Photogrammetric Week '15 Dieter Fritsch (Ed.) Wichmann/VDE Verlag, Belin & Offenbach, 2015

CityGML goes to Broadway

Thomas H. Kolbe, Barbara Burger, Berit Cantzler, Munich

ABSTRACT

This paper presents the results of a project in which a semantic 3D city model of New York City (NYC) has been created based on datasets provided in the NYC Open Data Portal. It is shown how different 3D feature types can be derived from existing public 2D and 2.5D datasets using spatial and semantic transformations together with (some) photogrammetric analyses. The realized process integrates 26 different datasets from five departments of the NYC administration. The resulting 3D city model is represented in a homogenized and integrated way using the international standard CityGML of the Open Geospatial Consortium. It comprises all NYC buildings, land parcels, roads, parks, the digital terrain model, and water bodies – all with 3D geometries. To the best of our knowledge it is the first publicly available big 3D city model of a large city in the USA which is based on official governmental data. The paper gives an overview on the many challenges that were faced regarding the handling of the huge data volume, semantic transformations, linking of different datasets, spatial corrections and also highlights some methodological aspects. Finally, we provide information where the dataset and the transformation tools can be freely downloaded.

1. A SEMANTIC 3D CITY MODEL OF NEW YORK CITY

Virtual 3D city models are a spatial representation of the relevant entities within cities. 3D city models have traditionally been used for many application areas like urban planning, navigation, tourism, disaster management, training simulation and gaming. For most of these application areas the 3D visualization is of highest importance. Hence, the representation of 3D geometry and graphical appearances is sufficient in many cases.

Nowadays, 3D city models are increasingly being used as information platforms for complex simulations like environmental analyses (air pollution, city climate, noise dispersion) or energy planning. In order to be usable for these kinds of applications, the 3D objects do not only have to be represented spatially and graphically, but also the object meanings, their thematic properties and their logical interrelationships must be represented. Such semantic 3D city models provide the input data for diverse simulations and analyses on the one hand. On the other hand they integrate the results by enriching the respective city objects with the simulation output data. An example is given by (Kaden & Kolbe, 2014) where the semantic 3D city model of Berlin is being used to estimate the heating, electrical, and warm water energy demands of all individual residential buildings. The estimated energy demands are added to the building objects as user defined attributes which in the following are being used in the assessment of the specific energy demands of streets, quarters, and city districts. Similar approaches were presented by (Agugiaro et al., 2015) and (Nouvel et al., 2013). Semantic 3D city modeling can be compared to Building Information Modeling (BIM), however with fewer details on the one side, but full spatial coverage of all urban entities of different types on the other side, including different types of man-made constructions and natural features like vegetation and water bodies.

While in Europe, the Middle East, and the Far East semantic 3D city models are becoming more popular and common (c.f. Gröger & Plümer, 2014), there has not been much movement in the USA so far. When it comes to 3D city modeling, in the US it is nearly always seen in the light of 3D visualization. When it comes to semantic modeling, in the US the focus is almost only on BIM. However, BIM data is only available for newly created buildings, and the widely used BIM standard IFC is limited to the representation of building models so far. This was the starting point for two master theses carried out by the two co-authors of this paper at the Chair of Geoinformatics at Technische Universität München.

In previous student projects we already examined the different Open Data Portals of a number of US cities like New York City, San Francisco, Chicago, and Philadelphia. A look at the situation in NYC showed that there are already a number of 3D models of NYC, but they are a) commercial, b) mostly used for visualization, or c) only accessible as a 3D visualization (e.g. Google Maps and Google Earth, Apple Maps etc.). While in Open Streetmap some work is being done on integrating 3D building models and the NYC Open Data, there is no semantic 3D city model yet providing also the DTM, roads, vegetation, land parcels, water bodies etc. in a homogenized model.
Now the questions of the master theses were how to automatically generate a semantic 3D city model of NYC from the public datasets and what different and separately stored datasets can be integrated in a sensible way. This paper highlights and summarizes the results of this work. The resulting 3D city model dataset will be made available to the public and can then be used both by stakeholders and other interested parties from NYC and – since its volume is huge – by developers for creating new applications and GIS developers to benchmark and improve the performance of their tools and systems. Details on the resulting dataset are given in section 4.

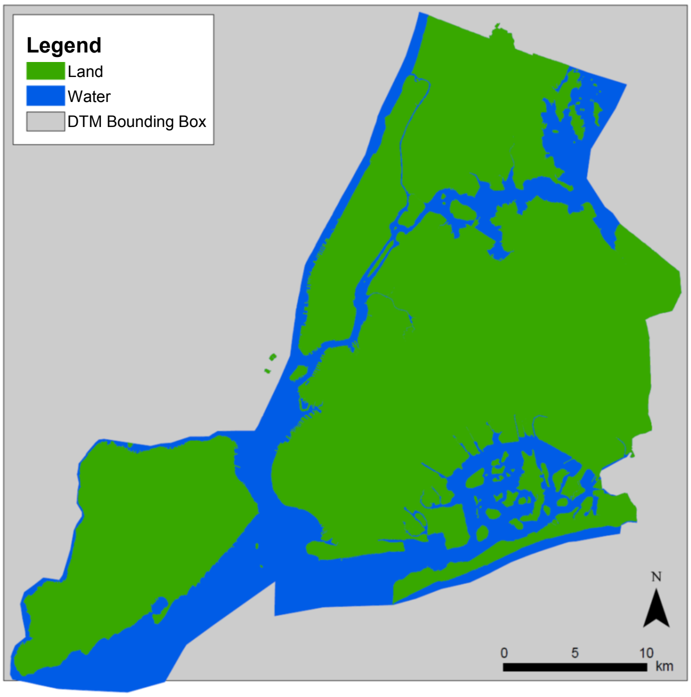
1.1. Semantic 3D City Models and the CityGML standard

Applications for semantic 3D city models should be able to work with datasets of many different cities in order to make their development interesting also from an economic point of view. As stated above, a semantic 3D city model can be used as an integration platform for urban information that many users and systems have shared access to. This requires guaranteeing a high degree of interoperability in order to not being locked-in to one vendor or system.
The international standard CityGML issued by the Open Geospatial Consortium (OGC) (Gröger et al., 2012) defines an open data model and XML based exchange format for semantic 3D city models. CityGML provides common definitions and data specifications for different thematic objects like the digital terrain model (DTM), buildings, bridges, tunnels, roads, railways, water bodies, vegetation, street furniture, etc. All city objects can be represented in up to five levels of details (LOD) simultaneously with their semantics, 3D geometry, 3D topology, and appearance (see Kolbe, 2009). The CityGML ontology provides a standardized interface to the 3D city model and in principle allows running the same applications on the city models of different cities worldwide. CityGML uses object oriented modeling for a high level of expressivity. It supports taxonomical class hierarchies, object aggregation hierarchies, and general associations between object classes. In contrast to other exchange formats like Shapefiles, DXF, Excel tables, or RDF files, CityGML does not require to split up data on the city to a multitude of files (or triplets in case of RDF), but instead can keep everything grouped consistently within one framework. Since CityGML is based on the OGC standard GML, all datasets can immediately be made accessible over the Internet using standard OGC web services like the Web Feature Service (WFS), Catalog Service for the Web (CS/W), and processed using the Web Processing Service (WPS) in an interoperable way.

1.2. Open Data about and from NYC

All data that was being used to generate the 3D city model of NYC originated from the (NYC Open Data Portal, 2015). In March 2012 the legal foundations for the portal were laid out. One year later the first datasets were made accessible. According to (Bloomberg & Merchant, 2012), all relevant and suitable datasets shall be made publicly available by the end of 2018. Currently (June 2015), 1343 datasets are accessible over the portal.
Data are acquired and maintained by different departments of NYC administration. All suitable datasets are made public. The goals of the Open Data strategy are (among others) to improve the reachability and transparency of the municipal administration. The public can use and process the data and can publish the results again on the NYC Open Data Portal. All data are machine readable

and are regularly updated. Additionally, metadata are provided describing the type of data and the way they are collected. The contents are diverse and comprise besides geodata, permissions, and admissions even information on citizen complaints and criminal activities. Where possible original data are linked with data from other departments using unique object IDs (like the Building Identification Number, BIN).
The conditions of using the NYC Open Data Portal are described in the "terms of use" document (NYC Open Data Portal, 2015). Datasets are provided on the Web to inform the public and for application developers. The terms of use explicitly state that no warranties are given on the completeness, up-to-datedness, contents, and suitability of the provided data. Also no responsibilities are taken for data that is published by third parties. The largest shares of the published data are provided by the different departments of NYC administration. Their authors are responsible for the data quality and regular updating. In addition, the "Terms of Use of NYC" apply governing the lawful use, the provision of information, and aspects of intellectual property (NYC.gov, 2015). The *Open Data Policy and Technical Standards Manual* (Bloomberg & Merchant, 2012) stipulates that the respective administration always remains owner of the data and that data remain a public resource. Public use of the data also covers the development of applications. In such cases the developer holds the rights on the application while the ownership of the data remains with the NYC administration. Finally, the *Local Law 11 of 2012* of the City of New York rules the provision, publication, and use of Open Data.



Feature types	Input datasets	Dept.
Addresses	NYC Address Points	DoITT
Buildings	Building Footprints MapPLUTO	DoITT DTM
DTM	1 ft. Digital Elevation Model (DEM)	DEP & DoITT
Lots	MapPLUTO (Bronx, Brooklyn, Manhattan, Queens, Staten Island)	DTM
Parks	Mèmèkas Meadow Parks Properties	DoITT DPR
Steet	LION File Geodatabase	DCP
Trees	Street Tree Census (Staten Island, Brooklyn, Bronx, Queens, Manhattan)	DPR
Waterbodies	Hydrography	DoITT
Waterbody Structures	Hydrography-structures	DoITT
Zoning	Zip Code Boundaries Census Tracts 2010 Borough Boundaries	DoITT DCP DCP

Figure 1: Spatial extent of New York City territory (left) and used datasets from the NYC Open Data Portal (right). The datasets are issued and maintained by the following NYC administrative departments: DEP = Dept. of Environmental Protection, DCP = Dept. of City Planning, DoITT = Dept. of Information Technology & Telecommunications, DPR = Dept. of Parks and Recreation, DTM = Dept. of Finance's Digital Tax Map.

The selection of datasets from the NYC Open Data Portal for the generation of the 3D city model was driven by the city modeling ontology as specified in the CityGML 2.0 standard. This means that it was checked which feature types defined by CityGML can be generated or derived from which NYC datasets. Additionally, those datasets from NYC were considered which could be used to enrich specific feature types. For example, the MAPPLUTO dataset of NYC provides information about land lots and their tax assessments. It also gives information about the most important buildings on the lot like their net floor area, construction year etc. If exactly one building lies within a lot, these information can unambiguously be assigned to the respective CityGML building object. In the end, 26 datasets issued from five different departments of the NYC administration were identified and used. These are listed in Figure 1 above. The bounding box of the entire geographic area covers around 2,300 km², from which the water and land areas belonging to NYC make up ca. 1,080 km². The land surfaces only cover around 783 km².

2. CHALLENGES AND TASKS

The generation of a semantic 3D city model for NYC faces a number of challenges. First of all there are issues concerning data transformation and integration. All spatial datasets on the NYC Open Data Portal have 2D geometries, except for the DTM which is 2.5D. From these geometries 3D geometries must be generated, but the methods differ substantially for different feature types. In some cases new 3D objects have to be created based on the given 2D representation (e.g. volumetric building shapes from footprints, or areal 3D street surfaces from 2D center lines).

The geometries of the source datasets are using different coordinate reference system (CRS). In order to generate an integrated 3D city model with aligned 3D geometries a common CRS should be used. We decided to employ a compound 3D coordinate reference system (2D+1D). It consists of the 2D CRS *'NAD83 New York State Planes, Long Island, Meter [NY83-LI]'* (EPSG code 32118) for planimetry and the 1D height CRS *'NAVD88 height'* (EPSG code 5703). Since the planimetric CRS is a projected, metric system and the vertical reference system is also metric, the compound CRS defines a metric Cartesian system making it easy to compute distances, lengths, areas, and volumes. Length or area values in object attributes of the source datasets are generally given in feet which are also converted to meters.

Another major challenge are the semantic transformations from the separate datasets defined and provided by the different departments of the NYC administration to the common semantic model of CityGML. The difficulties here are to define the correct mappings from the source data models to the one of CityGML. A preliminary investigation showed that many mappings are not just 1:1, but often 1:*n* and sometimes even *n:m*. This means, that *n* objects from the NYC datasets have to be mapped to *m* CityGML objects. Since we intend to enrich all objects of the 3D city model by thematic attributes, suitable and relevant datasets need to be identified first.

Last but not least handling of the huge data volumes must be addressed explicitly. New York City is a very big city both regarding its regional and vertical extent. This means that the methods and tools must be able to cope with large files and a huge number of spatial objects. Geoobjects with large spatial extents may need to be subdivided in order to be able to handle (and use them) efficiently.

3. CONSIDERED OBJECT TYPES

3.1. Digital Terrain Model

The NYC Open Data Portal provides a DTM covering the entire city territory. It consists of a single rectangular raster dataset of 158,100 x 156,100 cells covering an area of around 2,300 km² with a cell resolution of 1x1 ft. The file size is 121 GB making it difficult to work with it when not using a geoinformation system or a geodatabase management system. The DTM is based on data that was acquired in an airborne LiDAR measurement campaign carried out in 2010.

In CityGML terrain models are represented by *ReliefFeature* objects. These can be composed of *ReliefComponents* of different subtypes like *TINRelief*, *MassPointRelief*, *BreakLineRelief*, or *RasterRelief*. The component schema allows on the one side representing the whole DTM by a

number of smaller components (e.g. tiles). On the other side it allows to combine different subtypes for the same area (e.g. a *RasterRelief* component with a *BreakLineRelief* component).
Since we intend to integrate street and water surfaces with the DTM we decided to perform a triangulation of the raster based DTM according to a regular tiling schema, which creates CityGML *TINRelief* components that can easily be handled by users and applications. With a tile size of 250m x 250m in total 35,153 tiles were created. In order to ensure that the height profiles of neighboring tiles match along their boundaries, the height profiles of the bounding boxes of all tiles were computed in the form of 3D lines. These 3D lines were then used as break lines in the triangulation of each tile. This is illustrated in Figure 2. As can be seen in the right image the height values of neighbored tiles do perfectly match along their boundary (yellow dashed line).

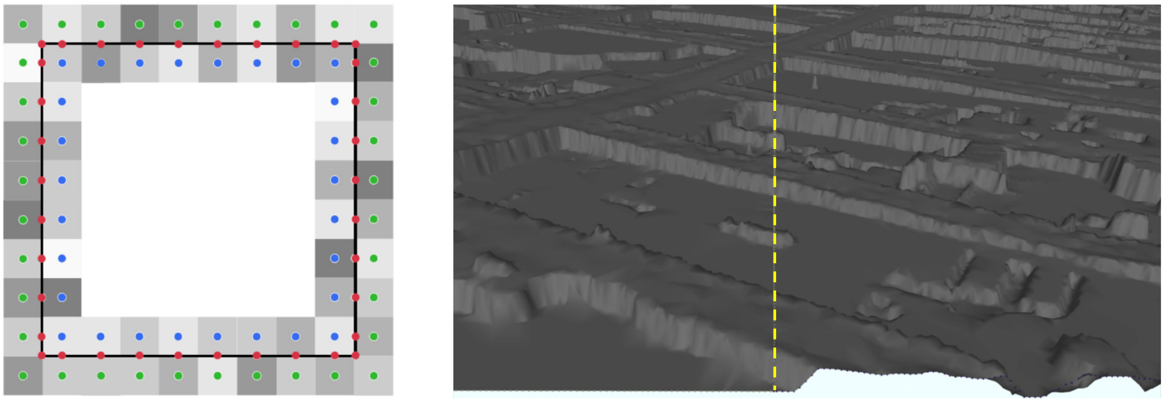


Figure 2: Left: DTM height points considered in computing the boundary height profile for each TIN tile. Blue points belong to height values from cells within the tile and green points lie outside. From the blue and green points the heights of the red points are interpolated. The tile boundaries (black lines) are used as break lines in the triangulation. Right: visualization of two adjacent TIN tiles (the boundary is indicated by the yellow dashed line).

3.2. Buildings

Building objects are generated from a building footprint dataset. The footprints are first elevated to a base height by projecting each 2D polygon onto the DTM and selecting the height value of the lowest polygon point. This ensures that all wall surfaces are completely grounded on the terrain. Then the footprint polygons are extruded upwards in vertical direction according to a measured height value coming with each polygon, effectively creating an LOD1 3D solid geometry. The original building footprint dataset contains 1,082,005 polygons having 15 attributes each. Garages and sheds are also represented by their footprints. Some of the building attributes like the *building name, usage, function,* and *measured height* are mapped to the respective predefined attributes of the CityGML building model. The others like *building identification number (BIN)*, *borough block lot number (BBL)* are represented by generic attributes of the CityGML building objects. For each building the volume of the 3D solid geometry is computed and added as an attribute which enables simple subsequent queries and computations without the further need of 3D geometric operations.
The NYC Open Data Portal also contains the so-called PLUTO dataset from the Department of Finance which provides data on the land lots. As was stated before it also gives information about the most important buildings on the lot like their *net floor area*, *number of floors*, *construction year, renovation year* etc. The problem is that it is not explicitly stored which are the most important buildings on the lot. Therefore, we first check whether exactly one building lies within a lot by intersecting the building footprint polygons with the land lot polygons. For these cases, about 35 further attributes about the building are taken from the PLUTO land lot polygons and added to the respective CityGML building objects.

The building addresses are provided by a different source dataset where each address is located by a 2D point lying within the 2D footprint polygon of the respective building. Each address also references the building by the BIN. We linked address and building data by using the BIN. Please note that buildings generally can have more than one address. For each address a CityGML Address object is being generated and is stored within the respective CityGML building object giving a total number of 2,020,523 objects.

Each CityGML building object is further enriched by a number of *external references*. Each consists of the URL of the source dataset and the original ID value within that dataset. Also the so-called *Terrain Intersection Curve (TIC)* is computed and added as an additional 3D ring geometry denoting the intersection of the 3D building solid geometry with the DTM. Figure 3 shows a visualization of some buildings together with the DTM.

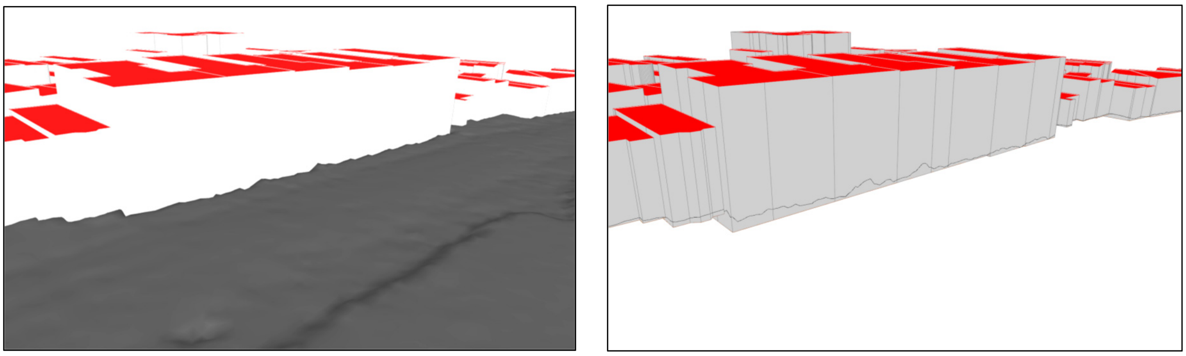


Figure 3: Left: generated 3D building objects shown together with the DTM. Right: terrain intersection curves (TIC) are 3D rings around the building solids denoting the intersection line with the terrain.

3.3. Streets and Roads

The NYC Open Data Portal provides street geodata within the so-called LION dataset of the Department of City Planning. It consists of street segments which are geometrically represented by 2D center lines. Furthermore, it contains 2D point objects representing street crossings. The two datasets establish a geometric-topological network, i.e. a graph where street line segments represent the arcs and street crossing points the nodes. Both the street center lines and crossing points come with a number of thematic attributes like *street name, traffic direction*, *height level,* and *priority for snow removal*.

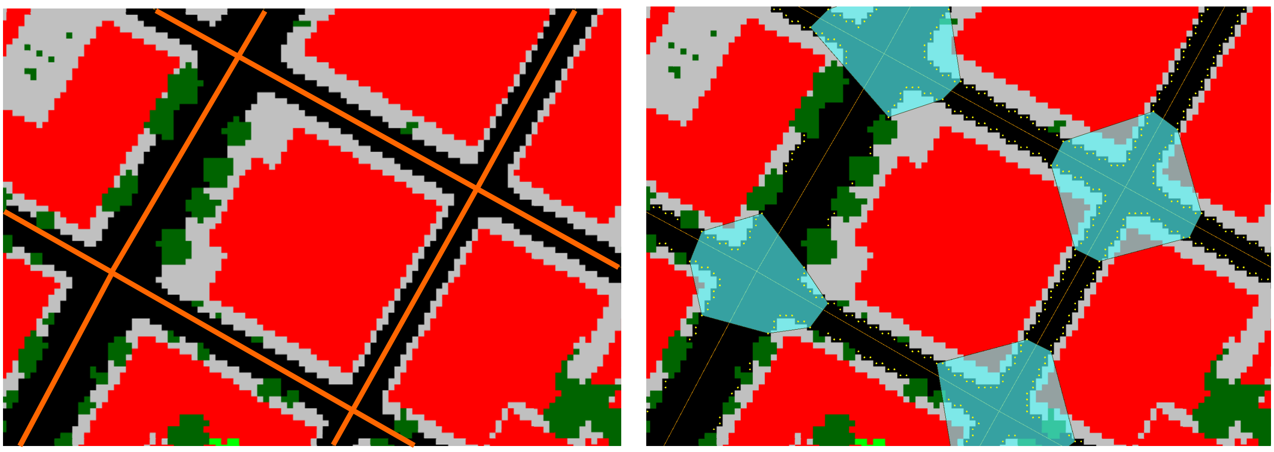


Figure 4: Road network (in orange colour) overlayed onto the land cover classification raster dataset. Red areas denote buildings, black areas streets, green areas vegetation, and grey areas pedwalks or other paved areas. The right image shows the automatically determined street intersection areas (light blue) and street boundary points (yellow dots).

In CityGML streets and roads are modeled by the feature type *Road*. In LOD0 their geometry is given by a geometric complex consisting of connected 3D line segments and points. Transformation of the source street data is straight forward in this case; the 2D geometric network just has to be mapped onto the DTM in order to generate the required 3D geometries. In LOD1, however, streets are represented by the so-called *TrafficArea* objects which basically represent the street surfaces. Since streets are geometrically only given by their center lines in the NYC Open Data Portal, we had to develop a method to generate the areal street geometries. We want the 3D model to be as realistic as possible. Thus, we do not want to use a buffering of the center lines by some default street widths, but intend to determine the widths of all street segments individually. For this purpose we used the land cover classification map that is also available as Open Data. We used a map with a resolution of 3 feet cell length and a file size of 200MB. It distinguishes between a) tree crowns, b) other vegetation, c) bare soil, d) water, e) buildings, f) streets, and g) other paved areas. Figure 4 shows on the left side the street network overlaid onto the land cover map.
The employed method creates sample points along the center line in regular steps. Then for each sample point the orthogonal distances from the centerline to the last cell of the land cover map that is classified as street area are determined for the left and the right side. Since the street widths are generally varying around street crossings, distances from these areas are filtered out (see blue areas in Figure 4 on the right). At some places the view onto the street surfaces is occluded by trees giving wrong distance values. However, we can observe that for the majority of the samples the correct distance is given. Therefore, a histogram analysis is performed on the sampled distances and the distance value class with the highest number of occurrences is considered to represent the street width (multiplied by 2). The center line then is buffered according to the width of the street segment. This is illustrated in Figure 5 below. Polygons overlap at intersections giving a connected surface for each street. An assessment of the achieved accuracy will remain as future work.

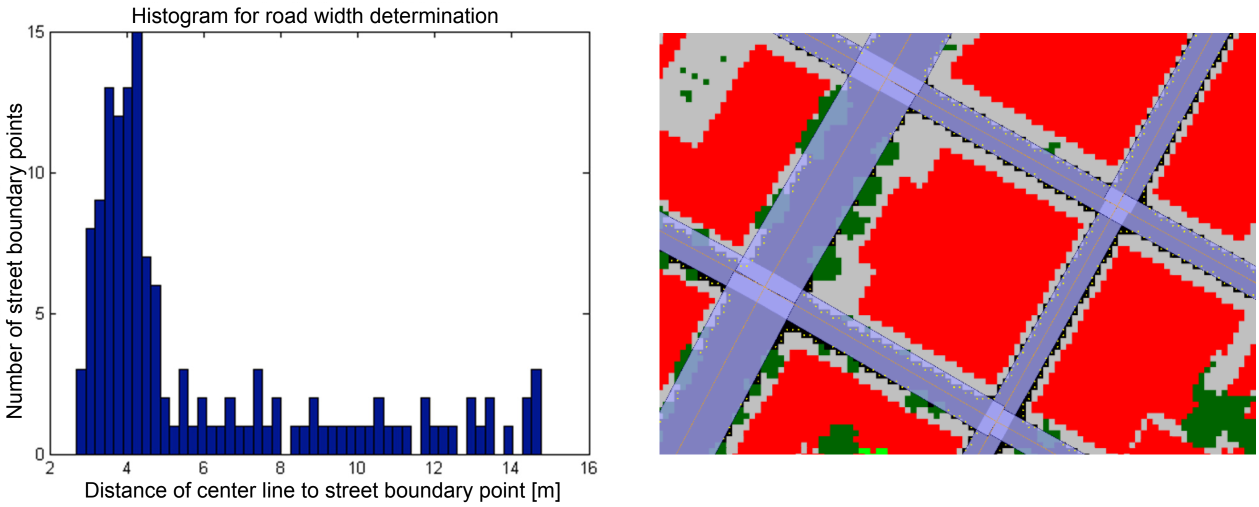


Figure 5: Left: Histogram of computed distances from sample points along a street center line to the last land cover cell classified as 'street' in transversal direction. Right: 2D street surface polygons (in blue) created from buffering the center lines with the estimated street width.

In a final step the 2D street segment surfaces must be vertically aligned with the DTM. This is not trivial in general, because due to previous height filtering and removal of objects like buildings, trees, and cars the DTM contains "bump artifacts" on the street surfaces which are not there in reality. Paved street surfaces are typically smooth with no or little transversal slopes. Above, the DTM may be inaccurate near to buildings due to occlusions which may cause height gradients near to the side limits of the streets. This topic has been discussed much in the past and different strategies were suggested to tackle the smoothing of street polygons and their integration with the DTM (Koch & Heipke, 2006), (Schilling et al., 2007), (Oude Elberink & Vosselman, 2006). Due to the limited project duration and

unforeseen difficulties in the general handling of the huge data volumes we finally implemented a simple strategy here, where the street polygons were triangulated according to the DTM with the polygon boundaries used as breaklines.
Anyway, in NYC there are also many elevated roads, complex motorway junctions, bridges, and tunnels which have to be treated in a special way. All street segments have two attributes indicating the qualitative height level at the start and the end of the segment respectively. The level information is given by one of 17 different letters which means that 17 different height levels relative to the ground are distinguished. Since no quantitative height information is given, we assumed a vertical distance of 4m between two consecutive levels. This allows lifting or lowering the 3D geometries of the streets (both the center line and the street surface) by the level difference to ground multiplied with 4m. The heights of segments having different start and end levels are linearly interpolated between the two height levels. Examples are shown in Figure 6.

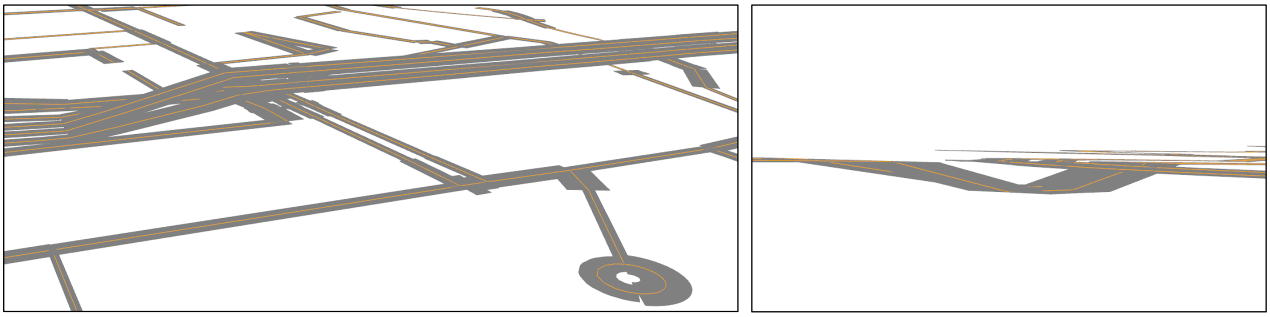


Figure 6: Road network and road surfaces mapped to the DTM. Local heights are variated for bridges and elevated roads (left) and tunnels or underpasses (right). Due to the lifting or lowering of road heights, road center lines or road surfaces that do not intersect in reality do not intersect spatially in the resulting CityGML 3D dataset either.

The lifting and lowering of the road segments also guarantees that a topologically correct 3D road network is being generated. Hence, two street segments only are considered connected, if they are incident to the same 3D intersection point (on the same height level). This allows running routing or tour planning applications on the 3D network that, in addition, can also consider the street slopes. Figure 7 shows an example of a complex motorway junction with street segments on many different height levels.

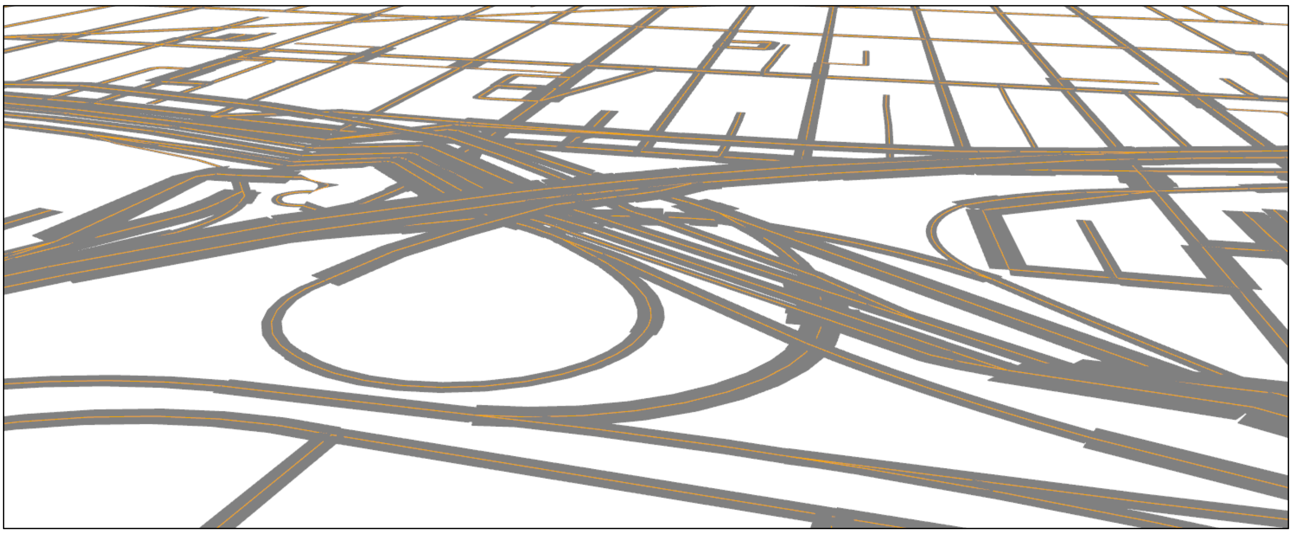


Figure 7: Road network and road surfaces of a complex motorway junction mapped to the DTM. Local heights are variated for bridges and elevated roads according to their different height levels. For road sections linking different height levels the height values are linearly interpolated along the centerline.

3.4. Other Feature Types

Besides the DTM, buildings, and streets further feature types were generated. Due to limited space, we can describe them only shortly in the following. Detailed explanations are given in (Burger, 2015) and (Cantzler, 2015).

From the NYC MAPPLUTO dataset of the Department of Finance, the land lots – geometrically represented by 2D polygons – were transformed to 866,853 CityGML *LandUse* objects. Each object has 75 thematic attributes including land ownership information together with tax assessment information.

NYC Parks were also transformed to CityGML *LandUse* objects. The 2D spatial extents were mapped onto the DTM resulting in triangulated 3D surface geometries. Each of the 16,159 objects has 10 thematic attributes.

The datasets about street trees contains 623,920 entries, but it turned out that the same trees were contained multiple times and that the 2D center points are typically lying in the middle of the streets and not on either street side. Nevertheless, a 3D tree model was created in the shape of a 'lollipop' at the location of each tree. All trees are represented by 277,108 CityGML *SolitaryVegetation* objects with 16 thematic attributes each.

Water bodies are provided in the NYC Open Data Portal by attributed 2D polygons. Since rivers are often represented by just one polygon, they were segmented into smaller extents. The 2D polygons were then mapped onto the DTM generating 9,542 CityGML 3D *WaterSurface* objects. Due to the limited project time their geometries were not corrected or filtered any further as described in other articles, see (Koch & Heipke, 2006).

4. IMPLEMENTATION AND RESULTS

All of the described transformation processes were implemented as workspaces for the spatial Extract, Transform, and Load (ETL) tool *Feature Manipulation Engine* (FME Oracle Edition 2014 SP4), see (Safe Software, 2014). For the storage, intermediate processing, and the management of the 3D city model the Open Source 3D geodatabase *3DCityDB* (Version 3.0) was employed (3DCityDB, 2015; Stadler et al., 2009). 3DCityDB was installed on top of the spatial relational database management system Oracle Spatial 12c Enterprise Edition (64Bit). A workstation running Windows 7 Enterprise (64Bit) on two 2.4GHz Xeon E5-2609 Quad Core CPUs with 32GB RAM, 512GB SSD system drive, and 8TB hard disks was used as the database server machine. Similar machines were used to perform the spatial ETL processes. The workstations and the DB server were attached to the same 1GBit Ethernet LAN. Figure 8 shows the principle data flows.

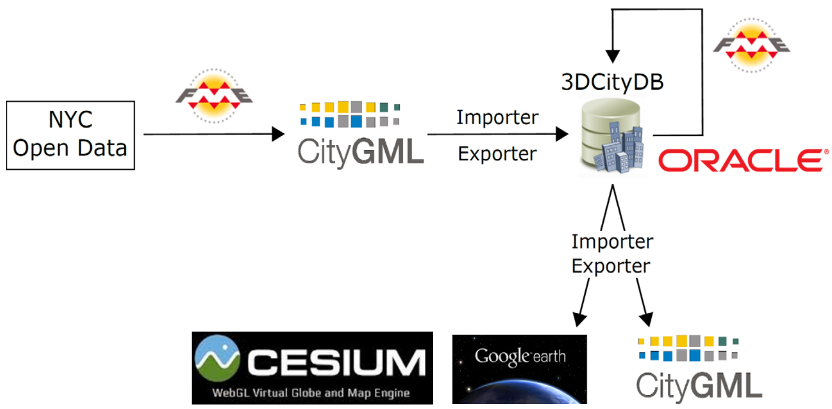


Figure 8: Workflow and employed tools. The source data from the NYC Open Data Portal is integrated and transformed to CityGML files using FME and imported into the 3DCityDB. Then the 2D-3D transformations are carried out in the geodatabase. The final results are exported to CityGML files for further applications and to KML files for visualization.

In a first step the NYC datasets were processed, integrated, brought into the same coordinate reference system, and transformed to CityGML (all with base height 0). In the 2nd step the resulting CityGML files were imported into a 3DCityDB account. In order to integrate the features with the DTM and to cope with the large data volumes and the huge number of objects, an ETL master process was defined that performs its sub processes on tiles of the stored geodata only. The data inside each tile were then transformed to 3D, enriched by further attributes, and reimported to the 3DCityDB. In general all CityGML objects are enriched by *external references* pointing with their URLs to the download addresses of the original datasets and to the object IDs within the respective datasets. These links allow tracking back from city model objects to the original source data which will be useful in the case that some objects are being updated within or deleted from the 3D city model.
The following table gives an overview on the number of objects from the source datasets and the resulting CityGML model, together with the number of included thematic attributes for the latter. Also the file sizes are given.

Dataset	Format	Geometry types	Number of objects	Num of attributes	Data size [GB]
Buildings/ Addresses	Shape	2D polygon/point	2,023,531		0.931
	CityGML	3D Solid	2,020,523	20 – 55	11.085
DTM	Raster	Grid	1		121
	CityGML	Tiled TIN	35,153 tiles	–	1,450
Land Cover	Raster	Grid	1		0.2
	CityGML	–	–	–	–
Lots	Shape	2D Polygon	857,853		0.867
	CityGML	3D Polygon	866,853	75	8.021
Parks	Shape	2D Polygon	14,674		0.025
	CityGML	3D Polygon	16,159	10	0.054
Streets	File Geodatabase	2D Line	212,890		0.128
	CityGML	3D Line+Polygon	149,292	31	0.482
Street Inter-sections	File Geodatabase	2D Point	125,118		0.128
	CityGML	3D Point	104,754	1	0.055
Trees	Shape	2D Point	623,920		0.206
	CityGML	3D tree shape solid	277,108	16	113
Water Bodies	Shape	2D Polygon	1,976		0.01
	CityGML	3D Polygon	9,542	5	0.025
Water Body Structures	Shape	2D Polygon	2,464		0.003
	CityGML	3D Polygon	2,464	3	0.006
Zoning	Shape	2D Polygon	2,436		0.005
	CityGML	CityObjectGroups	2,436	23	≤ 1
Total	Original	2D + 2.5D	3,864,864		123.4
	CityGML	3D + 2.5D	3,484,284		1,583.7

Table 1: Overview of the generated CityGML feature types and the respective numbers of objects. CityGML file sizes are given for the uncompressed XML files. The largest share is taken by the DTM (1.45 TB) caused by the GML data for >5 billion TIN triangles. File compression effectively reduces CityGML files to about 5% of their original sizes.

The total processing time was rather long with high variations regarding the different feature types. It took 17 days to generate all CityGML features and another 21 days to import everything into the 3DCityDB. However, most of the time was required to handle the DTM (9 days to generate the tiles and 20 days to import the resulting 1.5TB CityGML dataset). The 3D generation and data integration of the 1 million buildings in LOD1 in CityGML took less than 10 hours and the import less than 3

hours. The creation of all CityGML Road objects required 23 hours and the import to 3DCityDB just 13 minutes. Please note that no tuning of the Oracle DB installation was performed and also the FME transformation workspaces – while running in parallel on multiple workstations – are not optimized yet regarding execution speed. A critical bottleneck we identified was that the workstation used as the database server did not scale well due to its normal hard disk when multiple users or processes accessed the database concurrently. Furthermore, for this large number of geometry objects, partitioned spatial indexes should be used in the database.
Regarding the resulting datasets we can confirm that all objects belonging to the selected feature types have been transformed to 3D objects in CityGML. The LOD1 building models, the triangulated DTM tiles, the LOD1 park areas, and the LOD1 land parcels can be considered to have final quality. For the street objects the LOD0 geometries (3D network) are fine, but the LOD1 road surfaces will need further treatment and refinement in the future. Also bridge and tunnel objects should be created. Water bodies need to be filtered to have consistent Z values for all points of the same 3D water surface. While the process to generate 3D tree geometries is working, there are problems with respect to the data quality of the input data. Most trees are not located at their correct positions but often stand in the middle of the roads. This problem will have to be resolved in order to create correct tree objects.
Finally, the resulting CityGML model was exported using the 3DCityDB tools in CityGML format and as a tiled KML visualization model. The latter can be visualized and explored e.g. with *Google Earth* or the Open Source *WebGL Virtual Globe Cesium* (see Figure 9 below).

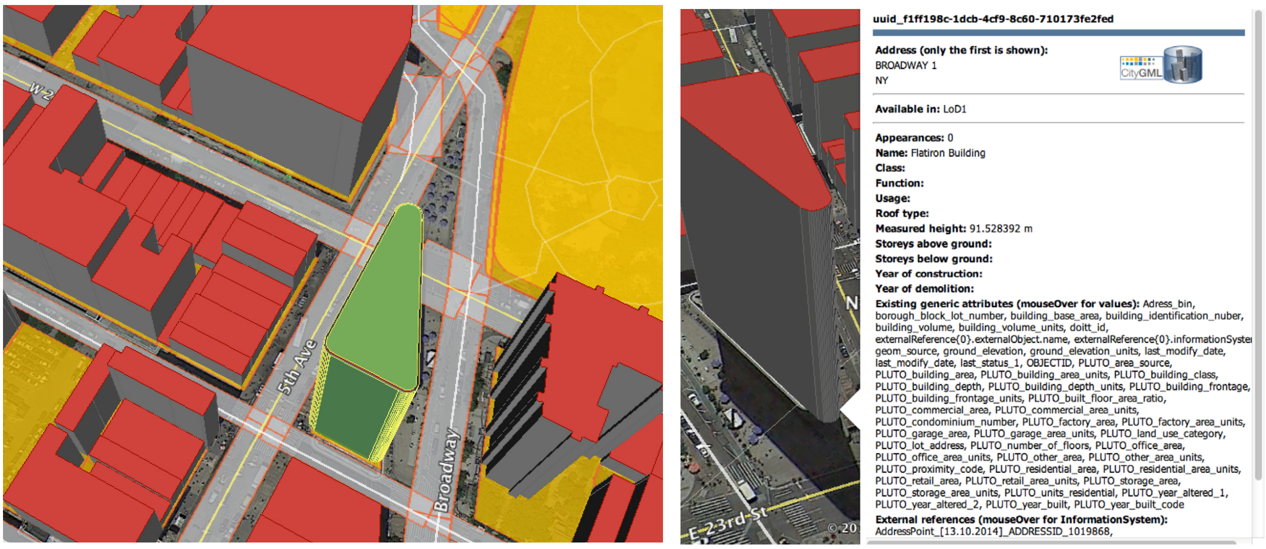


Figure 9: Screenshot showing the 3D city model of Manhatten where 5th Avenue meets Broadway. Left image: LOD1 buildings, LOD1 street surfaces (light grey), land parcels (yellow). Right image: list of attributes attached to the CityGML building model of the famous 'Flatiron Building'. These can be queried and used for further analyses.

5. CONCLUSIONS AND OUTLOOK

The generated 3D city model is complete in the sense that it covers the entire area of New York City. The required work on the management of the huge data volume took much more time than was expected before. Therefore, not all project goals were achieved to the same degree. However, the presented approach in general seems feasible to generate (and update) a 3D city model in LOD0 and LOD1 from the public 2D and 2.5D datasets using a fully automatic transformation process.
The CityGML dataset, all FME workbenches and the two master theses (of Barbara Burger and Berit Cantzler) will be made freely available via our homepage (TUM Geoinformatics, 2015). Now, different kinds of applications can be tested with the NYC CityGML dataset. We intend to perform a

solar potential analysis for entire NYC and to further enrich the building objects by their respective solar irradiation values resolved by weekly or monthly estimations. It would also be interesting to investigate the computation of city indicator values directly from the semantic 3D city model. City administrations and private stakeholders like real estate firms and banks are interested in monitoring and comparing the performance of specific aspects of cities using city indicators like the ones specified in the recently released ISO standard 37120 for a “holistic and integrated approach to sustainable development and resilience” (ISO 37120, 2014).

We hope that by providing a semantic 3D city model of NYC as an open dataset we can increase the interest of users, academics, and stakeholders in CityGML in the US on the one hand. We provide the transformation tools as well as the 3D geodatabase software as Open Source software. Hence, on the other hand we hope that (some of) these might be adopted by New York City administration to generate updated 3D city models in the future. In any case, the resulting huge semantic 3D city model can serve as a CityGML reference dataset to test, benchmark, and compare performance of 3D GIS, 3D geodatabases, and 3D visualizations.

There are similar Open Data stores for other US cities. A first inspection showed that in principle the available data for San Francisco, Chicago, and Philadelphia would provide enough information to generate analogous 3D city models to NYC. While the formats that are being used there are almost the same as in NYC (ESRI File Geodatabases and Shapefiles, Excel tables, Autodesk DXF files, raster based DTMs), the structuring of these datasets and their feature types, attributes, and relationships differ to a wide extent. Hence, no semantic interoperability is given, which would be required to run the same 3D city model applications on these different cities. The integration of the different datasets and their homogenizations into the common semantic data model of CityGML would resolve this problem. The tools that have been created in the course of the project and described in this paper can rather easily be adapted to the local specialties of the Open Data of the aforementioned cities.

The 2D-3D transformation process could profit from procedural modeling. However, to the best of our knowledge no commercial or free software is able to create besides 3D geometries and graphical materials also semantic information yet. Nevertheless, the work presented by (Biljecki et al. 2014) goes in this direction and should be further investigated.

6. REFERENCES

All web resources were last accessed in June 2015.

3DCityDB (2015): 3D City Database, Version 3.0. Open Source Geodatabase Software for the Management and Visualization of Semantic 3D City Models. http://www.3dcitydb.org

Agugiaro, G., Hauer, S. & Nadler, F. (2015): Coupling of CityGML-based Semantic City Models with Energy Simulation Tools: some Experiences. In: Schrenk, M., Popovich, V. V., Zeile, P., Elisei, P. & Beyer, C. (Eds.): Proc. of REAL CORP 2015, 5-7 May 2015, Ghent, Belgium.

Biljecki, F., Ledoux, H. & Stoter, J. (2014): Error propagation in the computation of volumes in 3D city models with the Monte Carlo method. In: Proc. of the Joint Int. Conference on Geospatial Theory, Processing, Modelling and Applications, Oct. 2014 in Toronto, Canada, ISPRS Annals.

Bloomberg, M. R. & Merchant, R. N. (2012): Open Data Policy and Technical Standards Manual. NYC Department of Information Technology & Telecommunications, final 1.2.1 edition, http://www.nyc.gov/html/doitt/downloads/pdf/nyc_open_data_tsm.pdf

Burger, B. (2015): Ein semantisches 3D-Stadtmodell der Stadt New York auf Basis von Open Data – DGM, Straßen und Zonierung. Master Thesis, Chair of Geoinformatics, Technische Universität München (in preparation).

Cantzler, B. (2015): Ein semantisches 3D-Stadtmodell der Stadt New York auf Basis von Open Data – DGM, Gebäude, Flurstücke, Gewässer, Parks und Bäume. Master Thesis, Chair of Geoinformatics, Technische Universität München (in preparation).

Gröger, G., Kolbe, T. H., Nagel, C. & Häfele, K.-H. (2012): OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0. OGC Doc. No. 12-019, Open Geospatial Consortium. http://www.opengeospatial.org/standards/citygml.

ISO 37120 (2014): Sustainable development of communities – Indicators for city services and quality of life. International Standard, International Organization for Standardization.

Kaden, R. & Kolbe, T. H. (2013): City-wide Total Energy Demand Estimation of Buildings using Semantic 3D City Models and Statistical Data. In: Proc. of the 8th Int. Conference on 3D Geo-Information 2013 in Istanbul. ISPRS Annals, Vol. II-2/W1, pp. 163-171.

Koch, A. & Heipke, C. (2006): Semantically correct 2.5D GIS data – The integration of a DTM and topographic vector data. ISPRS Journal of Photogrammetry and Remote Sensing, Vol 61, No 1.

Kolbe, T. H. (2009): Representing and Exchanging 3D City Models with CityGML. In: Lee, J. & Zlatanova, S. (Eds.): 3D Geo-Information Sciences, Springer LNG&C Series, pp. 15-31.

NYC.gov (2015): Terms of Use. http://www1.nyc.gov/home/terms-of-use.page.

NYC Open Data Portal (2015): https://nycopendata.socrata.com.

Nouvel, R., Schulte, C., Eicker, U., Pietruschka, D. & Coors, V. (2013): CityGML-based 3D City Model for Energy Diagnostics and Urban Energy Policy Support. In: Proceedings of BS2013, 13th Conf. of International Building Performance Simulation Association, Chambéry, France, August 26-28, 2013.

Oude Elberink, S. & Vosselman, G. (2006): 3D Modelling of Topographic Objects by Fusing 2D Maps and Lidar Data. In: Proceedings of the ISPRS TC-IV Int'l Symposium on Geospatial databases for sustainable development, pp. 199-204.

Safe Software (2014): The Feature Manipulation Engine (FME) Version 2014 SP4. An Extract, Transform, Load (ETL) software tool for geospatial datasets. http://www.safe.com.

Schilling, A., Basanow, J. & Zipf, A. (2007): Vector based Mapping of Polygons on Irregular Terrain Meshes for Web 3D Map Services. In: Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST) in Barcelona, Spain.

Stadler, A., Nagel, C., König, G., Kolbe, T. H. (2009): Making interoperability persistent: A 3D geo database based on CityGML. In: Lee, J. & Zlatanova, S. (Eds.): 3D Geo-Information Sciences, Springer LNG&C Series.

TUM Geoinformatics (2015): Homepage of the Chair of Geoinformatics, Technische Universität München. http://www.gis.bgu.tum.de.