Enriching lower LoD 3D city models with semantic data computed by the voxelisation of BIM sources

van der Vaart, Jasper; Stoter, Jantien; Agugiaro, Giorgio; Arroyo Ohori, Ken; Hakim, Amir; El Yamani, Siham

DOI
10.5194/isprs-annals-X-4-W5-2024-297-2024

Publication date
2024

Document Version
Final published version

Published in
ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences

Citation (APA)
van der Vaart, J., Stoter, J., Agugiaro, G., Arroyo Ohori, K., Hakim, A., & El Yamani, S. (2024). Enriching lower LoD 3D city models with semantic data computed by the voxelisation of BIM sources. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *10*(4/W5-2024), 297-308. https://doi.org/10.5194/isprs-annals-X-4-W5-2024-297-2024

Enriching lower LoD 3D city models with semantic data computed by the voxelisation of BIM sources

Jasper van der Vaart, Jantien Stoter, Giorgio Agugiaro, Ken Arroyo Ohori, Amir Hakim, Siham EL Yamani

3D Geoinformation, Delft University of Technology, Delft, The Netherlands -
(J.A.J.vanderVaart, J.E.Stoter, G.Agugiaro, K.Ohori, S.Hakim, S.E.ElYamani)@tudelft.nl

Keywords: 3D city models, BIM, voxelisation, volume approximation, area approximation

Abstract

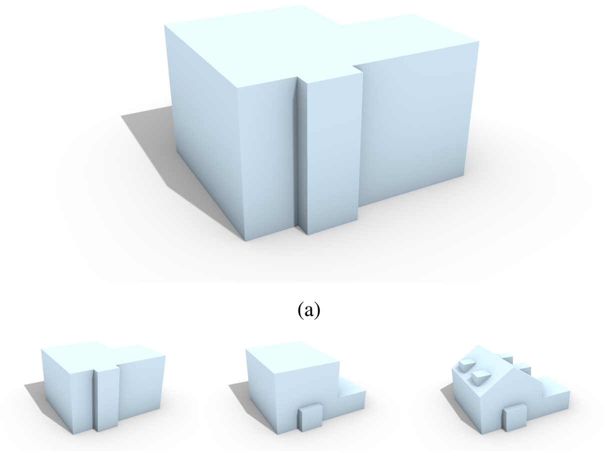
The role and adoption of 3D city models have been changing from a data endpoint to a centralised data source that is used for a variety of different analyses in different sectors. This change has not yet been fully completed and the transition process is still very noticeable at certain places. For example, data required for city-scale analyses are often missing, incorrect, or not stored in a standard way. A subset of these data (E.g. shell volume, shell area & footprint area) can be approximated from lower LoD shapes (LoD2.2 or lower) in the 3D city models. However, these models frequently simplify reality and therefore these approximations are not accurate. This paper proposes computing these data by voxelising Building Information Modelling (BIM) models representing the same buildings as the 3D city model. It is shown that a subset of these approximations (shell volume & footprint area) are more accurate than values computed from lower LoD shapes. Storing these data as attributes of the building models in 3D city models can improve the ease of use and the outcome of city-scale analyses. The computed values from BIM models can also be assigned to outputs of BIM to Geo conversions. This overturns the accuracy loss of the geometry caused by the conversion in which geometry is significantly generalised and simplified.

1. Introduction

The role and adoption of 3D city models have been changing at a rapid pace. 3D city models started initially as a data endpoint and were primarily used for visualization purposes. Now, they have evolved into a centralised data source used for a variety of different analyses for different application domains (Biljecki et al., 2015; Gröger and Plümer, 2012), such as taxation, energy consumption and production computations and city safety and comfort.

The change of 3D city models from a data endpoint to a data source has not yet been fully completed and the transition process is still noticeable. One of the places where this is apparent is in the lack of available attributes required for certain data analyses. For example, trivial numeric attributes such as the building volume, interior floor area, and number of storeys are required to run energy simulations at an urban scale (Rossknecht and Airaksinen, 2020). However, such attribute data are often missing in 3D city models. In the rare cases where it is present, it is often incorrect, or not stored in a standard way.

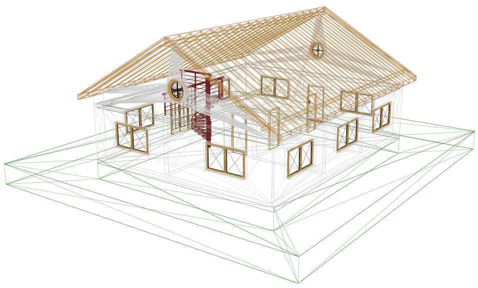
Such attributes can be approximated with the help of the shapes that represent the geometry of the buildings in the 3D city model. These implicitly stored geometric data, such as area and volume, are however not always accurate due to building shape simplification during the model creation. Almost every building shape in a 3D city model is simplified in relation to its real-world counterpart. Because it is often not known what the actual simplification entails, the impact of the simplification on those implicit values is also unknown (van der Vaart et al., 2023). This means that without detailed information regarding the utilised simplification methods, it is often not known how the simplified shapes deviate from reality, see Figure 1. Any value extracted from these shapes and the analyses utilising those values will thus also deviate by an unknown magnitude from reality.



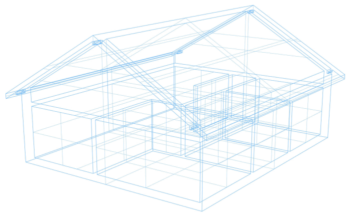
(a) (b) (c) (d)

Figure 1. Without knowledge of the original building, it is impossible to reconstruct the real shape of a building from its simplified form. The displayed model in Figure 1a could be an abstraction of a building with a single flat roof (1b), with multiple flat roofs at different heights (1c) or with a gable roof (1d). Example from van der Vaart et al. (2023)

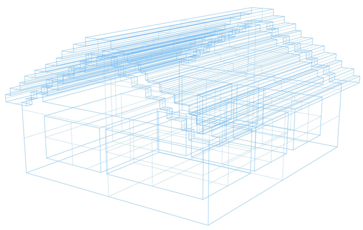
In this paper we investigate how relevant parameters can be computed from detailed Building Information Modelling (BIM) models in order to assign them as attributes to lower LoD building objects in 3D city models. Computing these parameters, which are frequently needed in urban data analyses, from BIM models provides several benefits. Firstly, these computed parameters can be assigned to outputs of BIM to Geo conversions. This would overturn the accuracy loss of the geometry caused by the conversion in which geometry is significantly generalised and simplified (e.g. small geometrical details and over-



(a)



(b)



(c)

Figure 2. Difference between a representation of a building as a BIM model (2a), as a member of a 3D city model (2b) and as a voxelised approximated object (2c). The BIM building model is constructed by volumetric objects. The 3D city building model and the voxelised model are represented by their shells.

hangs are eliminated in order to obtain valid outcomes). In addition, existing 3D City Models can be enriched with these computed parameters if BIM models become available for the same building. This would allow for possibly more accurate parameters than when derived from the building shapes available in 3D city models.

Converting and storing the geometry of a BIM model as a 3D city model would be the best solution, but this is challenging. BIM models are high-complexity models compared to building models in 3D city models, see Figure 2a and 2b. There is a major conversion step required to create suitable geometry for a 3D city model from a BIM source. Some attempts (including ours) are promising, but for large and complex BIM models these solutions are still computationally heavy and not very robust. BIM models have shown to be filled with errors created by either the users or the software that is being used (Arroyo Ohori et al., 2018; Noardo et al., 2021; van der Vaart, 2022). These errors range from the misuse of types and hierarchical structures to incorrect and missing geometry. These occurrences influence the outcome of the approaches presented in the past. A solution to avoid most of these issues would be the extraction of stored attributes. This is much easier since no complex conversion is required. However, these attributes are often also not reliable and do not always correctly reflect the actual modelled situation (Noardo et al., 2021; van der Vaart, 2022).

1.1 Computing parameters from BIM models

A strategy to overcome the BIM model complexity is voxelising the BIM model and calculating the required attribute data from the voxelised shape. Voxelisation can be seen as rasterising or pixelating a shape in 3D. Any voxel (**vo**lumetric pi**xel**) is considered as part of the shell when it touches the geometry of the BIM model. This enables the robust generation of a shape representing the building that is fairly easy to process and evaluate, see Figure 2c. Like the lower LoD building shapes in a 3D city model, these voxelised shapes will only approximate the original building shape. Thus, it will not yield precise values. However, if the voxel parameters are chosen properly, it could still approximate the value significantly better than a value approximation from a 3D city model. This will be investigated in this paper. Additionally, if the input variables of the voxelisation process are stored as metadata together with the computed values as attributes, the user is provided with the required information which reduces the ambiguity of the computed parameters to a minimum.

The literature covering the voxelisation of BIM models is limited. Krijnen et al. (2021) showed that voxelisation of a BIM model is possible for the reconstruction of thermal zones for energy simulations at a building scale. This approach looked promising but fundamental evaluation was missing, e.g. the effect of different voxel sizes. The size of the voxels used is a crucial factor in the voxelisation process. If the voxel size is enlarged the process is sped up, but it will also generally decrease the accuracy of the approximated results (Reitinger et al., 2003). Krijnen et al. (2021) showed that voxelisation of orthogonal buildings will perform better than more complex shaped buildings. Most other literature related to the voxelisation of constructions primarily covers the voxelisation process of buildings represented by point clouds. These methods and their results are outside the scope of our research since our focus is on the voxelisation of BIM models. Moreover, the two processes require different methods.

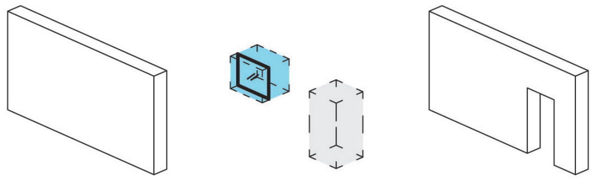
1.2 Research and overview of the paper

The goal of this research is to enrich lower LoD city models with semantic data computed by the voxelisation of BIM sources. Additionally, it is also evaluated if these data are more accurate than if it would be computed from the simplified shapes as available in 3D city models. This research will focus on two evaluations: the voxel size performance and the improvement of implicit 3D city model geometry-related parameters by deriving them from voxelised approximations of the BIM models.

From section 3 onwards the paper is split in these two parts. The first part, the voxel size performance, will investigate the "ideal" voxel size for the voxelisation of BIM models. The second part, the improvement of the 3D city model, will explore how the data computed with the help of the voxelised shape of BIM models can improve the implicitly stored geometric data as present in the 3D city building shapes.

2. Utilised Software

The IfcEnvExtracor (van der Vaart, 2024) is used as a tool to create the voxelised shape and to approximate the evaluation values. This is a software application that has been developed in ongoing research. The application converts BIM models to 3D



(a) (b) (c)

Figure 3. Visual representation of the simplification process of windows and doors from van der Vaart (2022). 3a displays the wall without voids applied. 3b displays the void objects (in dotted lines) that are related to the wall. The blue void has a window situated inside of it, the gray one is empty. 3c shows that only the gray void is applied to the wall. Resulting in simplified geometry at the location where a windows used to be.

city models by, among other things, extracting the outer shell of the building. It also allows for a voxelised (LoDV) output, although such shapes are not defined by common LoD frameworks. For this research, a new feature has been added that computes the required parameters (i.e. shell volume, shell area & floor area) and assigns these values as attributes to the output.

The primary goal of the voxelisation process of the IfcEnvExtracor is to aid in the detection of objects and faces that are part of the exterior shell. These elements are collected to form the outer shell of the building model. Due to this specific goal, the IfcEnvExtractor might follow different rules than a voxelisation process/tool that is purely developed to directly approximate building features. To clarify how the process might deviate, the voxelisation process of the IfcEnvExtractor is briefly described in the remainder of this section. It will also cover how the software has been extended with the computation of the additional parameters.

The voxelisation process of the IfcEnvExtractor for this study (including the computation of parameters) can be split into 3 steps: the input simplification, the voxelisation, and the final value computation.

2.1 Input simplification

The IfcEnvExtractor processes Industry Foundation Classes (IFC) files. IFC is an open and international standard for storing BIM data (Borrmann et al., 2018). IFC files are complex files, both semantically and geometrically. To ease the processing of these files, only a subset of the available IfcObjects are evaluated by the EnvExtractor: i.e. the space dividing objects (van der Vaart, 2022). This set of physical objects that are presumed to divide spaces, such as walls, floors, and windows.

Windows and doors, which are often geometrically complex in IFC, are pre-processed as described by van der Vaart (2022). Holes in walls are ignored if filled with door or window objects, see Figure 3. Additionally, windows and doors are simplified by replacing them with an oriented bounding box.

2.2 voxelisation

The first step of the voxelisation process consists of generating the 3D domain that is to be voxelised. This domain is constructed by creating an oriented smallest bounding box around the building that is being evaluated. The orientation of this bounding box results in a domain that in most cases has an optimised axis alignment for the voxelisation. This domain is then populated with a grid of voxels. For every voxel, it is evaluated whether it intersects with a space-dividing object. An intersection is detected if any part of the space-dividing object is inside the voxel.

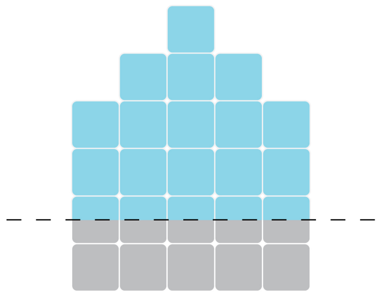


Figure 4. Side elevation of a voxelised shape that is being split at ground level. It can be seen that the voxels at ground level (the dotted line) are split and only the top part will be included in the following above ground computations.

After all the voxels are evaluated, the exterior of the shape is "grown". During this process, it is checked for each of the voxels if they have any surfaces that divide the voxelised interior shape and the exterior. If they do have any, these surfaces are also stored.

2.3 Final value computation

After these processes are completed, relevant parameters from the resulting voxelised shape are computed and stored as building attributes in the output 3D city model file. The parameters are related to the volume and areas of different parts of the building.

The computed values of the volumes and areas for shapes do not always add whole voxels. This is the case with evaluations that are split at ground level, e.g. above ground shell volume and area. The voxels that intersect with the ground plane are split and only the top part of the area and volume is used, see Figure 4.

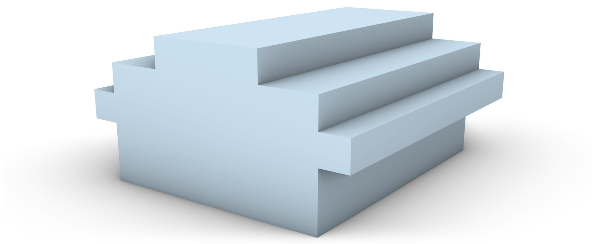
3. Method

This section presents the method that was developed to see how lower LoD city models can be enriched with semantic data computed by the voxelisation of BIM sources.

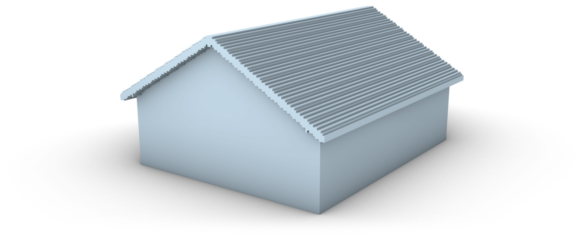
As mentioned in section 1, this study can be split into two parts: Voxel size performance (section 3.1) and enriching 3D city model data (section 3.2.

3.1 Voxel size performance

To be able to evaluate the performance of the voxelisation with regard to the impact on the accuracy of the calculated parameters, it is important to first establish the optimal, or "ideal", voxel size that is to be used. As mentioned in section 1, both the outcome and computing speed can be influenced by the used voxel size. This is due to the number of intersection tests that have to be executed. For a building that has a bounding box of $10 \times 10 \times 10$ m and a voxel size of $1 \times 1 \times 1$ m 1000 voxels are required. This means that 1000 voxels have to be evaluated for



(a)



(b)

Figure 5. The visual accuracy of this voxelised shape is a lot higher with small voxel sizes (5b) than with large voxel sizes (5a) when comparing it to the input model (2a).

a resulting shape with a resolution of $10 \times 10 \times 10$ voxels. If the same building is evaluated with a voxel size of $0.5 \times 0.5 \times 0.5$ m 8000 voxels have to be evaluated for a resulting shape with a resolution of $20 \times 20 \times 20$ voxels. The smaller the voxel size the more evaluations are required. However, smaller voxels will also yield a higher resolution shape that could resemble the original input more closely, see Figure 5.

The "ideal" voxel size is the optimal balance between accuracy and computing time. To evaluate the accuracy of the voxelised shape, we will look at two values computed from this shape, the above-ground shell volume and the above-ground shell area. The maximum deviation from the ground truth that will be considered acceptable is $\pm 5\%$. The "ideal" voxel size that falls within this range is approximated by processing six different IFC models for a range of different voxel sizes. The used models vary from a simple house to a realistic large apartment complex, see appendix 6.1. The used evaluation sizes are the depth or the width distances of the input models' footprint divided by incrementally enlarging steps. For example: width/1 $\rightarrow$ width/2 $\rightarrow$ width/4 $\rightarrow$ width/8 and so on.

To evaluate the accuracy of the voxelised shapes, a shape that is considered the ground truth has to selected. Using the input IFC file directly as ground truth is challenging, since a BIM model is constructed out of individual shapes that together make up the building. As mentioned in section 1 the building shapes in 3D city models are constructed differently than BIM models, see Figure 2. This makes them better suitable for calculations such as shell volume and shell area. A building shape in a 3D city model is a single shape representing the outer shell of the building. If interiors are included, they are also single shapes representing the voids inside the building shell. Values such as the total outer shell area and volume are thus fairly easy to compute from a building shape as included in 3D city models. Therefore, detailed LoD3.2 models are created based on the source BIM models and used as the ground truth. An LoD3.2 model is a very accurate representation of a BIM model, including overhang and minor details, see Figure 2b. In practice contemporary 3D city models only include building representations that do not include the basements or other subterranean constructions. So, for this study, the LoD3.2 models are manually created by tracing an above-ground subset of the original BIM model in 3D. This subset of objects is also filtered to primarily include the space dividing objects that are also evaluated by the IfcEnvExtractor. The manual modelling was done in Rhino 3D and this application is also used to compute the LoD3.2 model's volume and shell area.

3.2 Improvement of volume, footprint area and outer shell area data in 3D city models

Aside from the accuracy of the calculated values, the improvement of the data over the data in lower LoD models is also evaluated. As mentioned in section 1, the lower LoD shapes in a 3D city model implicitly contain a building's shell volume and area. It is necessary to evaluate if the calculated values from the BIM model are more accurate than these implicitly stored values.

This is evaluated by comparing the shell area, shell volume, and footprint area of the voxelised shapes to those of the LoD1.0, 1.2, 1.3 and 2.2 simplified representations of the same building in relation to the ground truth. The same six building models that were used in section 3.1 are also used for this evaluation.

The lower LoD models are created by the IfcEnvExtractor or, when this tool is unable to correctly generate them, created by hand in Rhino 3D. This was done by creating a shape that adhered to the geometry rules for LoD simplification of Biljecki et al. (2016) and also followed the approach used by Peters et al. (2022) and van der Vaart (2024). As with the LoD3.2 model that is used for the voxel size performance evaluation, the simplified LoD models only represent the above ground part of the building.

A summary of the shapes that were created per LoD:

- LoD1.0: Oriented smallest bounding box around the evaluated building.
- LoD1.2: Projected roof outline extruded upwards to the buildings highest z-height.
- LoD1.3: Flattened roof surfaces extruded downwards to footprint level.
- LoD2.2: Roof surfaces extruded downwards to footprint level.

Rhino 3D is used to compute the lower LoD models' shell volume, shell area, and footprint area.

4. Results & Discussion

As in section 3 the results section is split in two different subjects: Voxel size performance (section 4.1) and improvement of implicit 3D city model data (section 4.2).

4.1 Voxel size performance

Tables 1 and 2 show the performance of different voxel sizes for two of the models that were evaluated. These tables show the division steps described in section 3.1 as name, the resulting voxel size, the computing time, and the distance from the

Name	Voxel size (m)	Computing time (s)	Volume error (%)	Area error (%)
D/32	0.41	14.14	14.48	25.88
D/64	0.20	17.12	7.87	18.22
D/128	0.10	29.26	5.00	17.21
D/256	0.05	130.15	2.23	15.49
W/512	0.03	1026.69	1.07	15.00
W/32	0.34	15.14	13.18	19.48
W/64	0.17	17.16	7.77	15.81
W/128	0.09	38.39	3.91	16.18
W/256	0.04	202.89	1.62	15.41
W/512	0.02	1749.35	0.67	14.94

Table 1. Voxel size performance summary for the FZK haus model, see appendix 6.2.1 for the full table.

Name	Voxel size (m)	Computing time (s)	Volume error (%)	Area error (%)
D/32	1.32	80.62	29.92	5.89
D/64	0.66	93.76	16.03	16.13
D/128	0.33	279.4	7.41	22.38
D/256	0.16	1700.82	2.66	23.29
D/512	0.08	13489.48	0.84	22.10
W/32	1.15	75.66	30.03	3.84
W/64	0.58	100.84	13.57	17.70
W/128	0.29	346.08	7.27	21.33
W/256*	0.14	2289.62	-4.01	37.60
W/512*	0.07	20217.39	-6,59	37.88

Table 2. Voxel size performance summary for the APC model, see appendix 6.2.5 for the full corrected table. * shows the values where the voxels are “leaking”.

ground truth for both volume and area values. These tables are summaries of more extensive ones that can be found in appendix 6.2. This appendix also includes the results of the four other models that were evaluated.

From these tables and the tables in the appendix, it can be noted that the volume approximation from voxelised models can become very accurate compared to the ground truth data while the shell area approximation is unable to so to a similar degree. Regardless of the chosen voxel size, the resulting shell area is always significantly off. This is irrespective of whether the model is small, large, primarily orthogonal or not. The general trend that can be seen is that, when the voxel size is reduced, the area approximation does improve, as expected. The improvements however become weaker as the voxel size is increasingly refined and will in most cases never be within the +-5% error margin.

The exception to the observed area improvement stagnation might be the smiley model. The closest area approximation shows a 5.7% distance from the truth with no apparent signs of stagnation or fluctuation in the approximated values when reducing the voxel size. This is probably due to the shape of the model. It is the most box-like model out of the test set: it has a rectangular footprint, straight walls, there are no overhangs, and only a small subset of the roofs are angled. So, it can almost perfectly align with the voxel grid and be approximated very closely by a voxelised shape. Although this model has an almost ideal shape for voxelisation the approximated area was still not within the +-5% error threshold with a very small $5.6 \times 5.6 \times 5.6$cm voxel grid.

The results of the area approximation of the APC and the Almere model show some interesting behavior. In the Almere

Name	Voxel size (m)	Volume error (%)	Area error (%)
D/64	0.90	5.26	20.21
D/128	0.45	0.40	18.91
D/256	0.22	-16.67	83.16
D/64	0.90	10.27	15.34
D/128	0.45	5.86	13.21
D/256	0.22	2.69	12.69

Table 3. Comparison of a subset of the voxelisation of the Almere model with the IfcBuildingElementProxy elements ignored (top series) and included (bottom series).

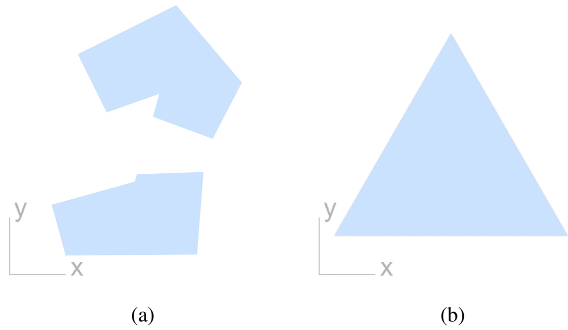


Figure 6. The footprint of the APC model (6a) and the Almere model (6b). The lower left X and Y axis are the voxelgrid axis. It can be seen that many edges are not, and cannot be, aligned to the voxelgrid.

model, the three smallest voxel sizes of the W (width-based sizes) and D (depth-based sizes) series show extreme value changes for both the volume and the shell area, see table 3. Unlike the other models refinement of the voxel size does not translate into improvement of the volume approximation. This is due to the voxels “leaking” into the building. The model has vent shafts covered with objects that are modelled as IfcBuildingElementProxy. With the default IfcEnvExtractor settings these are ignored during the voxelisation process. Consequently, the software will process these vent shafts as uncovered openings. When a voxel size is used that is smaller than the opening, the exterior could incorrectly grow through these openings into parts of the interior. When adding the IfcBuildingElementProxy as custom space dividing objects to the IfcEnvExtractor the outputted results reflect the expected values, see Table 3. Similar behavior can be seen in the APC model where voxels can grow into the interior due to a presumed faulty simplification of a door object. A small gap is left open between the simplified door and the wall through which the exterior can grow.

The results of the APC and Almere models show that it does not significantly matter for the accuracy of the volume approximation if the building shapes are properly axis aligned to the voxelisation axis. The software automatically orientates the models to be best aligned to the voxelisation axis. However, the APC and Almere models have footprint shapes that can not have every side aligned to the voxel grid, see Figure 6. The results still show a good performance that is similar to the performance of models that have a rectangular footprint. The APC model does show some extremely weak shell area accuracy compared to all other models. It is hard to evaluate if this is because of the complex footprint shape or because it is not axis-aligned. The APC model is the only model that shows this behavior.

Due to the mostly poor shell area approximation performance of the voxelised shapes, it has been decided to ignore these results when searching for the ideal voxel size. Determining the ideal voxel size will therefore be solely based on the volume approximation.

From Table 1, it can be seen that for a voxel size of about $0.1 \times 0.1 \times 0.1$ m or D/128, the volume error drops below 5% for the FZK Haus model. Similarly, W/128 also produces a volume error below 5%. The computing time for D/128 and W/128 can also be considered fairly reasonable. However, it can be seen that increasing a step (to /256) has a drastic effect on the time to compute.

Unlike the FZK Haus model, Tables 10 and 11 in appendix 6.2 show that choosing either depth or width does not always work. For both the Institute and the Smiley model the volume approximation of D/128 is far below the 5% volume error while the W/128 is above it. This is due to the shape of the building. The FZK Haus model has a somewhat uniform x and y domain size of the footprint. The Institute and Smiley models do not, since the width is more than two times larger than the depth (ten times for the Smiley model).

A solution for this could be to replace the depth and width terms with the shortest side (SH). In the case of the FZK Haus model that would be the width and in the case of the Smiley model that would be the depth. Both the SH/128 have a volume error below 5%. Applying this rule to other evaluated models, such as the APC and Almere models do not yield the desired results. Interestingly, the longest side (LO)/128 performs noticeably better compared to the SH/128 size for the Almere model.

A possible solution would be to refine the voxel size one step further: from SH/128 to SH/256. The resulting volume calculations will all fall within the +-5% error. But, for most models this will be overly precise and for large models this will be computationally heavy. The Almere model, which is the most complex and large model that was evaluated, is not a very large model compared to models representing highrise buildings. However, SH/256 takes over 40 minutes to process. Generally, the more objects and the more complexity an input model has, the slower the process becomes. SH/256 will thus not be a balance between accuracy and computing speed for models the size of the Almere model and larger.

Based on the evaluated models it can be concluded that there is not a singular rule that can determine the ideal voxel size for any model. This does not mean that the SH approach is without value. The results do show that if no details of the building are known, a voxel size that falls around SH/128 and SH/64 would be a good starting point. If details about the building model are known, the voxel size can be further refined. Based on the evaluations, three different voxel sizes can be recommended. For extremely small models representing a single small house a voxel size of $0.1 \times 0.1 \times 0.1$ m suffices. For small models representing small apartments or office buildings a voxel size of $0.15 \times 0.15 \times 0.15$ m suffices. For medium to large models representing large apartment or office buildings a $0.20 \times 0.20 \times 0.20$ m suffices. Table 4 shows the results using these defined voxel sizes for the six evaluation models. In theory, the ideal voxel size could keep growing with the size of the input model. The Almere model, does also perform well with even larger $0.40 \times 0.40 \times 0.40$ m voxels. It is however the only evaluated model of its size.

It should be noted that these values will not guarantee a volume approximation that falls within $\pm 5\%$ of the truth. For the evaluated models the results are favorable but the Smiley model does show a potential issue when looking at the full results. The W/512 voxel size is $0.1457 \times 0.1457 \times 0.1457$ m. This is a smaller voxel size than the selected $0.15 \times 0.15 \times 0.15$ m, but the resulting voxel volume approximation is performing worse. The approximation falls outside of the $\pm 5\%$ range with a 5.29% distance from the truth. This is potentially due to the relation between height of the building and the height of the voxels which can result in overestimation of the volume in models with a very large flat roof. The effect of this overestimation is reduced when a smaller voxel size is used. Although for the evaluated models the results comply with our theory in other models it could result in a volume approximation that is close to, but outside the $\pm\%$ value.

Source File Name	Voxel size (m)	Computing time (s)	Volume error (%)	Area error (%)
FZK Haus	0.1	29.25	3.92	16.12
Institute	0.15	91.72	2.97	8.97
Smiley west	0.15	68.00	4.42	7.29
Schependomlaan	0.15	160.32	2.63	13.35
APC	0.2	946.09	4.27	23.19
Almere*	0.2	3538.23	1.08	11.87
	0.4	684,01	2,81	12.65

Table 4. Voxel size performance summary for the evaluation models utilising the pre-defined voxel sizes. * Model has been evaluated with the inclusion of IfcBuildingElementProxy type elements as space dividing elements.

Model	Volume error (%)	Area error (%)
LoD1.0	53.06	19.48
LoD1.2	53.06	19.48
LoD1.3	53.06	19.48
LoD2.2	14.59	0.14
LoD3.2 (truth)	0	0
Voxel 0.1m	3.92	16.12

Table 5. Comparison of different LoD simplifications of the FZK Haus model and a voxelisation with a voxel size of $0.1 \times 0.1 \times 0.1$ m, see appendix 6.2.1 for the full table.

Model	Volume error (%)	Area error (%)
LoD1.0	153.84	22.33
LoD1.2	19.88	-17.25
LoD1.3	12.17	-16.54
LoD2.2	12.17	-16.54
LoD3.2 (truth)	0	0
Voxel 0.2m	4.27	23.19

Table 6. Comparison of different LoD simplifications of the APC model and a voxelisation with a voxel size of $0.4 \times 0.4 \times 0.4$ m and $1 \times 1 \times 1$ m, see appendix 6.2.5 for the full table.

4.2 Improvement of volume, footprint area and outer shell area data in 3D city models

Tables 5 and 6 show a summary of the performance of the voxelised values for different LoDs. The tables covering the other models can be found in appendix 6.2.

From these tables, it can be seen that the volume approxim-

Model	Approximation	Area error (%)
FZK	LoD2.2	19.17
	Voxel 0.1m	1.84
Institute	LoD2.2	21.60
	voxel 0.15m	1.72
Smiley	LoD2.2	5.26
	voxel 0.15m	2.29
Schependomlaan	LoD2.2	9.33
	voxel 0.15m	1.55
APC	LoD2.2	43.81
	voxel 0.2m	19.16
Almere	LoD2.2	6.80
	voxel 0.2m	0.98
	voxel 0.4m	2.54

Table 7. Performance of the footprint area approximation of the LoD2.2 3D city model building representation and the voxelised shape.

ation by voxelisation is significantly more accurate than the computed volume from the lower LoD simplified shapes. Even when the computationally efficient $0.4 \times 0.4 \times 0.4$ m voxel size is used for the Almere model, the computed volume is still considerably more accurate than the LoD2.2 simplification of the same building. As was shown in section 4.1, the area approximations by voxelisation are inaccurate. For all models, the voxelisation has been unable to improve the approximation of the shell area value of the LoD2.2 model. For the Smiley model it was even unable to improve the shell area of the LoD1.0 simplification.

This behavior can partially be attributed to the detection method of the EnvExtractor. Currently, the tool evaluates objects based on a selection of types, as mentioned in section 2.1. It is however not aware when these types are used in unpredictable ways. For example in the Smiley model there are garden fences modelled as IfcWall objects. The tool sees this as part of the building. The fencing is 2.00 m by 2.886 m with five regular vertical openings. If the model is voxelised with $0.15 \times 0.15 \times 0.15$ m voxels the resulting shape will have, in the ideal situation, an overestimation of 112 voxels per fence. This results is, per fence, a 0.378 m^3 volume difference, but a 9.90 m^2 difference in the shell area. This large overestimation of the area is not caused by the vertical openings in the fence creating more surface area. Even if the fence would not have had the openings the volume difference per fence would have been 0.756 m^3 while the shell area would have been 11.43 m^2. As can be noted by this example, incorrectly created protrusion adds substantially more area than volume error to the approximation.

Table 7 shows the performance of the footprint approximation for both the LoD2.2 building models and the voxelised building models. The table shows that for all six evaluated models, the voxel based approximation of the footprint area is more accurate than the LoD2.2 approximation. For the "ideal" voxel sizes, we can see that all these values are closer than $\pm 5\%$ from the ground truth. Even the $0.4 \times 0.4 \times 0.4$ m voxel approximation of the Almere model has been able to improve the value noticeably over the LoD2.2 model.

Although the 19.16% voxel footprint area error observed for the APC model is very large, it is still a notable improvement over the LoD2.2 footprint. This large over-approximation of the footprint area of the LoD2.2 model can be attributed to the large overhangs that the building has. The LoD2.2 models are roof-projected models. Therefore,the total exposed roofing area in these models is equal to its footprint. In case of an overhang, the lower LoD approximated footprint will be larger than the real situation. This behavior is visible in all six models, although not to the same extent as in the APC model. If a proper voxel size is chosen for a model, the voxel approximation will most likely be able to approximate the footprint shape and area better for any model that has a notable overhang.

The overestimation of the area when incorrectly including protrusion, mentioned in section 4.1, does not have a significant effect on the footprint area approximation. This is partially due to the 2D nature of the approximation. In this approximation only a single surface per voxel can be included. In the case of the Smiley model the fencing would add 0.315 m^2 per fence. This is considerably less than the 9.90 m^2 overestimation it adds to the shell area per fence. Additionally, in this particular case it does not add any overestimation to the footprint area because the fences are floating above the ground level. The footprint area is computed by only evaluating voxels that are at ground level. So the voxels intersecting with the fences are not included, but neither are other incorrectly detected protrusions that are not at ground level.

5. Conclusion

In this paper, we evaluated if values approximated from voxelised BIM models could improve the geometry-related parameters of building shapes in 3D city models. The goal of this was to see if it was valuable to store these BIM-based approximated values in the low LoD building shapes' attributes to aid further analyses.

To answer this question, the "ideal" voxel size had to be found first. This is a size that would yield sufficient accuracy while keeping computing speeds reasonable. The accuracy was tested by evaluating the shell volume approximation only. The shell area approximation proved to be too inaccurate to be used or included in this research.

Based on the used input models, it was concluded that for the evaluated cases no uniform size that fits all input models exist. This is why three different sizes were recommended. For small models a voxel size of $0.1 \times 0.1 \times 0.1$ m, for middle-large models a size of $0.15 \times 0.15 \times 0.15$ m and for large models a $0.20 \times 0.20 \times 0.20$ m. These values will most likely result in a shell volume approximation that falls within $\pm 5\%$ distance from the truth, but it can not be guaranteed.

It was also shown that the type of IFC objects that should be evaluated during the voxelisation process could not be picked blindly. If these are picked improperly, for example due to ambiguous modelling of the source IFC model, the results can be misleading.

Utilising the proper voxel sizes and types of IFC objects, it was found that the approximated values for shell volume and footprint area are noticeably improved over approximating these values based on the LoD2.2 (or lower) simplified building shapes. Therefore, storing the voxel approximated values as building objects' attributes in 3D city models, for example to BIM-to-Geo converted lower LoD models, would allow for improved analyses that require these parameters as input. Further research is needed to show how and to what scale these voxel-approximated values influence the subsequent analyses.

Acknowledgements

This project has received funding from the European Union's Horizon Europe programme under Grant Agreement No.101058559 (CHEK: Change toolkit for digital building permit).

References

Arroyo Ohori, K., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J., 2018. Processing BIM and GIS models in practice: experiences and recommendations from a GeoBIM project in the Netherlands. *ISPRS International journal of geo-information*, 7(8), 311.

Biljecki, F., Ledoux, H., Stoter, J., 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59, 25–37.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889.

Borrmann, A., Beetz, J., Koch, C., Liebich, T., Muhic, S., 2018. Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models. *Building information modeling: Technology foundations and industry practice*, 81–126.

buildingSMART, 2024. Sample-test-files. `https://github.com/buildingSMART/Sample-Test-Files/tree/master`. Accessed: 18-02-2024.

Gröger, G., Plümer, L., 2012. CityGML–Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71, 12–33.

IAI/KIT, 2021. Kit ifc examples. `https://www.ifcwiki.org/index.php?title=KIT_IFC_Examples`. Accessed: 18-02-2024.

Krijnen, T., El-Diraby, T., Konomi, T., Attalla, A., 2021. Thermal analysis of ifc building models using voxelized geometries. *Proceedings of the 38th International Conference of CIB W78*.

Noardo, F., Arroyo Ohori, K., Krijnen, T., Stoter, J., 2021. An inspection of IFC models from practice. *Applied Sciences*, 11(5), 2232.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2022. Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3), 165–170.

Reitinger, B., Bornik, A., Beichel, R., 2003. Efficient volume measurement using voxelization. *Proceedings of the 19th spring conference on Computer graphics*, 47–54.

Rossknecht, M., Airaksinen, E., 2020. Concept and evaluation of heating demand prediction based on 3D city models and the citygml energy ADE—Case study helsinki. *ISPRS International Journal of Geo-Information*, 9(10), 602.

van der Vaart, J., 2022. Automatic building feature detection and reconstruction in IFC models.

van der Vaart, J., 2024. Ifc_buildingenvextractor. `https://github.com/jaspervdv/IFC\_BuildingEnvExtractor`. Accessed: 18-02-2024.

van der Vaart, J., Stoter, J., Diakité, A., Biljecki, F., Arroyo Ohori, K., Hakim, A., 2023. Assessment of the LoD specification for the integration of BIM-derived building models in 3D city models. *Recent Advances in 3D Geoinformation Science*.

6. Appendix

6.1 Used models

Name	Source	Object count	Storey count	Width (m)	Depth (m)	Height (m)
FZK Haus	IAI/KIT (2021)	102	2	11	13	6.32
Institute-Var-2 (Institute)	IAI/KIT (2021)	896	5	44	19	12.20
Smiley-West-10 (Smiley)	IAI/KIT (2021)	972	4	75.12	7.28	8.32
Schependomlaan	buildingSMART (2024)	3505	4	21.78	23.42	13.38
Ascoli Piceno v1 (APC)*	ZWEI Ltd	1707	5	42.15	36.91	15.40
Almere BPA (Almere)*	LKSVDD Architecten	6760	4	57.49	66.39	16.00

Table 8. Summary of the evaluation models that were used.
* not open source models

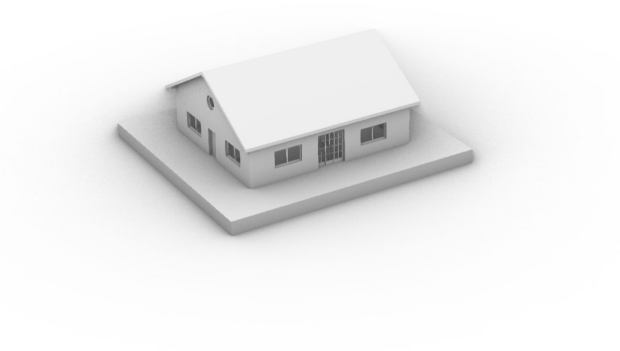


Figure 7. Visual representation of the FZK Haus model.

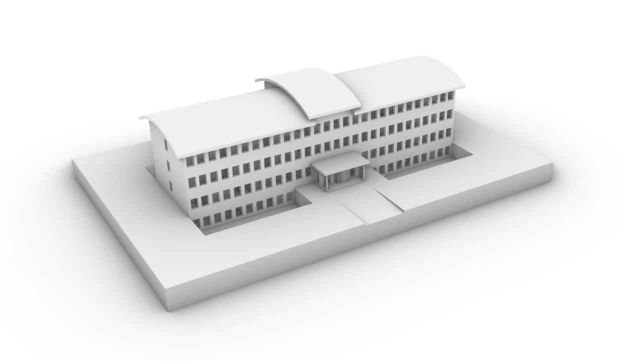


Figure 8. Visual representation of the Institute model.

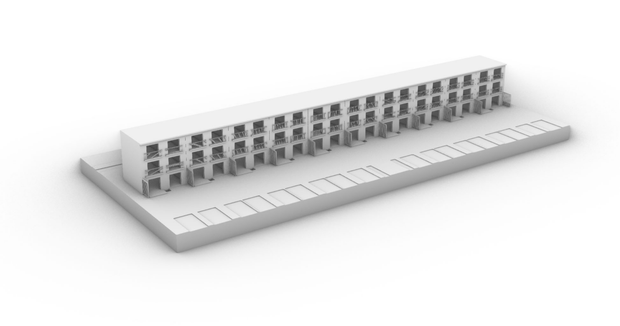


Figure 9. Visual representation of the Smiley model.

Figure 10. Visual representation of the Schependomlaan model.

Figure 11. Visual representation of the APC model.

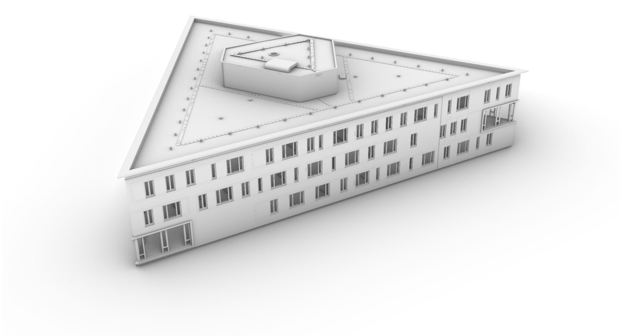


Figure 12. Visual representation of the Almere model.

6.2 Full results

6.2.1 FZK Haus

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	590,27	0,00	493,21	0,00		
D/1	13	12979,20	2098,86	3692,00	648,57	15,12	2
D/2	6,5	3227,90	446,85	1589,90	222,36	15,14	2
D/4	3,25	1674,16	183,63	1104,02	123,85	14,13	2
D/8	1,625	1094,67	85,45	840,53	70,42	14,14	4
D/16	0,8125	818,94	38,74	689,02	39,70	15,11	5
D/32	0,40625	675,75	14,48	620,85	25,88	14,15	10
D/64	0,203125	636,70	7,87	583,06	18,22	17,14	18
D/128	0,1015625	619,80	5,00	578,09	17,21	29,27	48
D/256	0,05078125	603,46	2,23	569,60	15,49	130,15	104
D/512	0,025390625	596,57	1,07	567,21	15,00	1026,69	236
W/1	11	11761,20	1892,51	3603,60	630,65	14,15	2
W/2	5,5	3254,90	451,43	1497,10	203,54	14,14	2
W/4	2,75	1259,16	113,32	867,35	75,86	14,15	2
W/8	1,375	1004,39	70,16	753,91	52,86	14,11	3
W/16	0,6875	785,59	33,09	642,64	30,30	15,16	5
W/32	0,34375	668,08	13,18	589,31	19,48	15,14	10
W/64	0,171875	636,12	7,77	571,18	15,81	17,16	20
W/128	0,0859375	613,33	3,91	573,03	16,18	38,39	61
W/256	0,04296875	599,82	1,62	569,20	15,41	202,89	123
W/512	0,021484375	594,21	0,67	566,88	14,94	1749,35	2455
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0	-	903,47	53.06	589,26	19,48		
LoD1.2	-	903,47	53.06	589,26	19,48		
LoD1.3	-	903,47	53.06	589,26	19,48		
LoD2.2	-	676,39	14,59	493,88	0,14		

Table 9. The FZK-Haus shell volume and area data.

6.2.2 Institute var 2

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	5731,45	0,00	2623,89	0,00		
D/1	19	45774,80	698,66	9992,10	280,81	45,45	2
D/2	9,5	18880,30	229,42	5477,70	108,76	44,42	2
D/4	4,75	13447,25	134,62	4409,42	68,05	44,43	3
D/8	2,375	8518,05	48,62	3397,79	29,49	45,46	4
D/16	1,1875	7118,22	24,20	3055,56	16,45	44,40	7
D/32	0,59375	6381,72	11,35	2824,94	7,66	44,42	12
D/64	0,296875	6022,71	5,08	2842,82	8,34	49,44	22
D/128	0,1484375	5913,47	3,18	2832,21	7,94	87,83	39
D/256	0,07421875	5824,10	1,62	2823,45	7,61	385,45	77
W/1	44	237256,80	4039,56	25995,20	890,71	45,36	2
W/2	22	54740,40	855,09	10784,40	311,01	45,39	2
W/4	11	25470,50	344,40	6532,90	148,98	44,42	2
W/8	5,5	12995,40	126,74	4424,20	68,61	44,34	2
W/16	2,75	9009,21	57,19	3458,40	31,80	44,40	4
W/32	1,375	7705,05	34,43	3053,81	16,38	45,41	6
W/64	0,6875	6648,20	16,00	2833,31	7,98	45,43	11
W/128	0,34375	6172,30	7,69	2820,62	7,50	47,39	19
W/256	0,171875	5953,47	3,87	2841,18	8,28	69,64	36
W/512	0,0859375	5844,35	1,97	2828,88	7,81	264,18	68
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0		10072,96	75,75	3190,17	21,58		
LoD1.2		7868,00	37,28	2824,17	7,63		
LoD1.3		6945,86	21,19	2684,79	2,32		
LoD2.2		6784,52	18,37	2543,35	-3,07		

Table 10. The institute shell volume and area data.

https://doi.org/10.5194/isprs-annals-X-4-W5-2024-297-2024 | © Author(s) 2024. CC BY 4.0 License.

6.2.3 Smiley west

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	4407,52	0,00	2833,64	0,00		
D/1	7,28	17774,07	303,27	6196,30	118,67	44,37	2
D/2	3,64	11199,89	154,11	4465,77	57,60	43,39	2
D/4	1,82	7631,47	73,15	3814,54	34,62	45,44	11
D/8	0,91	5955,13	35,11	3362,69	18,67	45,38	17
D/16	0,455	5207,27	18,15	3162,66	11,61	45,38	25
D/32	0,2275	4730,26	7,32	3147,24	11,07	55,46	53
D/64	0,11375	4475,65	1,55	3060,60	8,01	93,78	75
D/128	0,056875	4459,27	1,17	2995,56	5,71	553,58	102
W/1	75,12	817447,07	18446,65	55217,71	1848,65	44,36	1
W/2	37,56	295157,87	6596,69	32645,65	1052,08	46,39	2
W/4	18,78	56747,56	1187,52	11888,49	319,55	43,36	2
W/8	9,39	18084,98	310,32	6447,55	127,54	44,32	2
W/16	4,695	10636,86	141,33	4627,63	63,31	45,35	3
W/32	2,3475	7509,77	70,39	3948,88	39,36	43,40	9
W/64	1,17375	6002,22	36,18	3605,03	27,22	45,40	18
W/128	0,586875	5077,42	15,20	3414,17	20,49	45,42	23
W/256	0,2934375	4775,94	8,36	3193,19	12,69	48,49	44
W/512	0,14671875	4640,68	5,29	3117,32	10,01	75,64	70
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0		6702,71	52,07	3039,68	7,27		
LoD1.2		5422,02	23,02	3453,78	21,89		
LoD1.3		4565,45	3,58	2818,53	-0,53		
LoD2.2		4554,07	3,32	2800,41	-1,17		

Table 11. The Smiley shell volume and area data.

6.2.4 Schependomlaan

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	3616,79	0,00	1841,26	0,00		
D/1	21,78	57151,90	1480,18	10940,53	494,19	63,37	2
D/2	10,89	22118,61	511,55	5870,58	218,83	63,35	3
D/4	5,445	10050,53	177,89	3435,69	86,59	62,51	3
D/8	2,7225	6274,00	73,47	2635,12	43,11	63,58	5
D/16	1,36125	4749,70	31,32	2232,09	21,23	62,56	10
D/32	0,680625	4230,22	16,96	2163,06	17,48	63,58	28
D/64	0,3403125	3899,44	7,81	2160,98	17,36	71,64	69
D/128	0,17015625	3733,71	3,23	2128,38	15,59	128,23	142
D/256	0,085078125	3656,22	1,09	2176,77	18,22	746,20	323
W/1	23,42	47653,37	1217,56	9474,80	414,58	64,58	2
W/2	11,71	20615,24	469,99	5206,27	182,75	61,55	3
W/4	5,855	9556,52	164,23	3209,13	74,29	62,61	3
W/8	2,9275	6132,25	69,55	2631,88	42,94	62,58	5
W/16	1,46375	4876,38	34,83	2206,13	19,82	62,64	10
W/32	0,731875	4238,13	17,18	2102,22	14,17	62,58	23
W/64	0,3659375	3897,59	7,76	2110,52	14,62	68,66	54
W/128	0,18296875	3744,76	3,54	2096,81	13,88	128,18	122
W/256	0,091484375	3655,71	1,08	2152,96	16,93	567,14	278
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0		6755,23	86,77	2217,39	20,43		
LoD1.2		5341,17	47,68	2005,15	8,90		
LoD1.3		4387,10	21,30	1863,22	1,19		
LoD2.2		3907,15	8,03	1762,33	-4,29		

Table 12. The Schependomlaan shell volume and area data.

6.2.5 APC

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	9440,27	0,00	4534,76	0,00		
D/1	42,15	149236,29	1480,85	17728,29	290,94	75,77	2
D/2	21,075	55763,74	490,70	10621,80	134,23	73,69	2
D/4	10,5375	32528,85	244,58	7078,04	56,08	69,62	3
D/8	5,26875	17819,14	88,76	4915,94	8,41	67,63	3
D/16	2,634375	13778,02	45,95	4500,34	-0,76	78,98	6
D/32	1,3171875	12264,51	29,92	4801,83	5,89	80,62	21
D/64	0,65859375	10953,78	16,03	5266,13	16,13	93,76	80
D/128	0,329296875	10139,66	7,41	5549,62	22,38	279,41	226
D/256	0,164648438	9691,71	2,66	5590,81	23,29	1700,82	627
D/512	0,082324219	9519,27	0,84	5536,75	22,10	13489,48	1639
W/1	36,91	150239,75	1491,48	19028,58	319,62	72,60	2
W/2	18,455	49875,56	428,33	10178,86	124,46	72,65	2
W/4	9,2275	29613,62	213,69	6613,76	45,85	70,65	3
W/8	4,61375	20356,97	115,64	5508,24	21,47	71,65	4
W/16	2,306875	13880,98	47,04	4530,44	-0,10	71,62	8
W/32	1,1534375	12274,86	30,03	4708,71	3,84	75,66	26
W/64	0,57671875	10721,38	13,57	5337,19	17,70	100,84	91
W/128	0,288359375	10126,83	7,27	5501,88	21,33	346,08	261
W/256	0,144179688	9062,18	-4,01	6239,87	37,60	2289,62	874
W.512	0,072089844	8817,78	-6,59	6252,57	37,88	20217,39	2309
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0		23962,95	153,84	5547,31	22,33		
LoD1.2		11317,08	19,88	3752,34	-17,25		
LoD1.3		10588,78	12,17	3784,63	-16,54		
LoD2.2		10588,78	12,17	3784,63	-16,54		

Table 13. The APC shell volume and area data.

6.2.6 Almere

Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)	Computing time (s)	File size (KB)
32 (truth)	-	23511,82	0,00	6463,25	0,00		
D/1	57,49	371407,32	1479,66	32601,54	404,41	263,22	2
D/2	28,745	136047,84	478,64	19381,14	199,87	257,21	2
D/4	14,3725	55340,80	135,37	11643,90	80,16	259,17	3
D/8	7,18625	36850,37	56,73	9371,64	45,00	257,24	5
D/16	3,593125	29848,66	26,95	8089,95	25,17	258,18	10
D/32	1,7965625	27166,43	15,54	7687,95	18,95	263,22	21
D/64	0,89828125	25926,21	10,27	7454,51	15,34	299,58	44
D/128	0,449140625	24890,71	5,86	7317,30	13,21	552,61	142
D/256	0,224570313	24143,90	2,69	7283,17	12,69	2608,69	346
W/1	66,39	860638,65	3560,45	61014,80	844,03	271,32	2
W/2	33,195	140568,20	497,86	17284,50	167,43	261,15	2
W/4	16,5975	73061,60	210,74	13935,53	115,61	261,16	3
W/8	8,29875	45169,95	92,12	10281,47	59,08	259,17	5
W/16	4,149375	35849,50	52,47	8869,65	37,23	260,28	9
W/32	2,0746875	28098,19	19,51	7862,43	21,65	260,30	18
W/64	1,03734375	26267,88	11,72	7474,55	15,65	283,38	35
W/128	0,518671875	24313,39	3,41	7282,48	12,68	443,67	108
W/256	0,259335938	23930,33	1,78	7201,10	11,42	1752,23	294
W/512	0,129667969	23708,89	0,84	7246,03	12,11	13120,31	956
Name	Voxel size (m)	Volume (m^3)	Distance from truth (%)	Area (m^2)	Distance from truth (%)		
LoD1.0		61185,17	160,23	11599,66	79,47		
LoD1.2		30618,15	30,22	7012,16	8,49		
LoD1.3		24986,02	6,27	6552,72	1,38		
LoD2.2		24986,02	6,27	6552,72	1,38		

Table 14. The corrected Almere shell volume and area data.