name	Student number
R. Peters	1361511
B.S. Rijnders	1357069

Viewing GML models in Bing maps

8 January 2010

Approach

The first approach was to find out to which file the gml files needs to be converted in order to import them into Bing maps. The following page has been used: http://msdn.microsoft.com/en-us/library/cc966728.aspx

Apparently .obj wavefront files were supported by Bing maps. The conversion method is written in 'CityGML2OBJ.py'. For this conversion lxml has been used to read the gml files. This conversion will be discussed later in more detail. To position the model, the centroid of the model has been used and a transformation has been applied. The transformation was done with ogr module.

The uploading of the gml models has been done in php. The upload form was created using php. Also the table which loads 3D models from mysql database is done with php. Once a file has been submitted using 'uploader.php' it is processed by 'CityGML2OBJ.py'. This model will be stored in a mysql database, this is also done by 'CityGML2OBJ.py'. The viewer '3d_bing_maps_viewer.php' loads Bing maps and shows a table containing the uploaded models.

In table 1 an overview of used libraries/modules can be seen. A short description of the files which are included in the zip package can be seen in table 2

Table 1: Overview of libraries/modules that have been used

Name	Version	Description
_mysql	1.22 for python 2.5	Needed for communication between python and mysql database
datetime	Python 2.5	Used when model is stored in mysql database
lxml	2.2.4 for python 2.5	For reading gml files
mod_python	Python 2.5	Needed to redirect user back to bing maps viewer after converting gml to obj
osgeo	GDAL-1.6.1 for python 2.5	Used ogr for crs transformation
StringIO	Python 2.5	To store vertices and faces
sys	Python 2.5	Get passed arguments

Under Ubuntu linux 9.10 everything worked under python 2.6. Due to an incompatibility issue of the _mysql module with python 2.6 under Windows, it is recommended to use python 2.5 on the Windows platform.

name	Student number
R. Peters	1361511
B.S. Rijnders	1357069

Table 2: Short description of files that were included in the zip package.

File	Description
3d_bing_maps_viewer.php	Loads Bing maps, show available models in a table using mysql database and it contains an upload form
CityGML2OBJ.py	Does the conversion from gml to obj file and stores the model in mysql database.
footprints_extruded.obj	Final result after converting the 'footprints_extruded.xml' file
footprints_extruded.xml	Demo file that has been used for testing
sql	Contains the structure of mysql database table
uploader.php	Does the uploading and validation of the file (only a xml file is allowed). If file uploaded it will execute CityGML2OBJ.py to convert the gml file to an obj file.

Converting CityGML to wavefront .obj

The wavefront .obj format appeared to be the only supported fileformat which is currently supported by the Ajax Bing maps. For importing CityGML files it was therefore required to first convert the CityGML to .obj. A .obj file has the following basic structure:

v point1_x point1_y point1_z

v point2_x point2_y point2_z

v point3_x point3_y point3_z

v point4_x point4_y point4_z

…etc

f 3 2 1

f 2 3 4

f 1 2 4

..etc

The first lines starting with **v** defines all the points (vertices) in the model. Every **v** is followed by the x, y, z coordinates of the point separated by spaces.

The next block of lines starting with **f** defines the faces. The **f** is followed by a set of numbers. Every number refers to a vertex. For example the number '1' refers to the vertex defined in the first **v**-line, the number 2 refers to the vertex defined in the second **v**-line, etc. So all vertices referenced by the numbers on a **f**-line represent one face in the model.

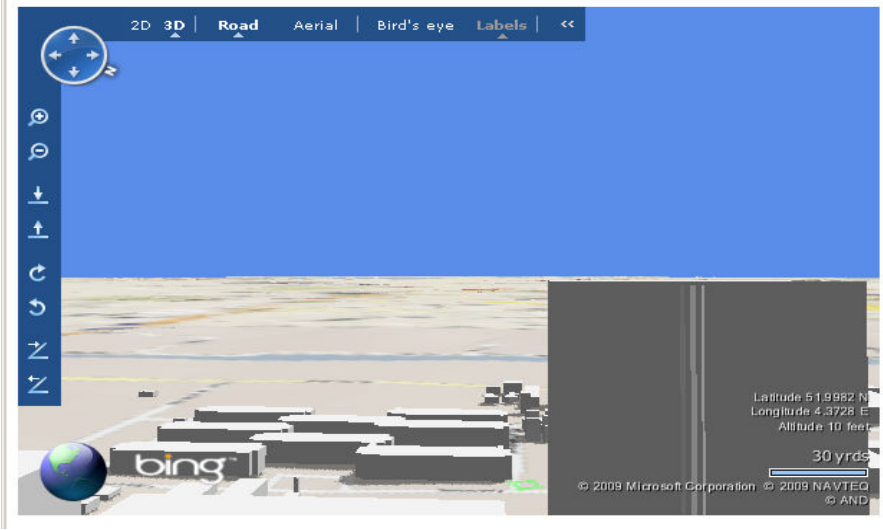
name	Student number
R. Peters	1361511
B.S. Rijnders	1357069

In order to create a .obj model from a CityGML file containing e.g. The buildings of the TU Delft campus the following algorithm was used:

- For every building (CityObjectMember)
    - For every face (LinearRing)
        - for every point
            - add all the points to a set *S* (every unique point ocures only once in the set)
    - Convert *S* to a list and add it to *pointlist*, so that every element (point) has an index.
    - For every face
        - add a **f**-line
        - for every point
            - look the index of the point up in the *pointlist*
            - add the index to the current **f**-line
- Now *pointlist* is completed, print all **v**-lines in the order in which the points occure in *pointlist*.

Encountered problems

There are some accuracy problems when models are loaded into Bing maps. This can be seen when 2 object files are loaded into Bing maps. One containing the whole campus and the other one contains 1 building of the campus. It can be seen that when those 2 models are added to Bing maps, that the position of the same building differs a couple of meters. Probably there are several reasons for this. It might be possible that the calculation of the center point is not accurate enough. Another reason can be that the accuracy that Bing maps uses differs from what is used for placing the 3D models. The latitude and longitude in Bing maps have an accuracy up to 4 decimals, see figure 1.



Latitude 51.9982 N
Longitude 4.3728 E
Altitude 10 feet
30 yrds
© 2009 Microsoft Corporation © 2009 NAVTEQ
© AND

Figure 1: Bing maps with longitude and latitude on the right bottom side.

name	Student number
R. Peters	1361511
B.S. Rijnders	1357069

The 3D models are imported with an accuracy up to 9 decimals. This can be seen from database or source code of our demo server.

For the transformation of the center point from RDnew to WGS-84 OGR was used. Due to a bug in one of the dependencies of OGR, proj4, transformations from and to RDnew are slightly inaccurate. This bug was circumvented by 'hardcoding' the RDnew (EPSG:28992) definition in the python code.

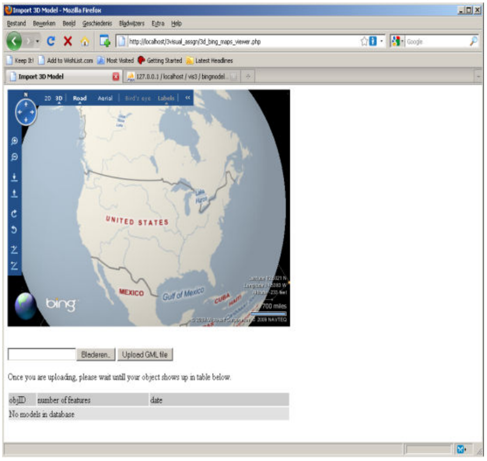
Another problem was the wrong default orientation of the 3D object model. All 3D objects that are added to Bing maps are therefore rotated 90 ˚ counter-clockwise about the x-axis. This rotation is done in “3d_bing_maps_viewer.php” by initializing a new instance of “VEModelOrientation”.

A final minor problem is that especially some larger models appear to be floating above ground at some points. This is probably because of the elevation model used by Bing maps. In the tested CityGML files all buildings were positioned on the same plane with height zero. If you would place this plane on a minor slope one edge would float a bit above the slope, and the opposite edge would intersect with the slope. This is also what happens in Bing maps.

Final results

The final result can be seen on:
http://gw.vrlab.tudelft.nl/3dve/bing/

Before uploading a file the following screen shows up, see figure 2.


Import 3D Model - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
http://localhost/3visual_assgn/3d_bing_maps_viewer.php
Google
Keep It! Add to WishList.com Most Visited Getting Started Latest Headlines
Import 3D Model
127.0.0.1 / localhost / vis3 / bingmodel
N
2D 3D Road Aerial Bird's eye Labels
UNITED STATES
Lake Huron
MEXICO
Gulf of Mexico
CUBA
HAITI
700 miles
bing
2009 NAVTEQ
Bladeren..
Upload GML file

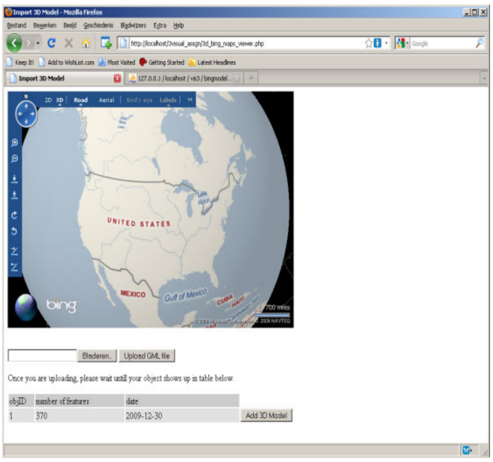
Once you are uploading, please wait untill your object shows up in table below.

objID	number of features	date
No models in database		

Figure 2: Initial Bing maps demo screen.

The files can be uploaded using the ‘Upload GML file’ button. Only .xml files are allowed to be uploaded. After hitting the upload button the browser will refresh twice. First the browser will refresh to process/validate the upload data (this is done by “uploader.php”). The second time python will redirect (refresh browser) the user back to the Bing maps viewer. This is done after the python script has converted the gml file to object file. The final result will look like this, see figure 3.

name	Student number
R. Peters	1361511
B.S. Rijnders	1357069

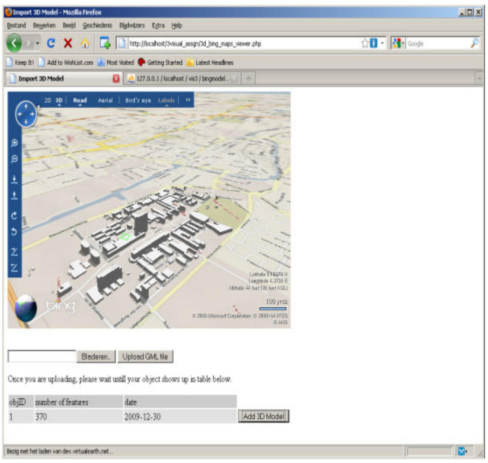

Import 3D Model - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
http://localhost/3visual_assgn/3d_bing_maps_viewer.php
Keep It! Add to WishList.com Most Visited Getting Started Latest Headlines
Import 3D Model
127.0.0.1 / localhost / vis3 / bingmodel...
2D 3D Road Aerial Bird's eye Labels
UNITED STATES
MEXICO
Gulf of Mexico
CUBA
HAITI
700 miles
2009 NAVTEQ
bing
Bladeren...
Upload GML file
Add 3D Model

Once you are uploading, please wait until your object shows up in table below.

objID	number of features	date
1	370	2009-12-30

Figure 3: Bing maps demo screen after uploading a cityGML file of TU Delft campus.

Now the 3D model can be added by clicking on “Add 3D Model”. Bing maps will automatically go to the imported model, see figure 4.


Import 3D Model - Mozilla Firefox
Bestand Bewerken Beeld Geschiedenis Bladwijzers Extra Help
http://localhost/3visual_assgn/3d_bing_maps_viewer.php
Keep It! Add to WishList.com Most Visited Getting Started Latest Headlines
Import 3D Model
127.0.0.1 / localhost / vis3 / bingmodel...
2D 3D Road Aerial Bird's eye Labels
bing
Bladeren...
Upload GML file
Add 3D Model
Bezig met het laden van dev.virtualearth.net...

Once you are uploading, please wait until your object shows up in table below.

objID	number of features	date
1	370	2009-12-30

Figure 4: Final 3D model of TU Delft campus loaded in Bing maps

Resources

Link	Description
http://msdn.microsoft.com/en-us/library/cc966728.aspx	Information about importing a 3D model into Bing maps
http://sourceforge.net/projects/mysql-python/	_mysql python module
http://www.modpython.org/	Used for executing python scripts on apache server
http://en.wikipedia.org/wiki/Centroid	Information about how to find centroid of polygon
http://pypi.python.org/pypi/lxml/2.2.4	lxml python module
http://pypi.python.org/pypi/GDAL/	osgeo python module
http://code.google.com/p/citygml2obj/	Svn repository used during development.