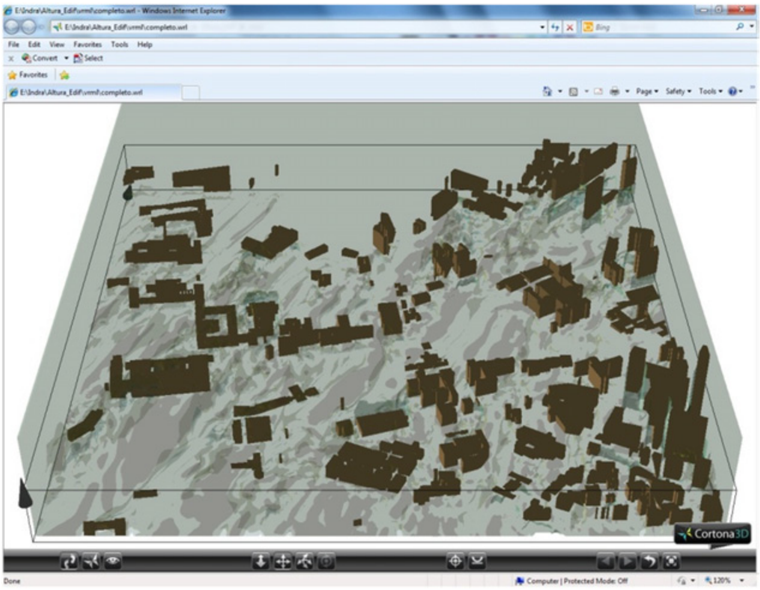

Figure 11. Example of visualization with VRML of buildings and isosurfaces of temperature over Madrid area.

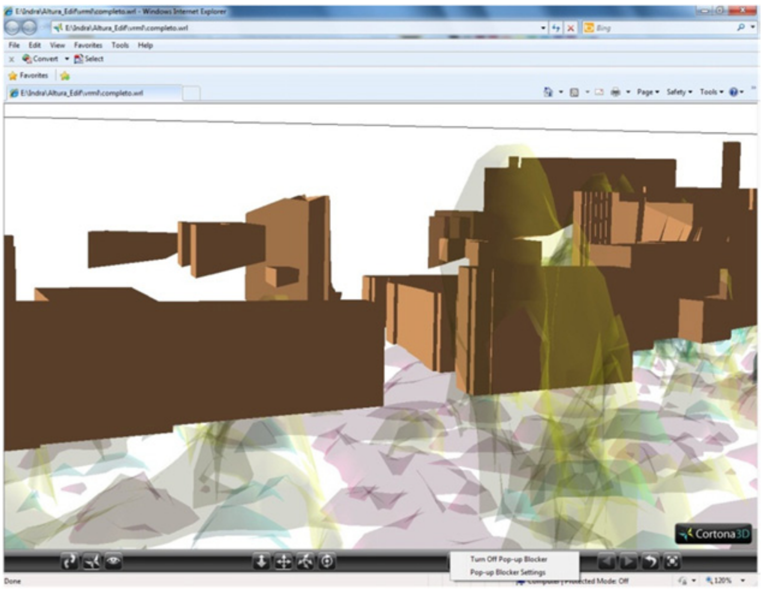

Figure 12. Example of interaction of the user with VRML.

and technologies. The second group is more focus on 3D urban environments, which combine urban morphology (buildings), and atmospheric data. So if you want display atmospheric results combined with urban morphology (buildings), it is necessary to combine different tools to produce the desired information.

In the case of the buildings the COLLADA format is showed one of the most suitable, because can be used in different systems as gvSIG-3D and Google Earth. In order to carry out this visualization it has been necessary to make a 3D model from the buildings data, which has been equipped with textures to increase the realism of the representations.

The atmospheric data representation is the part more complex. In this paper we show 2 different ways to show atmospheric data, with spheres in gvSIG-3D and isosurfaces with VRML and gvSIG-3D.

In case of the Google Earth approach, to represent building we use COLLADA and for the atmospheric part we use Ferret software. For the atmospheric visualization we have used different geometric figures available in gvSIG-3D: “QuadPoints”, “Polylines”, “Spheres” and isosurfaces. For the isosurfaces generation the “marching cubes” algorithm can be used and for the graphical representation in gvSIG-3D triangles will be used. Several examples of representation of 3D temperature data with buildings are showed, with different color scales and transparency levels.

All this visualization tools allow create images to represent abstract data (atmospheric data). They apply different techniques to transform data into visual information. Several steps must be done during the generation process. These steps can be implemented by different software tools, and the final product is an integrated software environment for visualizations of large datasets. The integration scheme follow the dataflow paradigm, different tools are linked together in a pipeline. This option produces modular visualization environments, which can be easily adapted to the data and scientific requirements. In case of 3D visualization a more difficult problems have to been solved. 3D data usually needs to be converted to an alternative geometric form in order to send it to the rendering module, as the marching cubes algorithm.

The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the Centro de Supercomputación y Visualización de Madrid (CESVIMA) and the Spanish Supercomputing Network. The authors also acknowledge the ESPAÑA VIRTUAL CENIT project funded by the Spanish Ministry of Science and Technology.

References

[1] Dobashi, Y., Kaneda, K., Yamashita, H., Okita, T. and Nishita, T. A simple, efficient method for realistic animation of clouds. In SIGGRAPH ’00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques (pp. 19–28). New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. (2000ç)

[2] Koussoulakou, A.. Spatial-temporal analysis of urban air pollution. In Visualization in Modern Cartography. MacEachren, A. M. and D.R.F. Taylor (ed). London: Pergamon, pp. 243–269 (1994)

[3] Gahegan, M., Wachowicz, M., Harrower, M., and Rhyne, T-M., The integration of geographic visualization with knowledge discovery in databases and geocomputation. Cartography and Geographic Information Science **28**(1): 29–44 (2001)

[4] MacEachren, A. M. and Monmonier, M., Geographic Visualization: Introduction. Cartography and Geographic Information Systems **19**(4): 197–200 (1992)

[5] H. Schuhmann and W. Muller. Visualisierung - Grundlagen und allgemeine Methoden (2000)

[6] Goodchild, M.F., Buttenfield, B. and Wood, J. Introduction to visualizing data validity. In: Hearnshaw, H.M. and Unwin, D.J. Editors, 1994. Visualization in Geographical Information Systems Wiley, New York, pp. 141–149 (1994)

[7] Gordon V. Bancroft, Fergus J. Merritt, Todd C. Plessel, Paul G. Kelaita, R. Kevin McCabe, Al Globus, FAST: a multi-processed environment for visualization of computational fluid dynamics, Proceedings of the 1st conference on Visualization ’90, October 23–26 (1990)

[8] Craig Upson, Thomas Faulhaber, Jr., David Kamins, David H. Laidlaw, David Schlegel, Jefrey Vroom, Robert Gurwitz, Andries van Dam, The Application Visualization System: A Computational Environment for Scientific Visualization, IEEE Computer Graphics and Applications, v.9 n.4, p. 30–42, July (1989)

[9] R. B. Wilhelmson, B. F. Jewett, C. Shaw, L. J. Wicker, M. Arrott, C. B. Bushell, M. Bajuk, J. Thingvold, J. B. Yost, A study of the evolution of a numerically modeled severe storm, International Journal of Supercomputer Applications and High Performance Engineering, v.4 n.2, p. 20–36, Summer (1990)

[10] Hibbard, W.; 4-D display of satellite cloud images. Digital image Proc. and Visual Comm. Tech. in Meteor., Cambridge, SPIE, 83–85 (1987)

[11] William Hibbard, Charles R. Dyer, Brian Paul, Display of scientific data structures for algorithm visualization, Proceedings of the 3rd conference on Visualization '92, October 19–23 (1992)

[12] Baltuch, M. S.: Unidata's Internet Data Distribution (IDD) System: Two Years of Data Delivery. Preprints, 13th Intl Conf. on IIPS for Meteorology, Oceanography and Hydrology, 478–480 (1997)

[13] Domenico, B., J. Caron, E. Davis, R. Kambic and S.Nativi: Thematic Real-time Environmental Distributed Data Services (THREDDS): Incorporating Interactive Analysis Tools into NSDL, J. Digital Information, 2(4) (2002)

[14] Hibbard, W.: VisAD: Connecting people to computations and people to people. Computer Graphics, 32(3), 10–12 (1998)

[15] Murray, D. and C. Murphy: Community Software Design – The Unidata Java-based MetAppsProject. Preprints, 17th Intl Conf. on IIPS for Meteorology, Oceanography and Hydrology, 367–368 (2001)

[16] Kolbe, T. H.; Gröger, G. & Plümer, L. CityGML - 3D City Models for Emergency Response. In Geoinformation-Technology for Emergency Response, ISPRS book series, Taylor & Francis (2006)

[17] Kalfoglu, Y., Schorlemmer, M., Ontology mapping: the state of the art. Knowledge Engineering Review, Vol 18:1, 1–31 (2003)

[18] Shvaiko, P., Euzenat, J. A survey of schema-based matching approaches. Journal ondata semantics, Vol. 4, 146–171, Springer Verlag (2005)

[19] A. Chen, G. Leptoukh, S. Kempler, C. Lynnes, A. Savtchenko, D.Nadeau, and J. Farley, "Visualization of A Train vertical profiles using Google Earth," Computers & Geosciences, Vol. 35, pp. 419–427 (2009)

[20] Y. Yamagishi, K. Suzuki, H. Tamura, H. Nagao, H. Yanaka, and S. Tsuboi, "Integration of geophysical and geochemical data", EOS Transaction of American Geophysical Union 87 (52) AbstractIN11A1142 (2006)

[21] M. Barnes, (Ed.), "COLLADA– Digital Asset Schema Release 1.4.1specification", Sony Computer Entertainment Inc., June 2006

[22] Google Inc., "KML reference", `http://code.google.com/apis/kml/documentation/kmlreference.html`, Google Inc. (2009)

[23] AlvaroAnguix**,** Laura Díaz, Mario Carrera. Open Source in Spain: the gvSIG Project. A GIS Desktop Solution for an Open SDI. Geoinformatics, 11(8): 36–39, December (2008), ISSN 1387–0858

[24] FEKETE G.: Rendering and managing spherical data with sphere quadtrees. In Proceedings First 90IEEE Conference Visualization Visualization 90 (1990), pp. 176–186

[25] The Virtual Reality Modeling Language. International Standard ISO/IEC 14772-1:1997. `http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm` (1997)

[26] Rikk Carey and Gavin Bell. The Annotated VRML 2.0 Reference Manual. Addison-Wesley Developer Press (1997)

[27] Michael Bailey Cherilyn Michaels. VizWiz: A Java Applet for Interactive 3D Scientific Visualization on the Web. In Proceeding IEEE Visualization '97, pages 261–267 (1997)

[28] R. Grosso and Th. Ertl. Progressive Iso-Surface Extraction from Hierarchical 3D Meshes. In Proceedings EUROGRAPHICS'98 (1998)

[29] Rodget Lea, Kouichi Matsuda, and Ken Miyashita. JAVA for3D and VRML Worlds. New Riders Publishing (1996)