Towards Defining a Framework for the Automatic Derivation of 3D CityGML Models from Volunteered Geographic Information

Marcus Goetz, Alexander Zipf
Chair of GIScience, Department of Geography, University of Heidelberg, Germany

ABSTRACT

High-quality geographic data sources are eminent for urban data management and the creation of detailed 3D city models. In the last two decades, Volunteered Geographic Information (VGI) increasingly gained attractiveness to both amateur users and professionals, resulting in a broad availability of urban data within VGI communities and especially OpenStreetMap (OSM). OSM provides detailed information about urban regions and especially more and more buildings are also mapped. Existing 3D-VGI applications e.g. KOSMOS Worldflier (Brejc, 2011) or the OSM-3D project (OSM-3D, 2011) only focus on visualization purposes, but a standardized usage for exchanging and sharing urban city models is not combined with VGI. Therefore, this paper presents a framework for an automatic VGI-based creation of 3D building models encoded as standardized CityGML models. That is, the usage of VGI as a proper data source for the creation of standardized city models will be proven.

Keywords: 3D city models, Computer Science, CityGML, Information Systems, OpenStreetMap, Urban Data, Volunteered Geographic Information

INTRODUCTION

Three-dimensional urban city models are used by the economy and public administration for different purposes, e.g. environmental simulations or facility management (Kolbe, 2009). Thereby, the field of application evolved from traditional applications such as network planning, typically requiring pure geometric models with low level-of-detail, to advanced applications in areas such as tourism. That is, the requirements of city models heavily increased, meaning that besides geometric information there is also a strong need for semantic information. However, the creation and maintenance of such detailed models is very expensive (Benner et al., 2005), because it is largely done manually and automatic procedures are rare, while semi-automated approaches are becoming more and more popular.

The City Geography Markup Language (CityGML) became the international standard for storing, visualizing and exchanging three-dimensional urban city models, thus allows an interoperable access to 3D city models (Kolbe et al., 2005). CityGML models do not only contain geometric information, but also a variety of topologic and semantic information, e.g. names, building types or addresses. The creation of CityGML models typically requires high-quality data, which is usually captured and provided by professional surveyors and cartographers, public authorities or commercial data providers. Existing standards such as Building Information Modeling (BIM) or Industry Foundation Classes (IFC) have also been shown to be transformable to CityGML (Benner, et al., 2005). Nevertheless, a large percentage of CityGML models are created manually by exporting the models from different CAD and 3D graphic applications (e.g. Google Sketchup).

In the last couple of years, the term Volunteered Geographic Information (VGI) became popular, whereat VGI describes that an ever expanding range of users collaboratively collects geographic data (Goodchild, 2007a). That is, hobbyists create geographic data based on personal measurements (via GPS etc.) and share those in a Web 2.0 community, resulting in a comprehensive data source of humans acting as remote sensors (Goodchild, 2007b). Especially in urban regions the coverage of VGI data is very good, leading to an increase of the usage of VGI in urban data management (Song & Sun, 2010).

Nevertheless, the data in VGI communities is mostly used for creating two-dimensional maps (e.g. OSM (2011a)). However, one step towards the usage of VGI in a 3D platform has been demonstrated by Schilling et al. (2009). This example also shows the potential of VGI for visualizing urban regions with 3D city models, but only focuses on the visualization of the geometry and not on semantics. Therefore, the main contribution of this paper is the development and suggestion of a framework for the automatic creation of CityGML models by purely using crowdsourced geographic information from OpenStreetMap (OSM). With such a framework, it shall be evaluated and demonstrated that VGI is capable for the creation of standardized city models which can be exchanged via Open Geospatial Consortium (OGC) standards (e.g. Web Feature Service, WFS) and utilized in professional applications and analyses e.g. the mapping of environmental noise pollution (Czerwinski et. al., 2006), urban planning, city business development, tourism (Döllner et. al., 2006), homeland security (Lapierre & Cote, 2007), disaster management (Kolbe et. al., 2008) or indoor navigation (Mäs et. al. 2006).

The remainder of this paper is organized as follows: First, the CityGML standard is described in the detail required for the subsequent work and discussion. This is followed by an introduction to OSM, providing the basics for understanding the conducted research. Afterwards, there is an overview about related work regarding 3D city model creation as well as (3D)-VGI. Thereafter, a framework for the creation of CityGML models from VGI is introduced. The last chapter summarizes the presented work and discusses future research.

INTEROPERABLE ACCESS TO 3D CITY MODELS

A model for the semantic and geometric description of urban regions is the City Geography Markup Language (CityGML) (Gröger et al., 2008; Kolbe, et al., 2005). CityGML became a global standard for storing and exchanging three dimensional city models, thus allows an interoperable access to 3D city models. It is based on the Geography Markup Language 3 (GML3) (Lake et al., 2004), which is commonly used for exchanging data in spatial data infrastructures (SDI) (cf. Zipf. et. al. (2007)) and web environments. Additionally GML3 is the native data format of the OGC WFS (Kolbe, 2009). CityGML does not only cover geometric aspects which are relevant for visualization, but also topologic and semantic information about urban regions such as labels or operation hours. The model “distinguishes between buildings and other man-made artifacts, vegetation objects, waterbodies, and transportation facilities like streets and railways” (Kolbe et. al., 2005, pp. 884).

For providing several differentially detailed city models, CityGML defines specific Level-of-Details (LoD), which vary with regard to the information resolution and generalization. The LoD concept comprises five classes: LoD0 for a 2.5 dimensional Digital Terrain Model (DTM), LoD1 for visualizing building as coarse building blocks, LoD2 for displaying roof structures and façade textures, LoD3 for denoting building details such as windows or doors and LoD4 for models with interior features.

Regarding the CityGML schema, one of the most detailed concepts of CityGML is the building model (Gröger, et al., 2008), which allows for the representation of thematic, spatial and semantic aspects of buildings and building parts. The class *_AbstractBuilding* describes the central class of the model, whereby entities of this class are either a *Building* or *BuildingPart*. Since an entity of *BuildingPart* is again a *_AbstractBuilding*, it is possible to aggregate a hierarchy with arbitrary depth (Gröger, et al., 2008). *_AbstractBuilding* is a subclass of *_CityObject* and therefore it additionally inherits all properties from *_CityObject* (e.g. *gml:name*, *address* etc.). Furthermore *_AbstractBuilding* contains specific building information. These are on the one hand semantic information such as *function*, *roofType* etc., and on the other hand quantitative or metric information e.g. *measuredHeight*, *storeysAboveGround* etc. The spatial representation of building features is given by geometric objects, i.e. *Geometries*, *MultiSurfaces* or *Solids*. A *Building* or *BuildingPart* is bounded by a *BoundarySurface*, which can be a *RoofSurface*, *WallSurface*, *GroundSurface* or *ClosureSurface*, whereby these *BoundarySurfaces* can additionally have *Openings*.

The given introduction ought to be enough for understanding the general concept of buildings in CityGML. For further information as well as UML-diagrams of the different CityGML features, please refer to Gröger et al. (2008).

OPENSTREETMAP: ONE OF THE MOST POPULAR EXAMPLES OF VGI

During the last couple of years, diverse VGI communities such as Wikimapia, Geonames, FixMyStreet etc. have been initiated, but somehow OpenStreetMap is the most popular example for VGI. OSM is a collaborative community which aims for the provision of free map data, which can be used and edited by the community at no charge. With currently more than 400,000 registered users (OSM, 2011b), OSM grew rapidly regarding the amount of data, leading to more than 1,140,000,000 geo-tagged points. What began as a free online world map, evolved very quickly to a huge source of diverse data about urban and rural areas. That is, OSM not only contains information about streets or land areas, but also different semantic information, e.g. about the surface of a road or speed limits. That is, the diversity of information is beyond an ordinary map. For adding information, OSM applies a concept of free-definable key-value pairs, so that a user can add various attributes to different geo-tagged locations. There are no strict rules for the key-value pairs, but there are diverse guidelines and best-practices available, e.g. for defining a street, there is the key *highway* with different values e.g. *residential, motorway* etc. A list of the most commonly used keys is provided by Tagwatch (2011a), as well as a list of all currently used keys by Tagwatch (2011b).

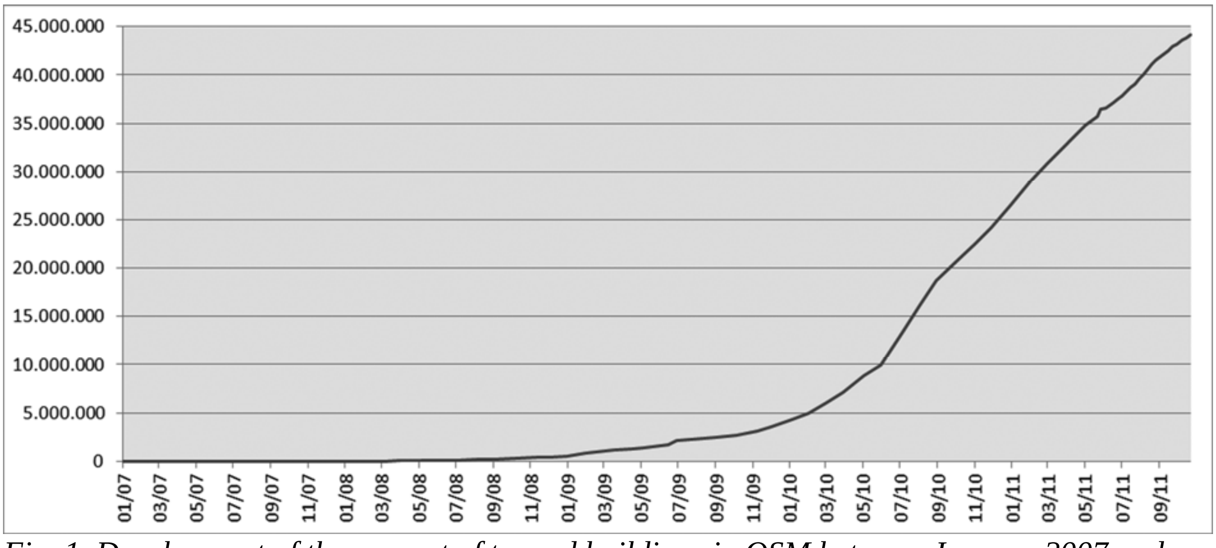

Fig. 1. Development of the amount of tagged buildings in OSM between January 2007 and October 2011. The values are derived from our internal OSM database (updated daily).

From a global perspective, it is evident that different regions differ regarding data quantity and quality. Nevertheless, it has been demonstrated that, especially in urban areas, OSM is able to compete against commercial or official data sources (Haklay, 2010; Mooney et al., 2010; Neis et al., 2010; Zielstra & Zipf, 2010).

Meanwhile, OSM users also started to map buildings. Therefore, users utilize closed ways for describing the ground shape of the building. A way comprises a set of points connected pair-wise with each other, and a closed way is a way whereat the first point equals the last point. Complex shaped polygons such as buildings with an atrium (i.e. a hole inside the ground shape polygon, cf. Fig. 5 (a)) can be mapped with relations, That is, the users map the outer bounding shape of the building with one closed way and additionally, they can provide multiple closed ways which represent the holes in the polygon (the inner polygons cut out holes of the building polygon). For tagging the shape as a building, users simply have to add the key-value *building = yes*. Currently there are more than 44.1 million tagged buildings in OSM (most of them are in Europe) and the amount steadily increases. The number of buildings between January 2007 and October 2011 is depicted in Fig. 1. In average, currently over 375,000 new building outlines are added to OSM every week and it is likely that this trend will increase even further, due to the large availability of high resolution imagery that has been provided as source for mapping for OSM (such as Microsoft Bing Maps in December 2010, cf. OSM (2011c)). In contrast, the amount of streets, which are currently the major part of OSM, comprises 45.2 million instances and an per-week increase of about 200,000 (based on our internal database). Thus, it is likely that soon there will be more buildings in OSM than any other kind of spatial feature.

For enriching buildings with information, there are different keys promoted in the community e.g. *height*, *building:buildyear* etc., which can be utilized for describing the appearance and semantic characteristics. Also, address information can be added with different keys e.g. *addr:street* or *addr:housenumber*. All relevant building keys will be further discussed later in this paper.

RELATED WORK

The problem of deriving, representing and visualizing three-dimensional building models which are close to reality (regarding both geometry and appearance) has been discussed for a while. Thereby, different approaches with different data sources were made. Research started with 2D image processing, but did soon turn towards 3D approaches (Henricsson et al., 1996; Lang & Förster, 1996). Also, the extraction of 3D building models from laser altimetry data is well researched (Maas & Vosselman, 1999; Weidner & Förster, 1995). Furthermore, the utilization of shape grammars for modeling urban areas was explored by different researches, e.g. shape grammars (Stiny & Gips, 1971) or split grammars (Wonka & Wimmer, 2003). Thereby, scientists focused on different purposes such as the creation of city models for movies/games (Müller et al., 2006), for reconstruction (Brenner & Ripperda, 2006), for facades (Müller et al., 2007), for detailed roofs (Dörschlag et al., 2008) or for the creation of stairs (Schmittwilken et al., 2007). Nevertheless, the aforementioned modeling approaches can only be used for visualization purposes, but standardized semantically enriched models cannot be derived.

Trying to generate semantic 3D building models, a new data model namely QUASY has also been presented by Benner et al. (2005), whereby this new model is very similar to CityGML. Additionally, it has been described how Industry Foundation Classes (IFC), i.e. a commonly used format for Business Information Modeling (BIM) (cf. IFC (2011)) can be transformed to this new model. However, QUASY is not a global standard, thus the massive application of the invented methodology and its benefit is questionable.

Falkowski et al. (2009) investigated how to generate full CityGML models from images. The authors utilize two stereo images and transfer them semi-automatically into a graph structure, which is then automatically transformed into a CityGML model. However, the approach requires manual work and is therefore not applicable for huge urban areas.

According to Isikdag & Zlatanova (2009) there are several use-cases for transferring IFC to CityGML, but literature lacks formal frameworks. Therefore, the authors present preliminary ideas, trying to semantically map both models with each other, while focusing on BIM. The presented framework describes fundamental ideas, but there is still a lot of work required for an automatic conversion. A first application of the framework concentrating on water utility networks is already presented by Hijazi & Ehlers (2009).

Altogether, the beforehand described approaches have in common that they all utilize proprietary or official data, which is on the one hand hard to acquire and on the other hand often expensive. As emphasized above, publically available data from VGI communities can serve as a real alternative data source, but none of the described approaches considers VGI.

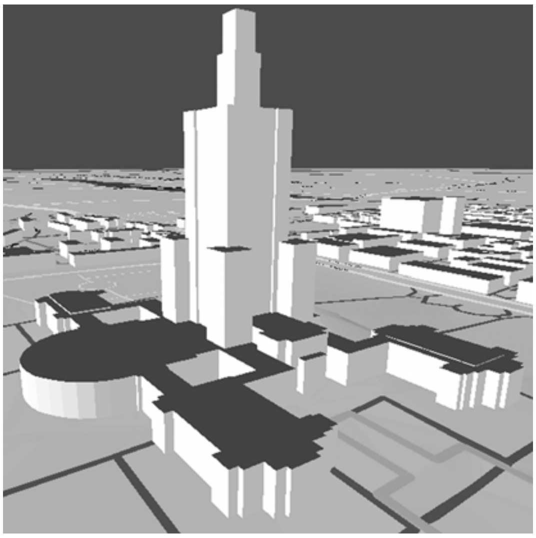

Fig. 2. 3D Visualization within the OSM3-D project (OSM-3D, 2011).

According to Over et al. (2010), there is currently no literature available which describes the 3D visualization of OSM data. There are only two applications providing a real three-dimensional perspective of OSM including digital terrain models and 3D buildings: the so called KOSMOS Worldflier (Brejc, 2011) and the OSM-3D project (OSM-3D, 2011). Some other applications try to show also perspective scenes (with flat terrain) and sometimes even extruded buildings, e.g. osm3d (Ziegler, 2011). However, most of these applications are very limited regarding the size of the scene, the application functionality and the selected data that can be visualized in 3D. In contrast, the mentioned OSM-3D project is based on a Web3DService (W3DS) and the W3DS-Client XNavigator provides a detailed virtual globe including terrain, landuse, Point-of-Interest (POI), buildings, streets, labels etc. Recently, realistic and detailed city models of entire Europe have been made available within OSM-3D, whereby the building models are updated regularly. Fig. 2. depicts an example of a building which has been mapped in OSM and rendered within the W3DS-Client XNavigator. It consists of several extruded polygons with different height, elevation and shape. This visualization already provides a coarse geometry of the building, however, currently the building generator behind OSM-3D does not yet support different roof types or other semantic aspects. The ongoing work in this project is supposed to improve this situation soon.

In general, to the authors of this paper's knowledge there is currently no work available on the (semi-)automated extraction of semantically enriched 3D models from OSM and in particular no work on the derivation of CityGML models.

THE FRAMEWORK

The derivation of CityGML models from OSM data must be accomplished in a two-step approach: on the one hand semantic information transformation is required and on the other hand the generation of valid geometries must be achieved. As described above, CityGML separates the semantic aspects strictly from the geometric aspects. In OSM there is no such strict separation. The 2D geometry of the ground shape is implicitly mapped by different tagged nodes. Additional geometric information, such as the height, as well as other semantic aspects, are attached as key-value pairs to the corresponding ground shape geometry. Because of this diversity in both models, the two aforementioned conversion steps must be accomplished together.

In order to perform a successful and consistent transformation, two operations are required. First, a comprehensive set of rules for the semantic mapping between key-values in OSM and attributes in CityGML needs to be defined clearly. Secondly, methodologies for the creation of geometries for each LoD in CityGML need to be developed.

The following sections present information transformations from OSM key-value pairs to CityGML attributes. Furthermore it is evaluated, which CityGML LoD (i.e. LoD 1 - 4) geometry can be created by purely using VGI from OSM.

Acquisition of Semantic Information from OSM for CityGML

For populating the attributes of the *_AbstractBuilding* class (cf. Fig. 3 (a)) in CityGML, several keys and/or values of OSM can be utilized.

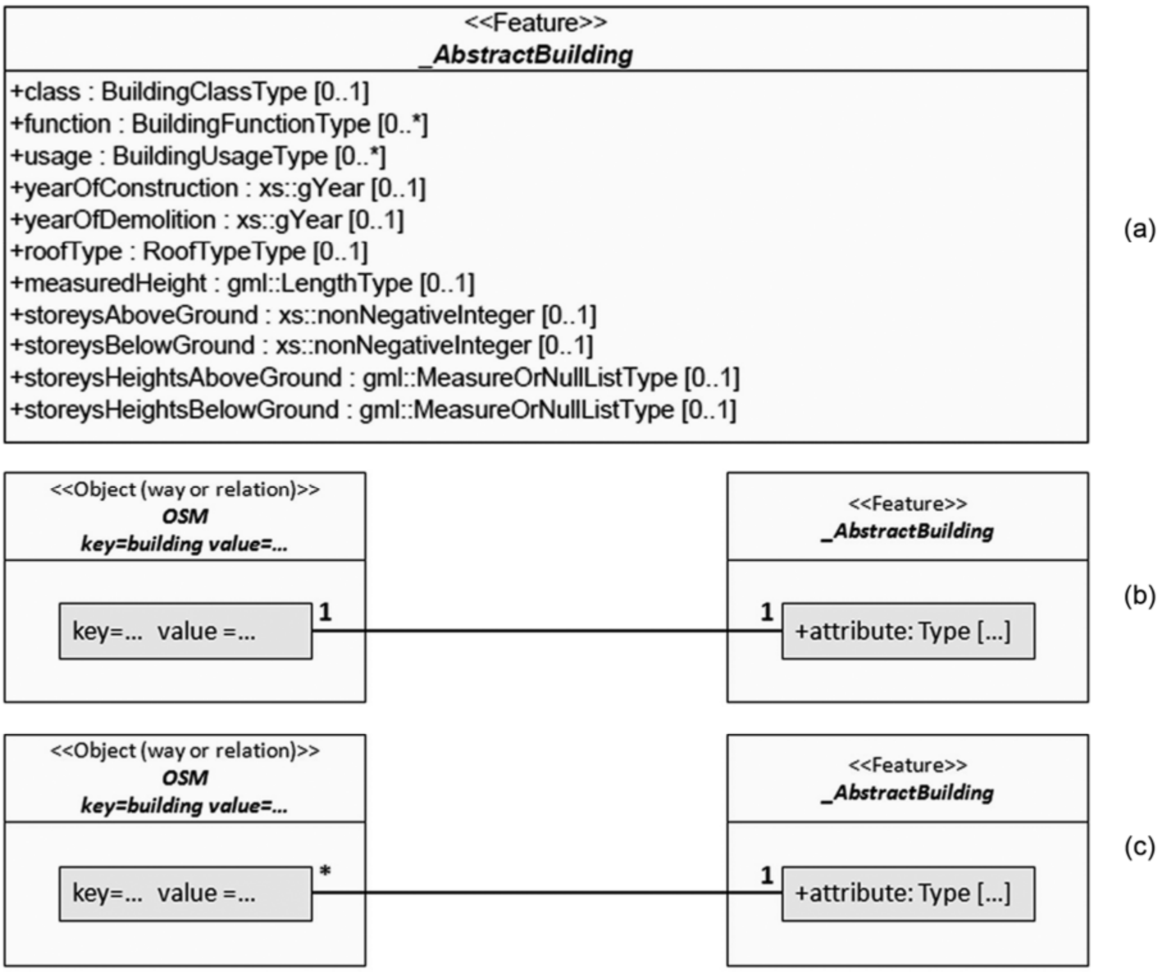

Fig. 3. Attributes of the _AbstractBuilding class (Gröger, et al., 2008) (a), OSM to CityGML relationship type 1:1 (b) and relationship type n:1 (c).

Generally, three classes of relationships can be distinguished when investigating the semantic mapping between OSM and CityGML:

(i) one key in OSM can be mapped to one attribute in CityGML (cf. Fig. 3 (b)), i.e. a 1:1 relationship
(ii) several keys in OSM can be mapped to a single CityGML attribute (cf. Fig. 3 (c)), i.e. a n:1 relationship
(iii) there is no suitable key in OSM for a CityGML attribute.

The case that one key in OSM can be mapped to several CityGML attributes (i.e. 1:n) does also occur. Since the developed framework focuses on unidirectional information transformation from OpenStreetMap to CityGML (i.e. it is only required to populate CityGML attributes by using OSM key-value pairs and not vice-versa), a 1:n relation can be simply divided into several direct relations (i.e. 1:1).

The examples for relationship type (i) are quite obvious. For the inherited attribute *gml:name*, the direct counterpart in OSM is the key *name* with an arbitrary value *v*. That is, using *name = v* results in *<gml:name>v</gml:name>* in CityGML. Similar to this, the attribute *bldg:yearOfConstruction* can be populated by using the value of the OSM key *building:buildyear*. The attribute *bldg:storeysAboveGround* can be populated respectively by using *building:levels:aboveground*, as well as the attribute *bldg:storeysBelowGround* by using *building:levels:underground*.

For the attribute *bldg:yearOfDemolition*, there is currently no counterpart in the OSM schema, thus this attribute belongs to the relationship type (iii). Currently, OSM does not support versioning of elements, i.e. whenever a building does no longer exist in real-world, it is deleted in the database. Therefore, the attribute *bldg:yearOfDemolition* cannot be populated by using VGI.

In contrast to the above mentioned attributes, the relationship type (ii) is more complicated, because different OSM keys can contain relevant information, thus a prioritization of those keys is required. One example for such an attribute is *bldg:class*. This attribute defines the class of a building, whereby the values are given as codes, representing a corresponding value. So e.g. a building with class code *1100* represents an academic facility such as a school or research department. Possible applicable counterparts in OSM are *amenity*, *building* or *building:use* whereby all of them can contain relevant information. Additionally, the key *building:type* can also contain relevant information, however, it has been declared as deprecated and shall no longer be used by the community (OSM, 2011d). That is, possible values for these three OSM keys have been investigated, evaluated according to their relevance and grouped in appropriate classes which can then be linked to the different possible BuildingClassTypes. Populating the CityGML attributes *bldg:function* and *bldg:usage* is done respectively. This value-partitioning and grouping is a very essential task and needs to be well considered, however, due to space limitations the complete framework cannot be discussed here in detail. As an example, *amenity = school* indicates that *bldg:class* is *1100*. In contrast, the *building:type = church* indicates that *bldg:class* is *1080* (church institution), and *building:use = residential* would result in *bldg:class 1000* (habitation). For more information on the code-lists of *bldg:class*, *bldg:function* and *bldg:usage*, refer to Gröger, et al. (2008) and for the most used values of the OSM keys *amenity*, *building:type* and *building:use*, please see Tagwatch (2011a).

For the CityGML attribute *bldg:measuredHeight* there are currently two potential OSM keys (namely *height* and *building:height*). Whenever both keys are available with different values, it needs to be decided which one to choose. However, this is not an easy decision, thus needs proper reasoning. A first proposal is to use the attribute with the latest value, as being most likely the most current version. Nevertheless, when investigating the data inside OSM it has been figured out that by end of October 2011 there are only few buildings available with differing height values (18 in total), whereby in most cases the height difference is less than three meters. Additionally, the key *building:height* has been declared as deprecated (OSM, 2011d), i.e. users are requested to no longer use it in the future. That is, if the community follows this request, *height* will be the only relevant key in the future, thus a one-to-one mapping between *bldg:measuredHeight* and *height* can be accomplished.

The CityGML attribute *bldg:roofType* is also defined via codes, representing specific values. Within OSM, the key *building:roof:shape* is mainly used for adding information about the roof type, although there are also three other keys (*building:roof:type* and *building:roof:style*). However, the latter two are hardly used (Tagwatch, 2011a), i.e. *building:roof:shape* is the most relevant key. Table 1 contains all possible codes for the CityGML attribute *bldg:roofType*. Furthermore, a mapping with the currently existing values of the key *building:roof:shape* (cf. Tagwatch (2011a)) is also proposed in Table 1.

Table 1. Mapping the values of building:roof:shape to bldg:class IDs.

OSM value for *building:roof:shape*	*bldg:class* Code	*bldg:class* name
flat	1000	flat roof
lean_to, lean-to, ridged, ridge	1010	monopitch roof
	1020	skip pent roof
gable, gabled, pitched	1030	gabled roof
hipped, hip	1040	hipped roof
	1050	half-hipped roof
mansard, gambrel	1060	mansard roof
crosspitched	1070	pavilion roof
cone, domical	1080	cone roof
	1090	copula roof
	1100	shed roof
catenary	1110	arch roof
pyramid, pyramidal,	1120	pyramidal broach roof
berlin	1130	combination of roof forms

The two remaining CityGML attributes *bldg:storeysHeightsAboveGround* and *bldg:storeysHeightsBelowGround* do not have a direct counterpart in OSM (i.e. mapping class (iii)). Nevertheless, they can be populated by approximate calculations. Both attributes contain an ordered list of the heights of the storeys, whereby those above the ground are listed in ascending order and those below the ground in descending order. The different values for the attribute list *bldg:storeysHeightsAboveGround* can be calculated by dividing the value of *bldg:measuredHeight* by the value of *bldg:storeysAboveGround*. However, this calculation is very approximative and the results can vary greatly from the real world. Due to missing information the attribute list *bldg:storeysHeightsBelowGround* cannot be provided. Nevertheless, a very coarse solution is to define all heights below ground equally to the average height of those above the ground.

One very important attribute of *_AbstractBuilding* is *bldg:address*, which describes the address of the building. It is provided within the feature *core::Address* and consists of several attributes. The values for these attributes can all be gathered from VGI, because there is a corresponding OSM key for all of them. That is, all attributes within *core::Address* can be populated with a direct one-to-one mapping, thus they belong to mapping class (i). The XML structure of *bldg:address* is depicted in Fig. 4, whereby the required OSM keys are given in braces.

Concluding it can be said that all attributes of *_AbstractBuilding* (except *bldg:yearOfDemolition*) can be populated with information from OSM. The question whether it is likely or not that these values will be provided by the OSM contributors, will be discussed in the last chapter.

```
<bldg:address>
  <Address>
    <xalAddress>
      <xAL:AddressDetails>
        <xAL:Country>
          <xAL:CountryName>{VALUE OF OSM KEY addr:country}</xAL:CountryName>
          <xAL:Locality Type="Town">
            <xAL:LocalityName>{VALUE OF OSM KEY addr:city}</xAL:LocalityName>
            <xAL:Thoroughfare Type="Street">
              <xAL:ThoroughfareNumber>{VALUE OF OSM KEY addr:housenumber}</xAL:ThoroughfareNumber>
              <xAL:ThoroughfareName>{VALUE OF OSM KEY addr:street}</xAL:ThoroughfareName>
            </xAL:Thoroughfare>
            <xAL:PostalCode>
              <xAL:PostalCodeNumber>{VALUE OF OSM KEY addr:postcode}</xAL:PostalCodeNumber>
            </xAL:PostalCode>
          </xAL:Locality>
        </xAL:Country>
      </xAL:AddressDetails>
    </xalAddress>
  </Address>
</bldg:address>
```

Fig. 4. Attributes of the _AbstractBuilding class.

Derivation of CityGML LoD1 Building Models

In CityGML, LoD1 buildings are visualized as the “well-known blocks model comprising buildings with flat roofs” (Gröger, et al., 2008, pp. 9). As described in the CityGML schema (cf. (Gröger, et al., 2008)), the geometry of a building in LoD1 can be represented in two different ways: on the one hand by utilizing *gml:_Solid* for modeling the building as a volumetric object, or on the other hand by utilizing *gml:MultiSurface* for modeling the exterior surface of the building. Generally, every wall of the building is represented as a single flat surface. However, general details e.g. edges or holes are also visualized (cf. Fig. 5 (b)).

A quite straight forward approach for generating LoD1 CityGML models from OSM is to acquire the ground shape of the building from the tagged nodes, ways and relations (representing the ground floor of the building) and the height of the building from the corresponding tag-value pairs in OSM, as already described in the previous section. That is, by extruding the building footprint with the corresponding height, a LoD1 model geometry can be easily created. Fig. 5 (a) depicts an exemplary building footprint. By extruding this footprint, a CityGML LoD1 model is created in Fig. 5 (b). As described beforehand in the OSM introduction, users can map buildings by either using one single closed way or by using a relation (consisting of several closed ways). In the former case, the footprint extraction is straightforward: the closed way represents the outer shell of the footprint polygon, i.e. the coherent area which is enclosed by the OSM linestring represents the footprint polygon. In the latter case, it needs to be considered that there are holes inside the building footprint, which makes the building footprint creation process a bit more complicated. After creating the footprint polygon based on the outer shell, it is additionally necessary to create a polygon for each individual inner hole (also by computing the coherent area which is enclosed by the corresponding OSM linestring). The final building footprint polygon can then be gathered by subtracting the inner polygons (i.e. the holes) from the outer polygon (i.e. the shell).

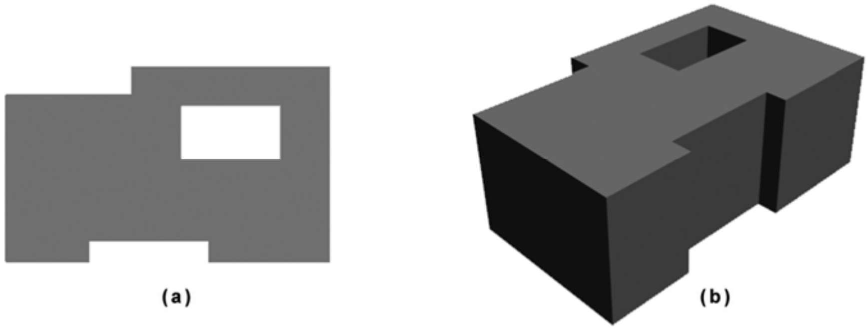

Fig. 5. Complex shaped building footprint with a hole (a) and corresponding geometrical representation of this building in CityGML LoD1 (b)

Derivation of CityGML LoD2 Building Models

Compared to a CityGML LoD1 model, a LoD2 model's façade and roof is represented in a greater detail, which is the main difference between these two LoDs. In particular, the shape of the roof is modeled as a real geometry (and not only as a flat roof). From a geometrical point of view, in LoD2 the outer walls are represented by multiple faces and curvatures in the façade are also visualized. Additionally, in LoD2 it is possible to represent outer building installations such as balconies, dormers, stairs etc. within the BuildingInstallation class as an aggregation of different geometric types of *gml:Geometry*. In contrast to LoD1, in LoD2 it is furthermore possible to model different outer building parts with different classes. These classes, namely *RoofSurface*, *WallSurface*, *GroundSurface* and *ClosureSurface* are combined in the parental class *BoundarySurface* and can be utilized for a separate visualization of roof shapes, wall shapes, ground shapes and closure shapes. Furthermore LoD2 models allow the provision of textures to walls and/or roofs. That is, different outer building parts can be either differentially colored or can be wrapped by a 2D texture.

In order to generate CityGML LoD2 models, the WallSurface objects in the CityGML model will be generated based upon the mapped ground shape. Each segment (i.e. the line between two adjacent OSM nodes) of the way which is utilized for mapping the ground shape is individually extruded with the height of the building, thus individual WallSurface geometries are generated. For modeling building elements which are above the ground or closures in the surface, there is additional information required. OSM provides the two keys *building:min_height* and *building:min_level*, whereby the former describes the height of the space between the ground and the building and the latter the amount of storeys which are between the ground and the building. So whenever a building(part) is enriched with one of these keys, the corresponding *WallSurface* geometry needs to be raised in the air. In the case that *building:min_level* is provided, general assumptions about the average height of a level need to be performed.

Table 2. OSM keys containing information about the building roof.

OSM key	Exemplary Value	Alternative
building:roof	tile, slate, flat, tile_red, reet	
building:roof:angle	30, 45, 10, 15	
building:roof:colour	grey, red, brown	building:roof:color
building:roof:material	shingles, slate, cardboard	
building:roof:orientation	along, across	
building:roof:extent	0.1, 0.3	
building:roof:ridge	yes	
building:roof:shape	pitched, hipped, flat, ridged	building:roof:style, building:roof:type

Additionally, the geometry of the roof (i.e. the *RoofSurface* class) needs to be modeled. For that purpose, OSM provides different keys with relevant information about the roof. Table 2 lists the most relevant OSM keys containing information about the roof of a building, as well as potential alternative OSM keys which might contain similar information (last column). However, the keys in the first column are those which are used more regularly, so wherever applicable, the keys in the first column are to be preferred. Concluding, it can be said that by analyzing the values of these keys it should be possible to create a roof geometry which is quite similar (in the best case even equivalent) to the real roof, at least for the sake of LoD2 buildings. Since methodologies and algorithms for the creation of roof geometries are quite complex, and information about detailed (sub)structures of the roof or triangulation points are missing in OSM, it is not possible to describe the geometry creation process in detail within this paper. Generally, good results can be achieved by using skeleton computation with procedural extrusion (Kelly & Wonka, 2011; Laycock & Day, 2003), however a broad application for many buildings results in high computation costs, and there are also many special cases and exceptions which yet require manual adjustments.

The ground geometry of the roof (i.e. the plane between the building and the beginning of the roof) can be generated by acquiring and elevating the geometry of the ground shape of the building. When a building roof has eaves, OSM provides the key *building:roof:extent* for describing their length. That is, whenever *building:roof:extent* is provided, it must be considered while creating the roof geometry. The *GroundSurface* object in CityGML LoD2 can be generated respectively (without considering the extent).

As stated above, CityGML LoD2 models can also be enriched with textures. Within OSM there are currently no keys proposed for providing a link to a texture for the façade or roof. Nevertheless, there are some OSM keys which contain information about the roof material and color (cf. Table 2). Additionally, there are two keys, namely *building:cladding* and *building:facade:colour* which contain information about the building façade. By analyzing these keys it should be possible to either create an appropriate but simplified synthetic texture on-the-fly (Coors, 2008) or to select a texture from some kind of predefined textures database.

For modeling details about building installations such as stairs or balconies, there are currently no appropriate keys promoted in OSM. That is, most building installations in CityGML LoD2 cannot yet be created from OSM data.

Derivation of CityGML LoD3/LoD4 Building Models

The main characteristics of LoD3 building models (in contrast to LoD2 models) are that outer building openings and installations such as windows, doors or chimneys are visualized. Since there are currently no examples in OSM of how to map such details and there is also no methodology presented of how to do it, the OSM database does not contain such detailed building information. That is, currently it is not possible to generate LoD3 models from OSM.

The even more detailed LoD4 models do visualize inner building parts. However, until now OSM does practically not provide any detailed information about inner floor plans and footprints. There are only some very rare examples (cf. (OSM, 2011e)) of footpaths that go inside buildings and provide indoor routes on the ground floor (but not on any other floors). Therefore, at the moment it is not possible to derivate LoD4 models from the OSM database.

CONCLUSIONS AND FUTURE WORK

Collaboratively and voluntarily collected geoinformation can serve as a real alternative data source for different applications. For automatically creating standardized and interoperable 3D CityGML models which can be used in professional GIS applications, information from

VGI communities (especially OpenStreetMap), could be utilized, if a formal framework can be made available. Following an introduction to CityGML and the VGI community OpenStreetMap, a background literature review is provided and afterwards a general overview of information transformation (both semantic and geometric) from OSM to CityGML is proposed. The presented framework discusses which semantic attributes of CityGML can be derived and furthermore investigates how geometries in different CityGML LoDs can be extracted from OSM data. An overview about the results of the conducted investigations is summarized in Table 3.

With this framework, an ideally mapped building (i.e. a building enriched with all required key-value pairs) can be extracted from OSM as CityGML model. However, due to missing ideas and methodologies for mapping building installations or indoor spaces in a key-value pair based way (as described beforehand), it is currently not possible to generate LoD3 or LoD4. Nevertheless, by applying the framework to an ideally tagged urban region, all buildings can be extracted as CityGML LoD1/LoD2 models, thus can be exchanged and utilized in professional applications.

Table 3. Summary about feasible and non-feasible transformations from OSM to CityGML.

What	Result	Comment
CityGML attributes	feasible	Nearly all CityGML attributes (except *bldg:yearOfDemolition*) can be populated by using VGI from OSM
LoD1 geometry	feasible	A blocks model can be created by extruding ground shape geometry of the building (provided as a way or relation in OSM) with the building height (provided as OSM key).
LoD2 geometry	feasible	A building model with individual surfaces (e.g. *GroundSurface*, *WallSurface* etc.) as well as a roof geometry can be created by extruding the different segments of the ground shape polygon (similar to LoD1) and computing a suitable geometry for the corresponding roof type (based on provided OSM keys).
LoD3 geometry	not feasible	OSM does not yet contain information about windows or doors, thus a LoD3 geometry cannot be created by purely using VGI from OSM
LoD4 geometry	not feasible	OSM does not yet contain information about rooms inside buildings or their interior structure, thus an LoD4 geometry cannot be created by purely using VGI from OSM

As described above, for an ideal result it is crucial that all relevant data is provided. Currently, there are only a few buildings with all relevant information available and therefore the broad application of the presented framework will not lead to satisfying results. That is, the usage of the relevant keys needs to be promoted inside OSM, so that more and more buildings will be enriched with the required information. One way for such a promotion is the development and improvement of 3D applications such as OSM-3D, because these demonstrate to the OSM community, why it is useful to contribute the corresponding building data. A broad availability of high-quality mapped buildings inside OSM will result in detailed CityGML models for several areas.

Also, the quality of the available data is not yet investigated in detail. There are some investigations regarding OSM accuracy and completeness (as described in the OSM introduction at the beginning of this paper), but their focus is more on the landscape and street network and not that much on the buildings and their attributes. That is, additional investigations and comparisons (for example to official data) need to be performed. By doing so, the expected high quality of OSM can be demonstrated.

The framework presented in this paper only concentrated on unidirectional information transformation from OSM to CityGML models, as the need for such a transformation appears more eminent today. Nevertheless, bidirectional transformation also might be required to support the import of official CityGML models to OpenStreetMap. However, the authors of this paper argue that the transformation from OSM to CityGML seems to be more complex than the other way round, thus a conversion from CityGML to OSM should also feasible.

As a future step, work on the development of an algorithm which implements the presented framework and automatically creates CityGML LoD1 and LoD2 buildings from OSM will be undertaken, which is an interesting, but also challenging task. Especially an algorithm which is capable to deal with buildings that are not ideally tagged, thus generates adequate building hypotheses, is important and desirable. Also, the creation of adequate roof geometries needs to be investigated.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for providing valuable comments towards the improvement of this paper. Furthermore we would also like to thank all members of the Chair of GIScience for their proofreading and helpful hints. This research has been partially funded by the Klaus-Tschira Foundation (KTS) Heidelberg.

REFERENCES

Benner, J., Geiger, A., & Leinemann, K. (2005). Flexible Generation of Semantic 3D Building Models. Paper presented at the 1st International Workshop on Next Generation 3D City Models, Bonn, Germany.

Brejc, I. (2011). Kosmos WorldFlier. Retrieved 07/09/2011, from http://igorbrejc.net/category/3d

Brenner, C., & Ripperda, N. (2006). Extraction of Facades Using RJMCMC and Constraint Equations. Paper presented at the ISPRS Commission III Photogrammetric Computer Vision, Bonn, Germany.

CityGML. (2011). CityGML Dateset of Castle Herten. Retrieved 07/15/2011, from http://www.citygml.org

Coors, V. (2008). Synthetic textures for 3d urban models in pedestrian navigation. Paper presented at the Techniques and Applications for Mobile Commerce (TAMoCo 2008), Glasgow, United Kingdom

Czerwinski, A., Kolbe, T. H., Plümer, L., & Stöcker-Meier, E. (2006). Spatial data infrastructure techniques for flexible noise mapping strategies. Paper presented at the 20th International Conference on Environmental Informatics - Managing Environmental Knowledge, Graz, Austria.

Döllner, J., Kolbe, T. H., Liecke, F., Sgouros, T., & Teichmann, K. (2006). The Virtual 3D City Model of Berlin - Managing, Integrating, and Communicating Complex Urban Information. Paper presented at the 25th Urban Data Management Symposium (UDMS 2006), Aalborg, Denmark.

Dörschlag, D., Gröger, G., & Plümer, L. (2008). Über die schrittweise Erstellung und Verfeinerung von Modellhypothesen für Gebäude. *Photogrammetrie, Fernerkundung, Geoinformation*, 3, 157-164.

Falkowski, K., Ebert, J., Decker, P., Wirtz, S., & Paulus, D. (2009). Semi-automatic generation of full CityGML models from images. Paper presented at the Geoinformatik 2009, Münster, Germany.

Goodchild, M. F. (2007a). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4), 211-221.

Goodchild, M. F. (2007b). Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0. *International Journal of Spatial Data Infrastructures Research*, 2, 24-32

Gröger, G., Kolbe, T. H., Czerwinski, A., & Nagel, C. (2008). OpenGIS City Geography Markup Language (CityGML) Encoding Standard – Version 1.0.0. OGC Doc. No. 08-007r1.

Haklay, M. (2010). How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B-Planning & Design*, 37(4), 682-703. doi: 10.1068/b35097

Henricsson, O., Bignone, F., Willuhn, W., Ade, F., Kübler, O., Baltsavias, E., Grün, A. (1996). Project Amobe: strategies, current status, and future work. *International Archives of Photogrammetry and Remote Sensing*, 31(Part B3), 321-330.

Hijazi, I., & Ehlers, M. (2009). Web 3D Routing in and between Buildings. Paper presented at the 4th National GIS Symposium, Al-Khobar, Saudi Arabia.

IFC. (2011). IFC Overview summary - Welcome to buildingSMART-Tech.org. Retrieved 10/21/2011, from http://buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary

Isikdag, U., & Zlatanova, S. (2009). Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models In J. Lee & S. Zlatanova (Eds.), *3D GEO-INFORMATION SCIENCES LNG&C Part II* (pp. 79-95). Heidelberg: Springer.

Kelly, T., & Wonka, P. (2011). Interactive Architectural Modeling with Procedural Extrusions. *ACM Transactions on Graphics*, 30(2), 14-28.

Kolbe, T. H. (2009). Representing and Exchanging 3D City Models with CityGML. In J. Lee & S. Zlatanova (Eds.), *3D GEO-INFORMATION SCIENCES LNG&C Part I* (pp. 15-31). Heidelberg: Springer.

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML – Interoperable Access to 3D City Models. Paper presented at the 1st International Symposium on Geo-information for Disaster Management, Delft, Netherlands.

Kolbe, T. H., Gröger, G., & Plümer, L. (2008). CityGML – 3D City Models and their Potential for Emergency Response. In S. Zlatanova & J. Li (Eds.), *Geospatial Information Technology for Emergency Response* (pp. 257-274): Taylor & Francis.

Lake, R., Burggraf, D., Trninic, M., & Rae, L. (2004). *Geography Mark-Up Language: Foundation for the Geo Web*. Chichester, United Kingdom: John Wiley & Sons.

Lang, F., & Förster, W. (1996). 3D-city modeling with a digital one-eye stereo system. *International Archives of Photogrammetry and Remote Sensing*, 31(Part B4), 261-266.

Lapierre, A., & Cote, P. (2007). Using Open Web Services for urban data management: A testbed resulting from an OGC initiative for offering standard CAD/GIS/BIM services. Paper presented at the 26th Urban Data Management Symposium (UDMS 2006), Stuttgart, Germany.

Laycock, R. G., & Day, A. M. (2003). Automatically Generating Roof Models from Building Footprints. Paper presented at the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG' 03), Plzen - Bory, Czech Republic.

Maas, H.-G., & Vosselman, G. (1999). Two algorithms for extracting building models from raw laser altimetry data *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2-3), 153-163.

Mäs, S., Reinhardt, W., & Wang, F. (2006). Conception of a 3D Geodata Web Service for the Support of Indoor Navigation with GNSS. In A. Abdul-Rahman, S. Zlatanova & V. Coors (Eds.), *Innovations in 3D Geoinformation Science Lecture Notes in Geoinformation and Cartography* (pp. 307-316). Heidelberg: Springer.

Mooney, P., Corcoran, P., & Winstanley, A. C. (2010). Towards quality metrics for OpenStreetMap. Paper presented at the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, United States.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., & Van Gool, L. (2006). Procedural modeling of buildings. *ACM Transactions on Graphics*, 25(3), 614 - 623.

Müller, P., Zeng, G., Wonka, P., & van Gool, L. (2007). Image-based Procedural Modeling of Facades. *ACM Transactions on Graphics*, 26(3), 85–93.

Neis, P., Zielstra, D., Zipf, A., & Struck, A. (2010). Empirische Untersuchungen zur Datenqualität von OpenStreetMap - Erfahrungen aus zwei Jahren Betrieb mehrerer OSM-Online-Dienste. Paper presented at the AGIT 2010 Symposium für Angewandte Geoinformatik, Salzburg. Austria.

OSM-3D. (2011). OSM-3D in XNavigator. Retrieved 07/14/2011, from http://www.osm-3d.org

OSM. (2011a). OpenStreetMap. Retrieved 07/14/2011, from http://www.openstreetmap.org/

OSM. (2011b). Stats - OpenStreetMap Wiki. Retrieved 06/30/201, from http://wiki.openstreetmap.org/wiki/Statistics

OSM. (2011c). Bing - OpenStreetMap Wiki. Retrieved 10/24/2011, from http://wiki.openstreetmap.org/wiki/Bing

OSM. (2011d). Proposed features/Building attributes. from http://wiki.openstreetmap.org/wiki/Proposed_features/Building_attributes

OSM. (2011e). Indoor Mapping - OpenStreetMap Wiki. Retrieved 10/21/2011, from http://wiki.openstreetmap.org/wiki/Indoor_Mapping

Over, M., Schilling, A., Neubauer, S., & Zipf, A. (2010). Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *Computers, Environment and Urban Systems*, 34(6), 496-507.

Schilling, A., Over, M., Neubauer, S., Neis, P., Walenciak, G., & Zipf, A. (2009). Interoperable Location Based Services for 3D cities on the Web using user generated content from OpenStreetMap. Paper presented at the 27th Urban Data Management Symposium (UDMS 2009), Ljubljana , Slovenia.

Schmittwilken, J., Saatkamp, J., Förster, W., Kolbe, T. H., & Plümer, L. (2007). A Semantic Model of Stairs in Building Collars. *Photogrammetrie, Fernerkundung, Geoinformation PFG*(6), 415–428.

Song, W., & Sun, G. (2010). The role of mobile volunteered geographic information in urban management. Paper presented at the 18th International Conference on Geoinformatics, Beijing, China.

Stiny, G., & Gips, J. (1971). Shape Grammars and the Generative Specification of Painting and Sculpture. Paper presented at the IFIP Congress, Ljubljana, Yugoslavia.

Tagwatch. (2011a). Watchlist. Retrieved 07/14/2011, from http://tagwatch.stoecker.eu/Planet-latest/En/watchlist.html

Tagwatch. (2011b). OpenStreetMap Tagwatch for Planet-latest :: Watchlist. Retrieved 07/14/2011, from http://tagwatch.stoecker.eu/Planet-latest/En/tags.html

Weidner, U., & Förster, W. (1995). Towards automatic building reconstruction from high resolution digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing* 50(4), 38-49.

Wonka, P., & Wimmer, M. (2003). Instant Architecture. *ACM Transaction on Graphics*, 22(3), 669-677.

Ziegler, S. (2011). osm3d Viewer. Retrieved 07/14/2011, from http://www.osm3d.org

Zielstra, D., & Zipf, A. (2010). A Comparative Study of Proprietary Geodata and Volunteered Geographic Information for Germany. Paper presented at the 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal.

Zipf, A., Basanow, J., Neis, P., Neubauer, S., & Schilling, A. (2007). Towards 3D spatial data infrastructures (3D-SDI) based on open standards – experiences, results and future issues. Paper presented at the 3D GeoInfo07, ISPRS WG IV/8 international workshop on 3D geo-information: Requirements, acquisition, modeling, analysis, visualisation, Delft, Netherlands.