reviewed paper

Enriching the 3D City Model for the Simulation of Urban Heat Demand

M. Esteban Muñoz H., Ivan Dochev, Hannes Seller, Irene Peters

(Marcelo Esteban Muñoz Hidalgo, HafenCity University Hamburg, Germany, marcelo.hidalgo@hcu-hamburg.de)
(Ivan Dochev, HafenCity University Hamburg, Germany, ivan.dochev@hcu-hamburg.de)
(Hannes Seller, HafenCity University Hamburg, Germany, hannes.seller@hcu-hamburg.de)
(Irene Peters, HafenCity University Hamburg, Germany, irene.peters@hcu-hamburg.de)

1 ABSTRACT

In this paper we describe the process of enriching the Hamburg 3D City model (3D-Stadtmodell) with energy relevant attributes for the simulation of heat demand. The 3D data of the city provided by the Landesbetrieb Geoinformation und Vermessung (LGV) is a combination of Cadastre footprints and LiDAR data. This combination of data allows the LGV to produce CityGML data with a level of detail 1 (LOD1). We use this data as basis for the computation of urban heat demand.

This paper presents the enrichment process of the CityGML data. We make use of the energy application domain extension (ADE) to store the energy relevant data in a standardized format. For the enrichment process we classify the residential building stock into building types. And classify the non residential sector by use. From the building types we extract heat transmission coefficients of building components. With the enriched 3D city model we perform a monthly heat demand estimation of a selected neighbourhood in Hamburg.

The aim of this enrichment process is to create a robust but flexible method for the estimation of heat demand at a neighbourhood level with little energy relevant information. This paper presents a method for a quick estimation of the monthly heat demand of a neighbourhood without the need of any extra data input. This approach can be used by the energy and urban planning community for a first estimation of the heat demand used on a given neighbourhood or the entire city.

The results from this approach present an urban heat demand model for the city of Hamburg based on the freely available 3D city model data. Possible uses of this approach are: (1) identification of hot spots in the city, (2) creation of base data sets for the simulation of retrofit scenarios, and (3) creation of temporal heat density maps.

2 INTRODUCTION

On this paper we present a small program for the enriching of the 3D city model of the city of Hamburg. We develop a python command line interpreter that is able to:

(1) download CityGML data from different locations;

(2) process the data and retrieve relevant information;

(3) save the data as different data formats, including CityGML with Energy ADE relevant attributes; and

(4) compute heat demand for the residential and non residential sector

The main idea of the command line interpreter is to create a simple set of python scripts to process geo-spatial data for the computation of heat demand. This process is scripted on small text files for documentation purpose and for the processing of large data-sets.

The enrichment of process is a simple read and write procedure. In between the data input and data output we define a set of templates for the construction of CityGML files. The definition of templates for the storage of data is exclusive for the CityGML data files. Other common data formats do not need a predefined template to be used. The developed tool makes use of the fiona[1] python library for the read-write of spatial data. This library can read and write all OGR Vector Formats[2].

3 METHOD

In this section of the paper we briefly describe the underlying method of the presented tool. One of the most important methods used in the tool is the use of templates for the creation of CityGML files. This approach

[1] https://pypi.python.org/pypi/Fiona/0.8
[2] http://www.gdal.org/ogr_formats.html

ISBN 978-3-9504173-0-2 (CD), 978-3-9504173-1-9 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER

has the advantage of been very flexible because the files are created only based on the predefine templates. The disadvantage of this method is that the tool is not aware of errors implemented in the template. The tool provides a link to the xmllint[3] XML files validator.

Internally the tool will populate a template with predefine variables retrieved from either the downloaded data of from other sources joint to the data. For the example presented on this paper we define 3 templates:

(1) A header template, containing the header information of the xml file including all the namespace definitions;

(2) Some lines describing the generated file, including the bounding box of the geometry features; and

(3) A template describing a building element;

Both the header and the file definition template are used a single time. The building template is used for each processed building on the data-set. In order to start populating the xml file with the processes templates the tool needs to process the entire data set. This is required in order to compute the bounding box of the file. Within the developed tool all the information required to populate the xml file is stored on a GeoPandas[4] DataFrame.

One of the motivations to use a template system is its ability to adapt to new schemas without much effort. In this case we construct the templates based on the EnergyADE schema (Nouvel, Bahu, et al., 2015, Nouvel, Kaden, et al., 2015).

3.1 Defining CityGML Data Structures Through Templates

The use of templates for the construction of xml files is extremely straightforward. The first step is to create some pieces of xml code with pre define variables on them. These pieces of code can be arbitrarily defined decoupling so the development of the tool to any development of CityGML schema of any other xml schema.

Listing 1 shows an extract of the building template. On this template we define a singe variable: $sqm. The value of this variable will be populated by the computed heated floor area by the tool. In this case we leave the uom variable with our preset default m2. Each template can have as many variables defined as needed. In the case of Listing 2 we define five variables. In this case we populate this template with the computed heat demand of each building. Because the tool computes both monthly and yearly heat demands the unit of measure (uom) and the time interval unit have to be defined as variables as well.

```
1 <!-- Energy heated area -->
2 <energy:floorArea>
3     <energy:FloorArea>
4         <energy:type>EnergyReferenceArea</energy:type>
5         <energy:value uom="m2">$sqm</energy:value>
6     </energy:FloorArea>
7 </energy:floorArea>
```

Listing 1: Template for energy heated area.

3.2 Scripting the Enrichment Process

The tool is developed as a command line interpreter. This means that the tool can be used on the command line of we can run a script with it. We see a huge advantage of using a command line interpreter for this task rather than a GUI interface. The most important advantage is, developing time, without a GUI the tool is very light, easy to maintain and has a quick development cycle. But probably the mos important advantage of a command line interpreter is the ability to store the performed analysis on a script. Defining an analysis on a scrip can also be useful for the processing of large data sets.

Listing 3 shows the script used to estimate the heat demand of the urban area presented on this paper. Available commands for the command line interpreter are still limited and many of the options are still embedded on the code as default values. The example presented on Listing 3 shows the simplest process to compute heat demand for an urban area using the default values of the tool.

[3] http://infohost.nmt.edu/tcc/help/xml/lint.html
[4] http://geopandas.org/

Code line 2 defines the data set to use, if the data set is not available on disk, the tool will attempt to download it from a pre defined source. The information needed to perform this operation is stores a json file containing relevant information on available data sets. For now the tool has a limited understanding on how to access external data. The basic definition of data locations is for now just the download link and a list of default attributes available on this data set. We aim to expand this by defining more complex connections to external data sets like databases.

```
1  <!-- Computed energy demand -->
2  <energy:energyDemands>
3      <energy:EnergyDemand>
4          <energy:endUse>SpaceHeating</energy:endUse>
5          <energy:energyAmount>
6              <energy:RegularTimeSeries>
7                  <energy:id>$heatid</energy:id>
8                  <energy:temporalExtent>TMY</energy:temporalExtent>
9                  <energy:timeInterval unit="$timeintervalunit">
10                     $timeinterval
11                 </energy:timeInterval>
12                 <energy:values uom="$uom">
13                     $heatdemand
14                 </energy:values>
15             </energy:RegularTimeSeries>
16         </energy:energyAmount>
17     </energy:EnergyDemand>
18 </energy:energyDemands>
```

Listing 2: Template for computed heat demand

Code line 4 deletes the attribute list retrieved from the configuration file. We delete this list because we do not want to keep all the attributes. On the next line we add three attributes to be used for the computation of heat demand. The data set define on code line 2 contains all the buildings for the city of Hamburg. We do not want to compute the heat demand for the entire building stock but just for a small urban area. We define this urban area as a bbox_clip. This bounding box can be define with an upper left and a lower right coordinates or with a spatial file. In this case we use a precomputed GeoJSON file to define the simulation area.

The tool has a small option to enrich the input data through the use of data store on csv files. In this case we input a csv file called bja.csv located on the path data relative to where the command line interpreted is run. The default options assumes that the file contains headers and use them to identify the variables store on the file. We make use of the headers name to filter them. In this case we join three attributes to the input data set. On code line 13 we can see that we enrich the dataset with construction years bja and building type baw. The join of both datasets is performed on the index. The index of the city 3D data is taken from the xml files and the index for the csv file is define on the first column. It is important to have the same index on both datasets, otherwise a join will fail.

```
1  #download input data by name
2  download hamburg
3  #prepare the attributes to use
4  delete_attribute all
5  add_attribute function measuredHeight storeysAboveGround
6  #clip the geometry with a geo-json file
7  add_bbox_clip data/output/2016-02-16T04:26:22.145116-clip.json
8  #get the geometry of the buildings within the specified area
9  get_geometry all
10 #keep only objects within the simulation area
11 clip all
12 #join some data stored on a csv file by index
13 join data/bja.csv bja baw
14 #simplify the building geometry
15 simplify all
16 #compute yearly heat demand with a static method
17 #using climate from Hamburg
18 compute_heat Year static Hamburg
19 #save the result as different formats
20 save_geometry GeoJSON csv CityGML shp
21 #validate the generated CityGML file against a defined xsd schema
22 validate xsd/energy.xsd
```

Listing 3: Template for computed heat demand

ISBN 978-3-9504173-0-2 (CD), 978-3-9504173-1-9 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER

The next command on the script: simplify all, code line 15, performs a geometrical simplification of the buildings, see next section. The computation of heat demand occurs on code line 18. On this line of the script we define a couple of parameters for the computation of heat demand. The first parameter defines de temporal aggregation of the simulation. The next parameter defines the type of simulation to perform, either a static steady state simulation using the heat (Muñoz H., 2015) library or a thermal simulation using EnergyPlus. The last parameter defines the climate data to use. This parameter can either be define as a path to a climate file or if using a climate one define on the heat library just as a string with the name of the climate zone.

3.3 Buildings

On the developed tool we have a small module for simplifying the geometry of buildings for the computation of heat demand. A simplification of the building geometry allows us to perform faster heat demand computations of individuals buildings and maintain a simpler data structure. An advantage of having a simple geometry definition for the individual buildings on our data set is the projection of the defined building stock into the future.

Kim, Plessis, Hubert, and Roux (2014) show that a simplification of the building geometry decreases the performance of the model by less than 1% of total annual heat demand but are able to perform the same simulation 700 times faster than taking a complex geometry into account. The implemented geometry simplification on the developed tool dos not take into account the neighbouring buildings. An algorithm considering the neighbouring buildings into account for a geometry simplification is described on (Muñoz Hidalgo, 2016 –in Press–).

The simplification algorithm does not take into account the overall orientation of the individual building components. In order to account for the building orientation of the individual building components we need to compute a weighted average of the orientations and define a general orientation for each building. The implemented algorithm defines a simple building geometry as length a with of the building, but keeping the building area to volume ratio.

3.4 Heat Transmission Coefficients

With the enriched data, through the csv file join, we classify the building stock into building typologies. The default building typology define in the tool is the IWU typology (Diefenbach, Cischinsky, Rodenfels, & Clausnitzer, 2010, Loga, Diefenbach, & Born, 2011). Table 1 shows the IWU building typology matrix. The classification of the building stock is performed based on the construction type baw and the construction year bja. These attributes are not part of the cityGML files provided by the city of Hamburg. This classification of the building stock does only work for the residential sector. We use this classification of the building stock into typologies in order to infer the value of the heat transmission coefficients of the building components and to estimate the percentage of gazing surface.

	< 1859	1860–1918	1919–1948	1949–1957	1958–1968	1969–1978	1979–1983	1984–1994	1995–2001	2002–2009
EFH[a]	183	180	164	181	146	155	118	132	110	88
RH		153	137	156	106	127	127	98	78	86
KMH	190	143	168	156	129	134	118	122	92	79
GMH		127	144	142	131	117				
HH					114	113				

Table 1: IWU-de building typology matrix for Germany. Source: (Loga, Diefenbach, & Born, 2011) Specific Heat demand (spez. Wärmebedarfskennzahl) [kWh/m2a]. (EFH) Single family house “Einfamilienhaus”; (RH) Terrace house “Reihenhaus”; (KMH) Apartment house “Mehrfamilienhaus”; (GMH) Large apartment house “Großes Mehrfamilienhaus”; (HH) High-rise “Hochhaus”.

Currently the estimation of heat demand for the non residential sector has been only implemented for a yearly temporal resolution. The computation of the heat demand for the non residential sector is carried with help of the VDI-3807 (Verein Deutscher Ingenieure, 1994). In order to estimate specific heat demand

coefficients for the non residential we aggregate the building functions listed on the input data to the functions describe on the VDI.

4 RESULTS

The computed results are depicted on Figure 1. On the figure we can see the simplified geometry of the individual buildings. The elements are colored by its specific heat demand. The pink line surrounding the computed elements is the geometry used to clip the input data set.

This particular part of the city contains many non residential buildings with a lower specific heat demand. On the figure we can identify some agglomerations of residential buildings with higher specific heat demand.

The implemented tool performs a geometry simplification that allows us to develop a quick prototype for the estimation of heat demand. We still need to quantify the loss in accuracy induced by the simplification of the building geometry. With these characteristics the simulation performance of the tool is very good.

The use of an application domain extension such as the Energy-ADE allows us to develop a tool that is potentially compatible with data generated by other developed also implemented this ADE. The gig advantage of using a template system as opposed to a hard coded data structure is its ability to cope with a rapid changing system. Through the templating system used in this tool we can populate a cityGML file with any type of data structure, even non valid data structures.

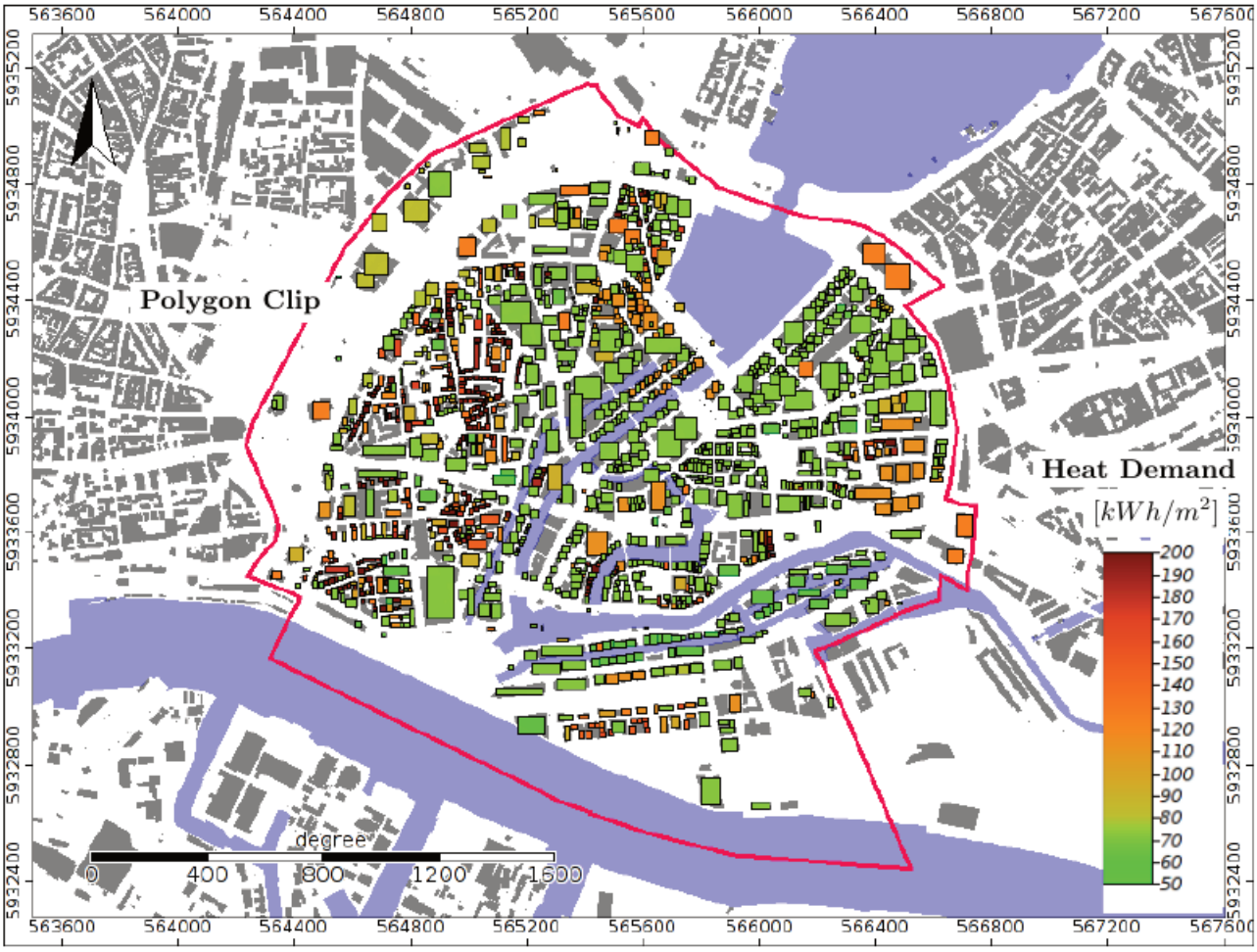

Fig. 1: Specific heat demand for selected urban area

5 REFERENCES

Diefenbach, N., Cischinsky, H., Rodenfels, M., & Clausnitzer, K.-D. (2010). Daten-basis gebäudebestand: Datenerhebung zur energetischen qualität und zu den modernisierungstrends im deutschen wohngebäudebestand (1. ed.). Darmstadt: Institut Wohnen und Umwelt (IWU) and Bremer Energie Institut (BEI).

Kim, E.-J., Plessis, G., Hubert, J.-L., & Roux, J.-J. (2014). Urban energy simulation: Simplification and reduction of building envelope models. Energy and Buildings, 84193 - 202. doi: http://dx.doi.org/10.1016/j.enbuild.2014.07.066

Loga, T., Diefenbach, N., & Born, R. (2011). Deutsche Gebäudetypologie: Beispielhafte Maßnahmen zur Verbesserung der Energieeffizienz von typischen Wohngebäuden.

ISBN 978-3-9504173-0-2 (CD), 978-3-9504173-1-9 (print)
Editors: Manfred SCHRENK, Vasily V. POPOVICH, Peter ZEILE, Pietro ELISEI, Clemens BEYER

Muñoz H., M. E. (2015). heat: R package to estimate heat demand of residential buildings. github.com/emunozh/heat. Retrieved from github.com/emunozh/heat

Muñoz Hidalgo, M. E. (2016 –in Press–). Construction of building typologies fromregional material catalog: Assessment of urban heat demand and the environmental impact of retrofit policies. Management of Environmental Quality - An international journal , -.

Nouvel, R., Bahu, J.-M., Kaden, R., Kaempf, J., Cipriano, P., Lauster, M., . . . Casper, E. (2015). Development of the citygml application domain extension energy for urban energy simulation. In 14th international conference of the international building performance simulation association (ibpsa). Retrieved from http://www.bs2015.in/files/BS2015 Proceeding.pdf

Nouvel, R., Kaden, R., Bahu, J.-M., Kaempf, J., Cipriano, P., Lauster, M., . . . Casper, E. (2015). Genesis of the citygml energy ade. In Cisbat 2015.

Verein Deutscher Ingenieure. (1994). Energieverbrauchskennwerte für Gebäude; Grundlagen: Richtlinien Energieverbrauchskennwerte für Gebäude (No. 3807). Berlin: Beuth