A SWEEP-PLANE ALGORITHM FOR THE SIMPLIFICATION OF 3D BUILDING MODELS IN THE APPLICATION SCENARIO OF WIND SIMULATIONS

R. Piepereit[1,*], M. Deininger[2], M. Kada[3], M. Pries[1], U. Voß[2]

[1] Beuth Hochschule für Technik Berlin - University of Applied Sciences, Department II, Luxemburger Straße 10, 13353 Berlin, Germany - (rpiepereit, pries)@beuth-hochschule.de
[2] Hochschule für Technik Stuttgart - University of Applied Sciences, Faculty C, Schellingstraße 24, 70174 Stuttgart, Germany - (martina.deininger, ursula.voss)@hft-stuttgart.de
[3] Institute of Geodesy and Geoinformation Science (IGG), Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany - martin.kada@tu-berlin.de

KEY WORDS: 3D City models, Automated Processing, BRep, Quality, Meshing, Sweep, CAE, CFD.

ABSTRACT:

As the number of virtual 3D city models is steadily increasing, so are the possible applications that take advantage of them. 3D models can be used for applications that range from simple graphic visualizations to complex simulations, such as air flow and acoustic simulations. The geometric requirements needed for Computer Aided Engineering (CAE) and Computational Fluid Dynamics (CFD) increase the already very high complexity of processing 3D models. If there are too many small geometric details, mesh generation may fail. In addition it will create small grid cells that consequently lead to a high computation time. So far, the necessary simplifications have been performed in a time consuming manual process. To reduce the preprocessing time for the considered simulation topic, the simplifications and modifications have to be automated. In this paper we introduce a sweep-plane algorithm designed to automatically simplify virtual 3D models (e.g. CityGML) by removing geometry information unnecessary for numerical simulations. The algorithm will search for edges whose length does not reach a predefined threshold and dissolve them by sweeping nearby faces. As a result we obtain a simplified geometry that can be meshed properly. This algorithm serves as a general basis for the creation of future simplification algorithms that may even be applicable to any simulation necessary. For this paper, one of Stuttgart's city blocks was processed with the developed algorithm and then used in a wind simulation carried out with ANSYS Fluent.

1. INTRODUCTION

In recent years, the availability of virtual 3D city models has increased rapidly. Generating these models from 3D point cloud data (originating from aerial laser scanning and dense image matching) has reached a high automation rate. 3D city models can be used for a wide range of applications, many of which already exist or are currently in development (Biljecki et al., 2015). Because the technology for rendering large area models is already well developed, 3D city and landscape models have currently been used primarily for visualization and planning purposes (Kolbe, 2009; Coors et al., 2016).

The use of Computational Fluid Dynamics (CFD) to obtain velocity and pressure fields around buildings started in the 1980s, at a time when CFD had already been an established method in other applications, for example aircraft design (Blocken, 2014). Urban CFD, however, tries to answer quite different questions, such as pedestrian comfort, wind load on buildings, pollution dispersion and urban microclimate. In urban CFD a lot of parameters have to be selected in any case (e.g. the modeling equations and the turbulence model, the modeling of the atmospheric boundary layer (ABL), dimensions of the computational domain, the boundary conditions), but in consequence of the question to be studied this is true even more. Best Practice Guides, such as (Franke et al., 2007), give useful hints on which parameters to select. Many studies have been performed to evaluate the influence of these parameters. They started with isolated, cubical bodies and continued with several buildings (even though mostly generic) based on prior results and due to increasing computer power in the 1990s (Blocken, 2014). Studying smaller or larger parts of real cities began around 2000 (Toparlar et al., 2017). Usually geometric information is taken from 3D city models and the geometry for simulation is set up in CAD or CFD tools. Only few studies focus on the influence of geometric details on CFD results (Ricci et al., 2017; Lee et al., 2013). As the understanding of urban CFD increases and digital city models become more and more available, there is a need to develop automated procedures to set up the geometry for CFD.

Even though 3D city models may formally meet all quality criteria and requirements for visualization purposes (with or without healing), they might not do so for CFD simulations. The principal point for an optimized geometry for CFD simulations is, that the shortest edge length of the 3D city model defines the shortest grid cell length and a large number of grid cells increases computation time. Therefore the complexity of the geometry should be reduced. On the other hand, all details that physically have an influence on the solution have to be kept.

In order to evaluate the influence the number of geometric details has on the CFD-results, several aspects have to be taken into account: The first criterion is the ability of the meshing tool to generate a mesh. The second is the quality of the mesh itself, e.g. the number and the skewness of the cells. Finally, the third criterion is the quality of the physical result, which can be evaluated through comparison with wind tunnel data or in situ measurements. It should be noted that wind tunnel experiments rely on geometric simplifications also, as they use small scale reproductions of the original buildings.

For modeling virtual 3D city models, the data format CityGML

[*] Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W10-151-2018

has been established by many authorities. In CityGML, the geometry of buildings is defined by polyhedrons. Meshing tools of common simulation software use the polygon edges as boundaries. Short edges, however, can unnecessarily increase the number of grid cells. Furthermore, the risk of high skewness of cells inside the mesh belonging to e.g. inflation layers increases. This could lead to numerical instability in the system of equations to be solved during CFD simulations. For this reason, narrow polygons should be avoided.

In this paper, we introduce a sweep-plane algorithm for the automatic simplification of 3D building models to reduce the number of unnecessarily short edges that interfere with the simulation. Edges whose length is below a certain threshold, are eliminated by pushing faces along their normal direction. Before a sweep is executed, it needs to be verified whether or not it is permitted, thereby ensuring that the resulting models are still valid with regard to their topology and geometry (e.g. the displacement of surfaces must not lead to intersections). In addition, the difference in shape between the optimized building model and the original one should be kept minimal (for this paper the buildings volume is the deciding factor). In order to achieve this, the displaced surface may be combined with coplanar ones and swept back into the opposite direction.

2. RELATED WORKS

In research and literary contributions the (automated) processing of 3D city models for simulations went mostly unheeded. Processing models for visualization, on the other hand, has been given far more attention. Especially for visualization, it is often necessary to simplify more complex models to generate models with different levels of detail (LoD).

Looking at 3D visualizations of city models, objects that are considered to be more important, because they are closer to the observer or of special interest (e.g. churches, castles), are depicted in greater detail, than objects that are further away or of less interest. As presented in this section, providing a variety of processed models and therefore various levels of detail, will lead to a gradual and especially smooth transition without a so-called pop-up effect while zooming in or out.

One approach to automatically simplify 3D building models is described in Forberg (2007). Inspired by the scale space theory from image analysis the objective here is to move parallel facets towards each other until they collide and can be merged. In this way small 3D elements can be removed or gaps closed. Seeing that this algorithm depends on facets that are parallel to each other, it is only suitable for orthogonal building models.

Fan and Meng (2012), however, aim not only at simplifying parallel or orthogonal geometries, but also 3D building models of any complexity. The characteristics of a buildings roof and wall structures are maintained as much as possible. However, they are simplified independently of one another. After the simplification of the ground plan in the first step, Fan and Meng generalize the roof structure of the building in the second step. Finally the 3D model is reconstructed by combining the processed ground plan and roof structure.

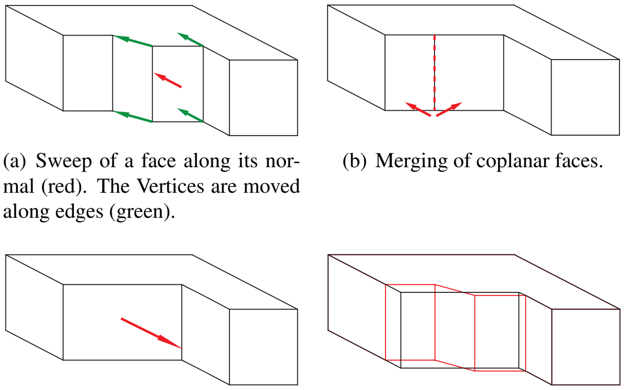
Even though these algorithms primarily aim at processing 3D city models for better visualization, some of them could also be applied to the preprocessing for simulations. However, the specifications for these the use of 3D city models used for simulations often differ from those required for pure visualization purposes and the algorithms may need to be adapted.

In a similar way, the sweep plane idea presented in (Kada et al., 2016), was adopted for this paper and adapted according to the specifications necessary for simulation purposes.

An approach for the simplification of virtual city models is presented in Piepereit et al. (2016). Here multiple narrow polygons, that represent a round surface, are replaced by freeform surfaces via a coons algorithm. While this approach is promising for round surfaces, it is not for offsets and other protrusions.

3. SIMPLIFICATION OF CITY MODELS

For the application of the sweep-plane algorithm, the CityGML model is used as the starting point. It is converted into a CAD BRep[1] data structure. The algorithm iteratively eliminates edges with a length below a predetermined threshold ϵ by moving faces, that are connected to them in one vetex only, along their normals. The following section will give an overview of the mechanics of the algorithm. For the building to be simplified F is defined as the set of all its faces and E as the set of all its edges. For a better understanding figure 1 is provided below and referred to in the following section.



(a) Sweep of a face along its normal (red). The Vertices are moved along edges (green).

(b) Merging of coplanar faces.

(c) DeSweep of the merged faces.

(d) Simplified Building model in comparison to the original one.

Figure 1. Sweep operation applied to a simple building model.

The following pseudo-code, illustrates a simplified version of the developed sweep-plane algorithm. For each building it iterates the following loop until all edges of the building are shorter than ϵ or no more faces can be swept:

```
Algorithm : Sweep-Plane Algorithm
Input  : Building with Faces $F$ and Edges $E$
Output : Modified Building
while $e \in E \,|\, lenght(e) < \epsilon$ do
    $F' = \{f \in F \,|\, u, v \in e, u \in f, v \notin f,$
      $isSweepPossible(f, dist(f, e))\,\}$
    $f' \leftarrow any(F')$
    $\sigma(f') \leftarrow Sweep(f', dist(f', e))$
    $f^* \leftarrow MergeFaces(\sigma(f'))$
    $DeSweep(f^*, dist(\Delta V))$
end
```

[1] The established data structure in CAD systems.

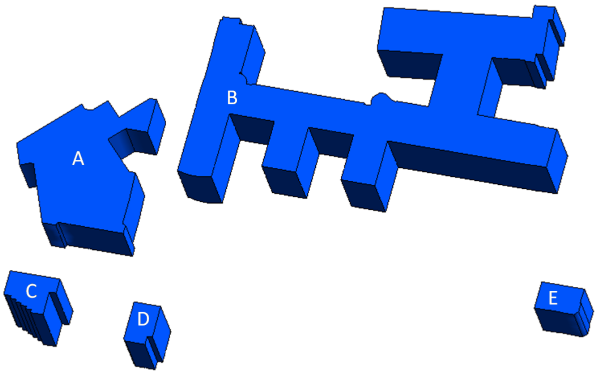


Figure 2. A part of a building block in Stuttgart.

The algorithm starts with searching for an edge e with a length $< \epsilon$ (line 1). If the sweep plane algorithm finds no such edges, it will terminate for this building and the optimization of the next building begins (should one exist).

In line 2, the set F' consists of faces that share a vertex u with e but not e itself (i.e. e is not part of the boundary of face f') and contains only faces for which a sweep is possible. This is the case, if a sweep does not result in any intersections and no edges will be inverted. If no faces meet these criteria, the sweep-plane algorithm will start with line 1 again. To avoid an endless loop, a flag is set, so that the just selected edge is not available for the next iteration.

A face is selected out of the aforementioned list in line 3. At the time of writing and publishing this paper it has not been investigated, what kind of influence the order of sweeps has on the final appearance of the building. Therefore it should be mentioned here, that the first face is picked off of the list, as the choice of neither e nor f' are of any importance at this point.

The actual sweep takes place in line 4. The edge e is supposed to be eliminated by pushing the face f' (consequently the length of e degenerates to zero). In order to avoid a rotation of f', it is pushed along its normal vector $n_{f'}$. The vertices $U' = \{u' \in U \,|\, u' \in f'\}$ of f' are not necessarily moved along the face normal but along the edges $E' = \{e' \in E \,|\, e' = (u', v'), u' \in f', v' \notin f'\}$ toward v'. The new position $\sigma(u')$ of the swept vertex u' is then determined by

$$n_{f'} \cdot \overrightarrow{uv} = n_{f'} \cdot \overrightarrow{u'\sigma(u')}.$$

Note that if $\sigma(u') = v'$ than the edge e' degenerates to a point and can therfore be deleted.

Figure 1a shows the displacement vectors of the individual vertices. The building's shape after the sweep is depicted in figure 1b. As can be seen here, two edges were eliminated as a result of the sweep. Additionally, one face degenerated into a line and could therefore be eliminated as well.

As coplanar faces may occur after every sweep, the algorithm checks for this occurrence in line 5 and merges the coplanar faces and eliminates redundant edges where necessary (compare figure 1b). The swept face is moved back in the opposite direction in line 6, in order to minimize the model's difference in volume

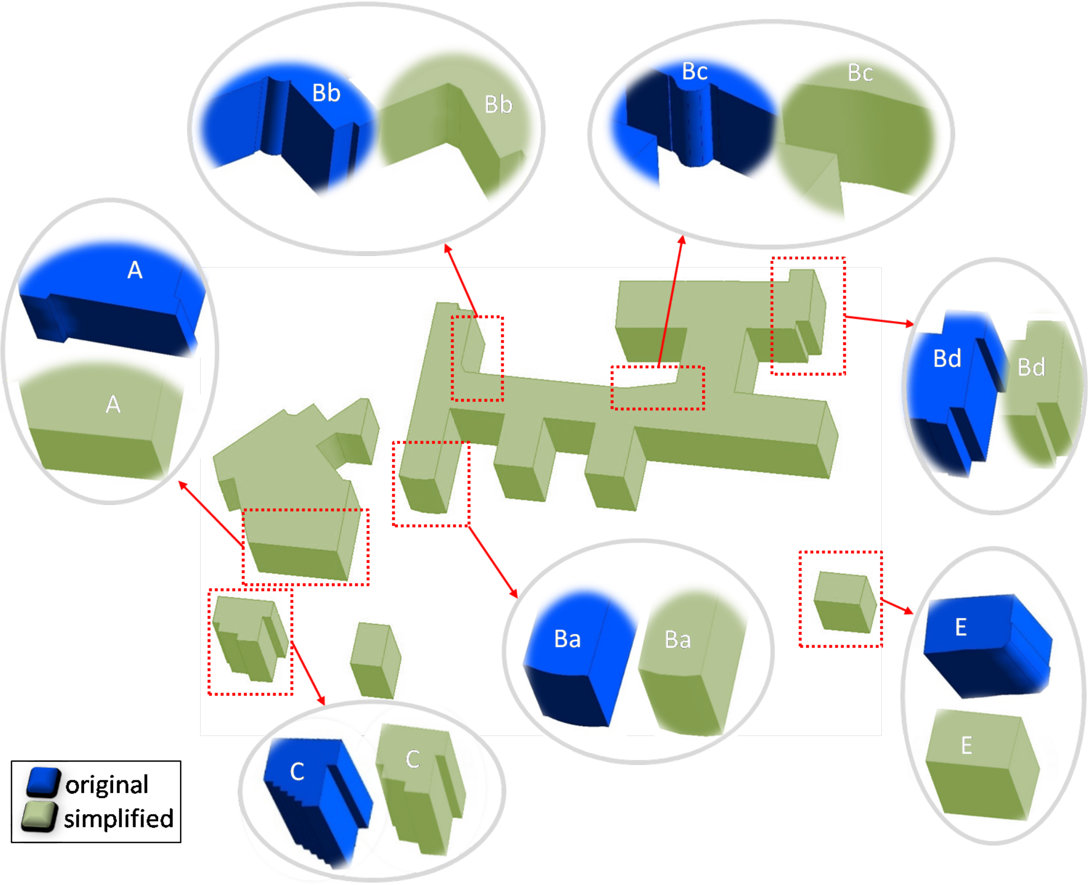


Figure 3. Optimized model of a building block in Stuttgart including details, before (blue) and after optimization (green).

from before and after the sweep (compare. figure 1c). The final version of the processed model is depicted in figure 1d.

4. EXPERIMENTS

In order to validate the proposed approach, a building complex located in the city of Stuttgart (see figure 2) has been processed with the sweep-plane algorithm. In this section the results are presented.

Figure 3 shows part of a building block from the Stuttgart city model, which was simplified with the sweep-plane algorithm and will be used for the wind simulation in section 5. A comparison of individual parts of the model before and after beeing processed is hightlighted in the zoom display. As can be seen, offsets, bulges (e.g. figure 3A, 3B and 3E) and corner offsets (figure 3E) were removed. The vertical steps of building C have been reduced so that no edge is smaller than the specified 2 m (figure 3C). Especially when round faces occur, e.g. in building B (see figure 2), they are usually represented by a large number of connected narrow polygons. As depicted in figure 3, these curves have been extremely simplified (figure 3.Ba) or completely dissolved (figure 3.Bb and 3.Bc) by the sweep-plane algorithm.

Table 1 summarizes the results obtained for each individual building. The second column specifies the number of edges of the original building model with a length smaller than the specified threshold of 2 m. After the optimization, this value is zero for each building and is therefore not listed in the table. The last two columns show the lengths of the smallest edge for each building model before and after the optimization.

	nr. of E	min. L_E (orig.)	min. L_E (simp.)
A	6	1.24	2.33
B	68	0.13	2.02
C	14	0.16	2.19
D	4	1.13	9.86
E	8	0.16	10.06

Table 1. Comparison of original and simplified building geometries: number of edges with length < 2 m (original), minimal edge length (original) and minimal edge length (simplified).

5. WIND SIMULATION

To simulate wind around the building block in Stuttgart (see section 3, figure 2) a suitable mesh (section 5.1) and CFD simulation setup (section 5.2) is assembled.

5.1 CFD setup and meshing

In order for the wind simulations to be effective, a cuboid space around all buildings (see figure 4) with a height of $6 \cdot H$, where H is the maximum height of buildings, as well as a width of $2 \cdot 6 \cdot H$ and a length of $(6 + 15) \cdot H$ (in main wind direction) is recommended by Franke et al. (2007).

For the selected region in Stuttgart a cuboid aligned with the main wind direction (compare windrose Stuttgart (City of Stuttgart, 2018)) with the space dimensions $1.0 \times 0.8 \times 0.35$ km^3 (length $\times$ width $\times$ height) has been used. The resulting air volume is cut with the building geometries as well as the solid ground (flat plane or terrain) to obtain the simulation domain for the CFD simulation. The wind is triggered by the atmospheric boundary layer (ABL - a velocity profile with a parabolic shape, (Tominaga et al., 2008)) as plotted in figure 4 (velocity shown as colored and black dashed lines at the inlet of the domain).

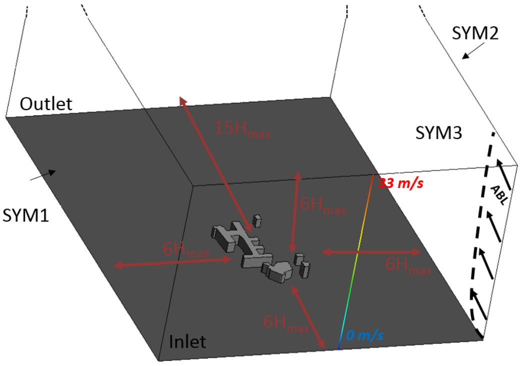


Figure 4. CFD simulation domain. Velocity on vertical line: ABL at inlet.

To resolve turbulent air flow near building walls (e.g. Sachs (1978)) the Reynolds Averaged Navier Stokes equations (RANS; compare Franke et al. (2007)) are solved within each cell depending on its neighboring cells.

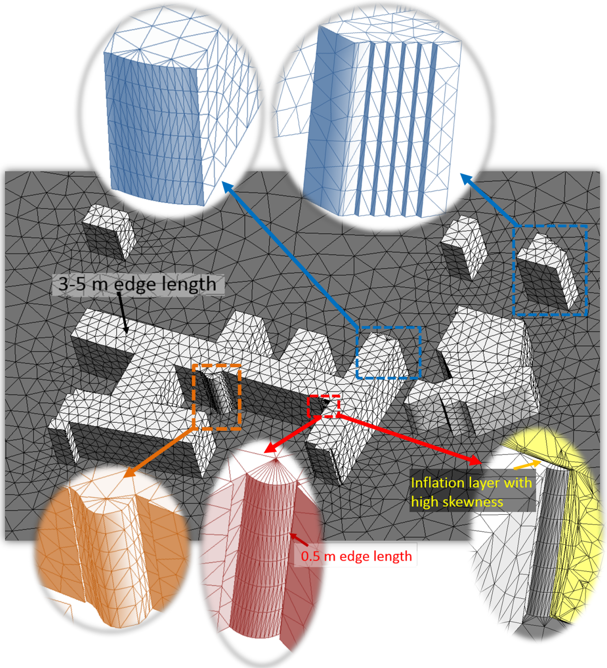


Figure 5. Original geometry: Boundary mesh of buildings. Suitable mesh with short edges (blue). Not suitable mesh due to high skewness of attached inflation layer (orange and red). Cross-section with inflation layer with to high skewness (yellow).

The resulting boundary for the original geometry mesh is depicted in figure 5. The surface mesh on the original geometry of building B (compare figure 5: orange and red regions of cylindrical parts) is valid. But, especially for the red region, the angles are quite small, which leads to the skewness beeing too high (0.96) inside of the mesh (compare figure 5: yellow region). During the CFD wind simulation in these regions the solver is not able to converge the turbulent variables.

To build a suitable mesh, it is recommended in Franke et al. (2007) to use a minimum of 10 cells per building length and five special hexaeder cell layers (inflation layers: boundary attached cells in yellow plane in figure 5 and 6) to capture the air- boundary interactions. In line with this, for the building block in Stuttgart a minimal cell length of 1 m and a minimal edge length of 2 m are used. Consequently, the sweep-plane algorithm (see section 3) is applied to the original building block in Stuttgart with a treshold $\epsilon = 2$ m for the minimal edge length. The mesh of the simplified geometry has a minimal cell length of 1 m. After the geometry simplification the mesh quality (minimal edge length, maximal skewness, maximal face size and number of cells) is much higher (compare Table 2) and the mesh, depicted in figure 6, is suitable for CFD simulation of the wind.

Building geometry	original	simplified
max. skewness	0.96	0.8
min. edge length [m]	0.14	2.03
max. face size [m^2]	62.2	64.9
number of cells	$119,700$	$114,400$

Table 2. Comparison of mesh properties for the original and the simplified building: minimal edge length, maximal face size and number of cells.

The mesh for the sweep-plane algorithm simplified building geometries is much smoother with a minimal edge length of 2 m and the inflation layers are well established (see figure 6). The maximal face size is higher and the number of cells is lower than in the original mesh.

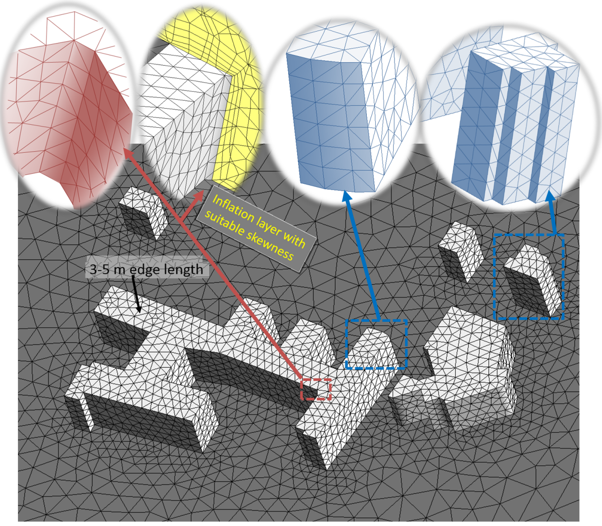


Figure 6. Simplified geometry (sweep-plane algorithm, Sec. 3): Boundary mesh of buildings. Suitable mesh with edges length ≥ 2 m (blue and red). Cross-section with suitable inflation layer (yellow).

To simulate the windfield around the building block, the ABL with a velocity of 1.5 m/s at a height of 10 m (compare measurements City of Stuttgart (2018)) is used at the inlet of the simulation domain (for details see e.g. Blocken et al. (2007)). On the left and right hand side (SYM1 and SYM2) as well as on top (SYM3) of the simulation domain so called symmetric boundary conditions are established to ensure a reduced influence on the flow inside the domain (compare figure 4). As outlet boundary condition the atmospheric pressure is pretended.

5.2 CFD results

As an example for the CFD simulation results, the wind velocities at pedestrian level for the simplified building block are depicted in figure 7. After the establishing phase of the air flow upstream the buildings, the air strikes the buildings and is deflected until it reaches the outlet.

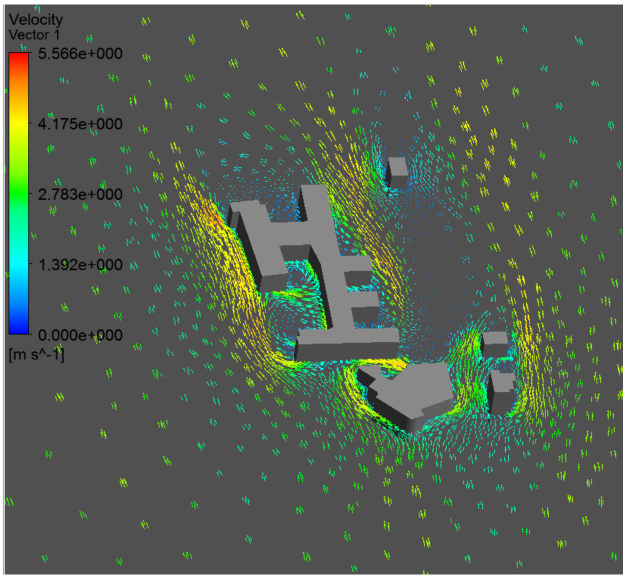


Figure 7. Example of wind velocity at pedestrian level for a building block in Stuttgart. ABL: 1.5 m/s at 10 m height.

6. CONCLUSION

In this paper, we introduced a sweep-plane algorithm with the aim for to automatically simplify virtual 3D City models. The algorithm thereby iteratively eliminates edges that are shorter than a given threshold. It has been successfully tested on several simple building models without features like doors, windows or chimneys. Starting with building models in the CityGML format, the models were simplified and then used in ANSYS FLUENT. For this paper part of a building complex in Stuttgart has been processed and used for a wind simulation. For the CFD simulation, the processing of the 3D city model with the sweep-plane algorithm is a good starting point for further development. It should focus on rising the amount of processed buildings, e.g. complete urban districts, including the interaction of buildings (e.g. to close or enlarge gaps between buildings smaller than a certain threshold and reduce complexity by merging buildings). Extensive tests with building models of a more complex geometry must be performed to fully understand the limitations of the algorithm. Combining the sweep-plane method and other algorithms, such as the coons approach in Piepereit et al. (2016), may be one approach to improve the simplification of virtual city models.

ACKNOWLEDGEMENTS

This work has been developed in the project iCity. The project iCity (Funding number: 03FH9I01IA) is supported by the German Federal Ministry of Education and Research (BMBF). The authors are responsible for the content of this publication. They thank Prof. Dr. Volker Coors for the support and the valuable input during the writing process of this publication. The authors further gratefully acknowledge the Stadtmessungsamt Stuttgart for the data of 3D citymodel of Stuttgart.

References

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

Blocken, B., 2014. 50 years of computational wind engineering: past, present and future. *Journal of Wind Engineering and Industrial Aerodynamics* 129, pp. 69–102.

Blocken, B., Stathopoulos, T. and Carmeliet, J., 2007. CFD simulation of the atmospheric boundary layer: wall function problems. *Atmospheric environment* 41(2), pp. 238–252.

City of Stuttgart, 2018. Stadtklima: Messstation Stuttgart-Mitte: https://www.stadtklima-stuttgart.de/index.php?klima_messdaten_station_smz.

Coors, V., Andrae, C. and Böhm, K., 2016. *3D-Stadtmodelle: Konzepte und Anwendungen mit CityGML.* VDE Verlag GmbH.

Fan, H. and Meng, L., 2012. A three-step approach of simplifying 3D buildings modeled by CityGML. *International Journal of Geographical Information Science* 26(6), pp. 1091–1107.

Forberg, A., 2007. Generalization of 3D building data based on a scale-space approach. *ISPRS Journal of Photogrammetry and Remote Sensing* 62(2), pp. 104–111.

Franke, J., Hellsten, A., Schlünzen, H. and Carissimo, B., 2007. Best Practice Guideline for the CFD Simulation of Flows in the Urban Environment: COST Action 732 Quality Assurance and Improvement of Microscale Meteorological Models.

Kada, M., Wichmann, A., Filippovska, Y. and Hermes, T., 2016. Animation Strategies for Smooth Transformations Between Discrete LODs of 3D Building Models. XLI-B2, pp. 413–420.

Kolbe, T. H., 2009. Representing and exchanging 3D city models with CityGML. In: *3D geo-information sciences*, Springer, pp. 15–31.

Lee, D., Pietrzyk, P., Donkers, S., Liem, V., van Oostveen, J., Montazeri, S., Boeters, R., Colin, J., Kastendeuch, P., Nerry, F., Menenti, M., Gorte, B. and Verbree, E., 2013. Modeling and observation of heat losses from buildings: The impact of geometric detail on 3D heat flux modeling.

Piepereit, R., Schilling, A., Alam, N., Wewetzer, M., Pries, M. and Coors, V., 2016. Towards automatic processing of virtual city models for simulations. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*.

Ricci, A., Kalkman, I., Blocken, B., Burlando, M., Freda, A. and Repetto, M., 2017. Local-scale forcing effects on wind flows in an urban environment: Impact of geometrical simplifications. *Journal of Wind Engineering and Industrial Aerodynamics* 170, pp. 238 – 255.

Sachs, P., 1978. Wind forces in engineering, 2nd edition. *Pergamon Press Oxford - New York - Toronto - Syndey - Paris - Frankfurt.*

Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M. and Shirasawa, T., 2008. AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings. *Journal of wind engineering and industrial aerodynamics* 96(10-11), pp. 1749–1761.

Toparlar, Y., Blocken, B., Maiheu, B. and Van Heijst, G., 2017. A review on the CFD analysis of urban microclimate. *Renewable and Sustainable Energy Reviews* 80, pp. 1613–1640.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-4-W10-151-2018