# Delft University of Technology

## Integration of GIS and CAD data to perform interactive preliminary environmental analyses at district scale

Tsai, Bing-Shiuan; Huizer, Lars; Giampaolo, Michele; Monté, Sérénic; Gong, Sicong; Garcia, Gabriel; Agugiaro, Giorgio

**Citation (APA)**
Tsai, B.-S., Huizer, L., Giampaolo, M., Monté, S., Gong, S., Garcia, G., & Agugiaro, G. (2024). Integration of GIS and CAD data to perform interactive preliminary environmental analyses at district scale. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*(4/W10-2024), 169-176. https://doi.org/10.5194/isprs-archives-XLVIII-4-W10-2024-169-2024

**Important note**
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

# Integration of GIS and CAD data to perform interactive preliminary environmental analyses at district scale

Bing-Shiuan Tsai[1], Lars Huizer[1], Michele Giampaolo[1], Sérénic Monté[1], Sicong Gong[1], Gabriel Garcia[2], Giorgio Agugiaro[1]

[1] 3D Geoinformation group, Delft University of Technology, Faculty of Architecture and the Built Environment, Department of Urbanism, Julianalaan 134, 2628BL Delft, The Netherlands,
{b.tsai, l.c.huizer, m.giampaolo, s.g.m.monte, s.gong}@student.tudelft.nl, g.agugiaro@tudelft.nl
[2] Royal HaskoningDHV, The Netherlands,
gabriel.garcia@rhdhv.com

**Keywords:** CAD, GIS, data integration, Grasshopper, solar analysis, wind analysis.

## Abstract

This paper describes the proposed methodology, the implementation, and the experience resulting from the further development of a tool, embedded in Rhinoceros/Grasshopper, that allows to perform preliminary environmental analyses at district scale in the case of a new planned building. The CAD-based parametric 3D model of a "new" building, generated in Grasshopper, is enriched with and embedded into a 3D urban scene of the block/district where it is planned to be built. The resulting 3D scene is then used to perform shadowing, solar and wind analyses that are used by architects and engineers in their preliminary development phases of the project. The work stems from a preliminary analysis in terms of data and software requirements carried out between practitioners from both the GIS and AEC domain.
More in detail, a series of modules in Grasshopper have been developed that allow to import GIS "surrounding" data at district scale (e.g. buildings, terrain) and to blend them with the "new" building model, in order to perform environmental analyses in (near) real time while the designer interactively changes the design parameters of the building and its position. The paper presents the results and discusses the inherent limitations.

## 1. Introduction

With the current high speed and scale of urbanisation, there is a growing demand for affordable housing – together with all other aspects that are tightly related to it: infrastructure for transportation, utility networks, etc. For this reason, integrated planning is increasingly playing a crucial role as the impacts of a new construction project should be investigated, evaluated and minimised from the very early stages of the design process (Josuf et al., 2017; Agugiaro et al., 2020). However, there still exists a "scale-dependent" dichotomy between the different disciplines involved in the different types of analyses (Ohori et al., 2017; Noardo et al., 2019). For example, a new building is generally planned and designed by practitioners (architects, engineers, etc.) using tools from the AEC (Architecture, Environment and Construction) domain that, traditionally, follow the CSG (Constructive Solid Geometry) modelling paradigm and use a local coordinate system. On the other hand, in order to estimate the impacts of the new building in the urban context (e.g. at block or district level), information about the "surroundings", i.e. its urban context, is needed. As a matter of fact, such information is more and more digitally available nowadays thanks to the growing availability of 3D city models that, however, generally consist of GIS data. The recent advances of spatial data acquisition and processing technologies have in fact brought a considerable yield of 3D data, with particular focus on the built environment. These data often consist either in point clouds, or in polygon-based models – the latter following the B-rep model. Additionally, GIS data are georeferenced, and semantics may be also added, as in the case of city models based on the international standard CityGML (Gröger and Plümer, 2012).

Collecting, harmonising, integrating and merging these kinds of heterogeneous data (both from AEC and GIS domains) can still be a challenging task for a practitioner, especially if they lack deep knowledge on data integration strategies (Clemen and Hendrik, 2019; Zhu et al., 2019; Zhu and Wu, 2021). A lot of scientific literature already exists on the topic of BIM and GIS integration. However, providing an extensive review of available articles and papers is beyond the scope of this work. Garramone et al. (2020) and Zhu and Wu (2022) provide nevertheless a good starting point for the interested reader.

The work presented in this paper is not meant to provide a general solution to tackle all BIM-GIS integration problems. Instead, it focuses only on an approach when it comes to the specific use case of performing interactive preliminary environmental analyses (e.g. shadowing, solar and wind) at block/district scale in the case of a new planned project (e.g. a building). The work stems from a collaboration between MSc students and members of the 3D Geoinformation group at TU Delft and Royal HaskoningDHV, a world-wide consultancy firm with expertise in engineering, architecture, digital technologies and software solutions. The main idea is to extend an existing tool based on Rhinoceros/Grasshopper (Rhinoceros, 2024), originally developed in-house by Royal HaskoningDHV to support the work of their architects, designers and engineers. In its original version, the tool allows users to directly import geospatial data in Rhinoceros, de facto bypassing the generally cumbersome data acquisition via different data portals. However, the tool is currently limited only to the Netherlands, and it does not support data sources from other countries. As a result, the goal is to enhance it, in that a more generic framework is defined and developed to enrich the tool when it comes to a) the usability with datasets from different countries, and, additionally, b) the capability to perform additional environmental analyses. The proposed methodology is the result of a preliminary analysis in terms of data and software requirements, carried out between GIS specialists and practitioners from the AEC domain. In particular, the latter have expressed their needs in terms of data and

functionalities, as well as set some limitations in terms of software solutions. These will be listed in the next section.

This paper presents an overview of this collaboration experience, its main goals, the developed methodology, its implementation, the results carried out with data from different countries, followed by the conclusions and a discussion on the current limitations and the future improvements.

## 1.1 Data and software requirements

During initial meetings between TU Delft and Royal HaskoningDHV, a preliminary analysis of the requirements was carried out in order to have a clear picture of what is needed and which functionalities are to be added/implemented. The reason behind this analysis, both in terms of functionalities and constraints, stems from the desire to extend the existing tool and to adapt it to the needs and the actual workflows of the end users, who are mainly AEC practitioners with limited GIS skills. The requirements of such preliminary analysis are summarised in the next points.

In terms of <u>data</u> requirements:
R1) Regarding the projected object, a vector-based, parametric 3D model of a "new" building will be used;
R2) Regarding the surrounding "urban scene", data of surrounding 3D buildings (at least in LoD1), 3D vegetation (e.g. trees), 2.5D terrain (e.g. a DTM), land use, cadastre, orthophotos, and transportation should be downloaded, *ideally* as open-data and, *ideally*, for several countries in a way that is as transparent as possible for the end user;
R3) Ancillary data, required for the different simulations (e.g. weather data) should be also downloaded "automatically" depending on the context. However, it was agreed to leave this specific requirement out during this first development stage of the prototype.

In terms of <u>software</u> requirements:
R4) The new functionalities should be developed within Grasshopper/Rhinoceros. The reason is that Grasshopper is one of the most commonly used software solutions in the AEC domain for parametric modelling and design;
R5) In order to reduce the software installation and management effort (an aspect that may represent a limiting factor in certain environments/companies), it has been decided to avoid adding or linking to external libraries, unless they are already natively supported by Grasshopper;
R6) The import of GIS data for the "surrounding" urban scene into Rhinoceros/Grasshopper should *ideally* prefer existing web-based APIs (e.g. WFS and WMS) instead of file downloads. Data import via file-based downloads should be considered only as a fall-back alternative;
R7) The identified environmental analyses cover the simulation of the shadowing effect of the new building on the surroundings (and vice-versa), solar analysis on all exposed surfaces of the new building (e.g. to evaluate PV potential), as well as the effect of the new building in terms of wind flow;
R8) In order to perform such analyses, the Ladybug tools have been identified as the target tools (Ladybug, 2024). The reasons are multiple: they are already well-know and used by the AEC community, and there are interfaces in Grasshopper to directly use the different Ladybug tools;
R9) Nevertheless, currently, support to feed GIS-based "3D urban scene" data into the Ladybug tools is rather scarce, therefore this step should be automatised and made more user-friendly;

R10) The overall user experience, including data retrieval, integration and, especially, simulation time, should be *ideally* short enough to allow the user to set up a simulation scenario (e.g. by defining the shape of the building, and placing it in the urban scene), run the simulation, look at the results, and repeat the process further adjusting the building shape and position without waiting for long simulation times;
R11) If the previous requirement cannot be fulfilled, then guidelines for the end user are to be drafted based on experience and on empirical evidence;
R12) Although some of the requirements are tailored to the needs of Royal HaskoningDHV, the developed prototype should be kept as generic as possible, in order to be reused by other practitioners, also outside of Royal HaskoningDHV;
R13) Finally, unlike the original tool developed in-house at Royal HaskoningDHV, the resulting prototype will be publicly available.

## 2. Methodology

As a result of the data and software requirements analysis described in the previous section, a methodology was defined and then implemented. Figure **1** provides an overview of it.
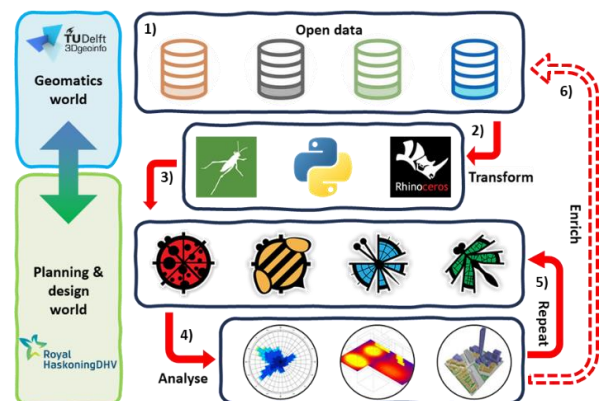


Figure 1. Overview of the developed methodology, successively implemented into a prototype tool for Rhinoceros/Grasshopper.

First of all (step 1), an evaluation of available data was carried out in different countries, the focus being – as far as possible – on the open data as listed in point R2), i.e. buildings, vegetation, DTM, land use, etc. In particular, besides the availability as *open* data, a research was carried out to find out how data are made available, i.e. via web services such as WMS or WFS (or not, i.e. via file download), their formats, etc. The findings will be presented in section 3.

Drawing from the results of the first investigation, data were collected for a selection of countries, transformed and imported into Rhinoceros (step 2), before being further processed in order to comply with the input data requirements of the different Ladybug simulation tools (step 3). In general, two main approaches were followed to prepare the 3D urban scene. The first approach is based on creating a B-rep-based geometry model, while the second one follows a voxel-based approach. Upon performing the desired simulations (step 4) the user can now explore the results and – if needed – adjust the 3D scene settings (e.g. the size and position of the building) and run the simulation again (step 5). Finally, the simulation results could be exported in order to enrich, if required, the input dataset (step 6). This last step would allow the output of a simulation to be later reused as the input for some other applications. Eventually, due to some of the issues encountered during the implementation, this

was not implemented and has been left for future improvements. The next sections will provide more details about each one of the steps shortly presented here.

### 3. Evaluation of (open) geospatial data sources

First, a sample of 10 candidate countries were selected (Australia, Finland, Frace, Germany, Hong Kong, Italy, Ireland, Spain, Taiwan, United Kingdom). After an initial screening, 5 countries (Germany, Hong Kong, Italy, Spain, United Kingdom) were chosen to be used. The findings of such data (and metadata) exploration were collected in tables, two of which are presented here in Table **1** and Table **2**. They are shown as representative opposite examples of the heterogeneity of cases that can be encountered in "real life". If Hong Kong offers a very rich set of open data, mostly reachable by standard APIs, the case of the Italian region of Piedmont is quite different, as data are available in a limited number of formats, and only via file download. The findings of all 5 selected countries, collected in similar tables, as well as of the remaining 5 countries, are available in Tsai *et al.* (2023).

At the conclusion of the first step covering the exploration and the evaluation of open data in (a sample of) different countries, it can be said that – as it was to actually expect – there is a huge heterogeneity of combinations in terms of data availability, formats, quality, retrieval solutions between the different

countries. If this, on the one hand, may sound obvious, on the other hand, performing this preliminary investigation has contributed to providing a less "fuzzy" picture of the situation – at least in the surveyed countries. Additionally, the survey strategies could be replicated with other countries that have not been considered yet. Nevertheless, such heterogeneity has a direct implication also in terms of the intended outcome of the to-be implemented tool, as it is unlikely that a single data import solution will fit for all countries. More realistically, different countries will require separate efforts if they were to be integrated into the tool. In other words, requirement R6 on GIS data import is, at the moment, reachable only partially. However, although similar considerations apply also to the different data formats, the number of "standard" formats was found to be more homogeneous, thus allowing for more integrated implementations.

### 4. Geospatial data import and preparation

This section covers steps 2 and 3, depicted in Figure **1**. Once the data source has been identified, the GIS data needed to reconstruct the 3D scene "around" the building must be first retrieved, and then transformed to be used in Grasshopper. The GIS data are then merged with the 3D model of the "new" building, and the resulting 3D urban scene must be further prepared depending on the type of envisioned simulation.

Table 1. Overview of findings on data in Hong Kong.

| Region | Theme | Name/Type | Format | API | Details | Coverage | Notes |
|---|---|---|---|---|---|---|---|
| **Hong Kong** | **Building** | 3D Visualisation Map (2017) | OBJ | WFS/WMS | 3D vector (LoD2) | Partially | Tiling indexes of the shapes; One file per tile |
| | **Land use** | Digital Topographic Map iB1000 | GML etc. | WFS/WMS | 2D vector | Fully | Tiling indexes of the files |
| | **Terrain** | Digital Terrain Model (DTM) | GEOTIFF | WMS | 2.5D raster | Fully | All the data at once |
| | **Orthophoto** | Digital Orthophoto DOP5000 | GEOTIFF | WFS/WMS | 2D raster | Fully | Tiling indexes of the files |
| | **Road** | Road Network | GML etc. | WFS/WMS | 2D vector | Fully | Tiling indexes of the files |
| | **Cadastre** | 3D Visualisation Map (2017) | OBJ | WFS/WMS | 3D vector (LoD2) | Partially | Tiling indexes of the shapes; One file per tile |
| | **Source** | CSDI (Common Spatial Data Infrastructure) portal: https://portal.csdi.gov.hk/geoportal/#searchPanel | | | | | |

Table 2. Overview of findings on data in the Italian region of Piedmont.

| Region | Theme | Name/Type | Format | API | Details | Coverage | Notes |
|---|---|---|---|---|---|---|---|
| **Italy (Piedmont region)** | **Building** | Footprints | SHP | Download | 2D polygon | Full | Has field for building height |
| | **Land use** | Land use map | SHP | Download | 2D polygon | Full | |
| | **Transport** | Roads | SHP | Download | 2D line | Full | |
| | | Train lines | SHP | Download | 2D line | Full | |
| | | Bike paths | SHP | Download | 2D line | Full | |
| | **Terrain** | DTM 5m | GeoTIFF | Download | 2.5D raster | Full | Download per tile |
| | | DSM 5m | GeoTIFF | Download | 2.5D raster | Full | Only available via email request |
| | **Source** | Piedmont geoportal: https://www.geoportale.piemonte.it/cms/ | | | | | |

## 4.1 Data retrieval

In order to import data into Rhinoceros and Grasshopper, two main strategies were developed to retrieve the necessary geospatial data and build the 3D urban scene. The first strategy relies on OGC APIs, namely the Web Feature Service (WFS) and Web Map Service (WMS) APIs. A WFS request is used to retrieve vector-based data. A WMS request is used instead to return raster-based data. As schematically represented in Figure 2, the implementation in the Grasshopper tool requires the user to provide only the geolocation of the "new" building (position) and a distance value used to compute the size of the buffer "around" the building (radius). These two parameters are then used to query the WFS or the WMS servers, and download the respective data, very often made available in tiles.

Drawing from the results of step 1, a second, alternative strategy was developed as a fall-back in case no OCG APIs are available. It is the case when the end user has to manually download files containing data, and to upload them into Rhinoceros/Grasshopper. Still, if an intermediate webpage is available, then the user can still draw the extents of the area of interest and locally download the respective files.
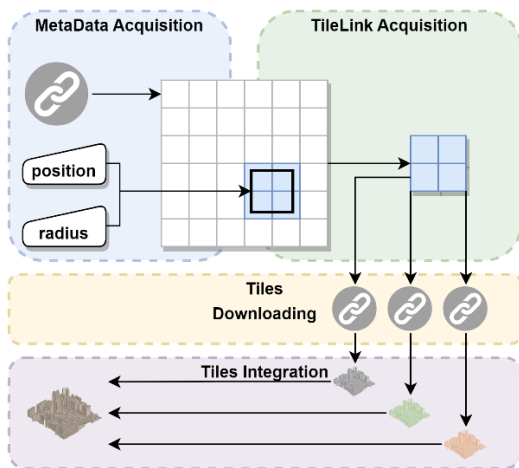


Figure 2. Schematic overview of the data retrieval interface developed for Grasshopper and based on OGC APIs.

## 4.2 Data import and preprocessing

Unfortunately, several "classical" GIS formats cannot be directly imported and used "as is" in Rhinoceros/Grasshopper, therefore an intermediate data conversion must be carried out, either using additional plugins (e.g. in the case of shapefiles), or by developing specific scripts depending on the source format (e.g. with CityGML or CityJSON). Besides data conversion, other potential problems may arise and must therefore be considered and, as far as possible, solved. For example, there could be issues due to data incompleteness, inconsistency or inaccuracy. Some issues can be solved, as in the case of overlapping/disjoint data or in presence of small "holes" in the data; other issues cannot, e.g. in the case of totally missing data. As a consequence, the end user is still required to perform some manual checks, as the process cannot be completely automated. Nevertheless, depending on the type of input data, different strategies were implemented to check and repair geometry issues for the successive environmental simulation analysis. A description of such issues, and some remedies are described for example in (Donkers et al., 2016; Padjen et al., 2022). Eventually, the GIS data must be translated to a local coordinate system (this is the

default case in Rhinoceros/Grasshopper), before the "new" building model can be integrated. Figure 3 provides a visual example of the final 3D urban scene with the "new" building model integrated into the urban "surroundings".

All the aforementioned operations were implemented either using existing tools/plugins in Rhinoceros/Grasshopper, or by developing specific scripts in Iron Python or in C# (both are supported within Grasshopper). Given the complexity and number of heterogeneous issues found so far, it was decided, from this point onwards, to focus on the generation of the urban 3D scene by using only the building and the terrain data as these two datasets are essential to the desired analyses. More implementation details can be found in Tsai et al. (2023).
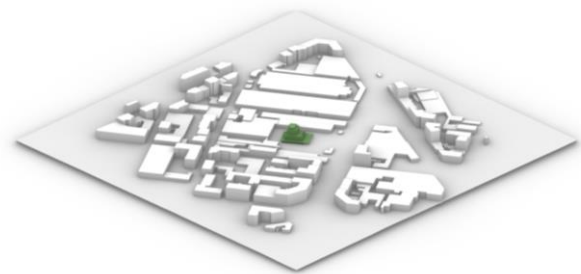


Figure 3. Example of the 3D model of the "new" building (in green) merged into the surrounding urban scene.

## 4.3 3D urban scene for simulation: B-rep and voxels

In order to prepare the 3D data for the Ladybug simulation tools, and based also on their input requirements, two strategies were developed to prepare and generate the 3D urban scene. The main difference lies in the geometry type used to represent the 3D urban scene.

The first strategy adopts a B-rep representation for the geometries. More specifically, a TIN (Triangulated Irregular Network) is obtained from the integration and (re)triangulation of terrain and building data. The basic idea is to use neighbouring features to check and repair non-consistent vertices (isolated vertices etc.), edges (dangling edge etc.) or faces (intersected faces etc.) to form a valid triangulated mesh (usually a 2-manifold) that closely approximates the original geometry. The most commonly used TIN is the Delaunay Triangulation (DT), which is fundamental data structures for terrains, both for their representation and for their processing (Ledoux et al., 2022).

The second strategy consists in a voxel-based method. This means that the 3D space of the urban 3D scene is divided into small, equally sized cubic cells, called voxels. Each voxel is analysed and used to represent the geometry within that region. Depending on the application, the exterior boundary (envelope) of the volumetric representation can be also used as a simplified approximation of the original one.

From our experiments, the TIN-based strategy can provide a finer representation of the 3D urban scene, but it is more complex in terms of operations required to obtain a geometrically and topologically correct surface. The voxelization is instead more stable and relatively simpler to implement, although it has a higher computational cost to be generated and, in general, a coarser representation of 3D urban scene (depending on the voxel size). Figure 4 presents an overview of the 3D models obtained from the data of the 5 selected countries. The top row contains B-rep models, the lower row voxel-based ones.
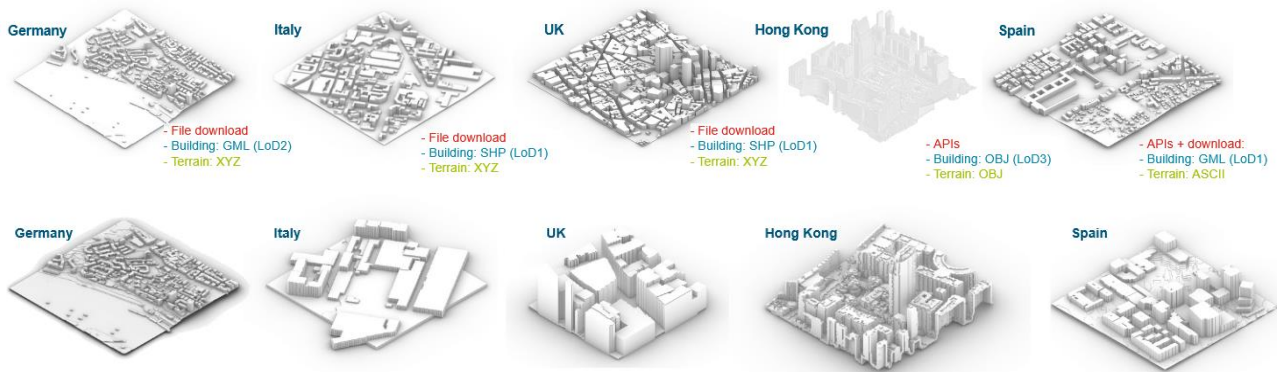
Figure 4. Overview of the 3D urban scene models obtained by integrating the buildings and the DTM from the input GIS data. In the top line, models are represented as B-rep geometries, together with some information on the origin of the data. In the lower line, the models resulting from the integration and voxelization of the input data. Some of the voxel-based models correspond actually to a portion of the respective B-rep ones.

## 5. Environmental simulations

With the 3D urban scene models ready, either based on B-rep or voxels, different simulations tests were carried out. In the following sections, a brief overview of the two main types of simulations that were carried out will be provided.

### 5.1 Solar analysis

For the solar simulations, the Radiance engine was used. Radiance is an environmental simulation engine primarily focused on the accurate simulation of light, daylight, and energy flows within built spaces. Radiance (Radiance, 2024) is known for its precision in predicting the distribution of light and visualising the impact of architectural designs on illumination and thermal comfort. Additionally, it is embedded in the Ladybug tool "Honeybee" for Rhinoceros/Grasshopper.

Although each simulation can have peculiar settings and characteristics, the general workflow can be summarised as follows:
1) Initial configuration of the simulation engine and input of the relevant geometry (i.e. our 3D urban scene);
2) Specification of the necessary environmental and simulation parameters (e.g. temperature, wind speed, as well as user-defined parameters like temporal and spatial resolution and iteration times);
3) Actual execution of the simulation;
4) Post-processing of the results, e.g. for visualisation and analysis purposes.

Various components are required to perform a solar analysis. First, a specific timestamp (or time interval) is needed, for which the corresponding weather data must be provided as an EPW (EnergyPlus Weather) file of a nearby weather station. The 3D urban scene is then provided as a geometrical input model, either as B-rep or voxel-based. In our test, we chose to simulate the direct sun hours hitting a certain surface. This involves analysing the influence that the "new" building has on the total hours of direct sunlight received by the surrounding areas. Secondly, the analysis can be "switched", i.e. the focus is on the solar radiation received by the "new" building itself, but also considering the nearby urban scene. In either case, the results are expressed in $kWh/m^2$, and can be visualised directly in Rhinoceros/Grasshopper as shown in Figure 5.
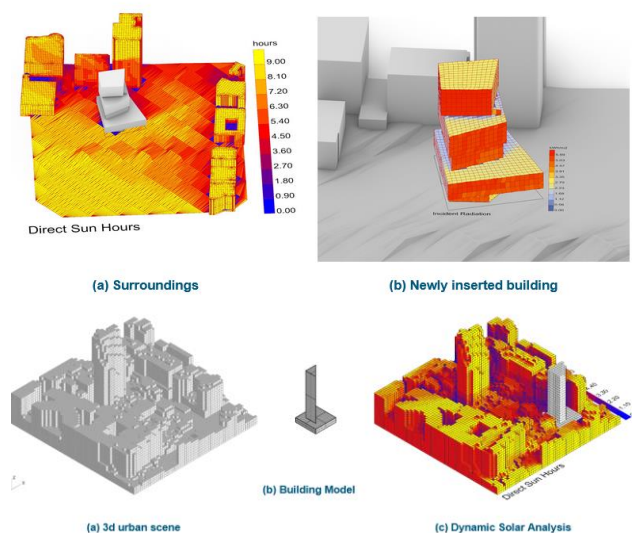


Figure 5. Examples of results from the solar analyses. [Top, left] Computation of direct sun hours on the "surroundings", including the shadowing effect of the "new" building (not textured), using the B-rep model. [Top right] A similar analysis, however carried out only on the "new" building but considering the shadowing from the nearby urban features. [Bottom] Representation of a conceptually similar analysis, however carried out using a voxel-based model.

### 5.2 Wind analysis

For the wind simulations, the OpenFOAM engine was used. OpenFOAM is an open-source computational fluid dynamics (CFD) engine which plays a significant role in simulating airflow, thermal comfort, and pollutant dispersion within architectural and urban spaces (OpenFOAM, 2024). OpenFOAM offers flexibility and customization, allowing AEC professionals to address complex airflow and environmental issues in building design and urban planning. Additionally, it is also embedded in the Ladybug tool "Butterfly" in Rhinoceros/Grasshopper. Paraview is used for the visualisation of wind simulation results and it is conveniently packaged with OpenFOAM during the Butterfly installation.

Also in the case of wind simulation, various components and configuration parameters are required, and the preparation of the simulation is generally much more complex than in the solar analysis case.
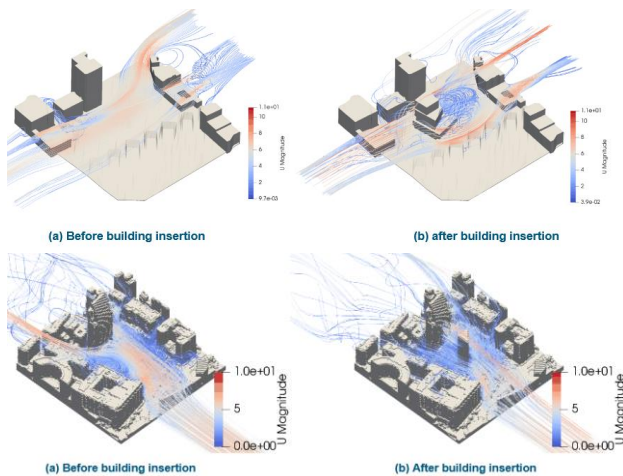
(a) Before building insertion

(b) after building insertion

(a) Before building insertion

(b) After building insertion

Figure 6. Examples of wind simulation results: [top] using B-rep input models, before and after inserting the "new" building. [Bottom] Visualisation of conceptually analogous simulation results, however using the voxel-based models.

For example, considerations must be made when it comes to the turbulence model, the solving algorithms and the solver itself. As these are specific settings affecting the accuracy of the results, they have been left out of scope, as the main goal was first to investigate how to succeed in running a simulation at all.

Therefore, for the sake of this work, an MRE (minimal reproducible example) was defined, adopting mostly default or automatically generated parameters for experimental purposes. Nevertheless, even when defining a simple test case, e.g. using the laminar turbulence model, the steady incompressible recipe and the default solver for simple-foam, there are still parameters to directly specify. Among them are the wind tunnel, boundary conditions, background mesh and snappyHexMesh (Blocken, 2015; García-Sánchez et al., 2021). It is important to emphasise that this MRE is not meant as an accurate representation of real-world scenarios due to the complexity of the involved parameters. It is solely intended to test the efficacy of the developed data import and conversion/transformation pipeline.

Keeping in mind the aforementioned clarification and limitations, the general steps can be summarised as follows:
1) Initial configuration of the simulation engine and input of the relevant geometry (our 3D urban scene) and simulation parameters, such as, for example, the wind direction (for wind tunnel creation), the cell size of blockmesh (for background mesh creation), the refinement level of buildings (for refined mesh creation), the number of CPUs (for the parallel running);
2) Actual execution of the simulation;
3) Post-processing of the results, e.g. for visualisation and analysis purposes. Due to the limited capability of Rhinoceros/Grasshopper, the post-processing and visualisation were carried out in ParaView.

Figure 6 provides examples of results from the wind simulation, expressed as wind speed in m/s, using both the B-rep and the voxel-based models as input

## 6. Challenges and discussion

This section contains an overall discussion on the challenges encountered during the implementation of the prototype tool (see the requirements in section 1.1, the methodology overview in Figure 1, and the first experiences collected using it. Only a

selection of main points will be discussed here, while the reader is invited to refer to Tsai et al. (2023) for a more detailed and comprehensive list.

First of all, regarding the (open) data retrieval – and according to requirements R3) and R6) – data should be ideally available only via OGC APIs. However, very often this was not the case. Actually, datasets were mostly retrievable only through file-based download. Several data formats required additional plugins or additional processing before they could be imported into Grasshopper. For example, terrain data were often available as gridded rasters (e.g. as GeoTIFF). Unfortunately, this rather common GIS format cannot be directly read by Grasshopper. Therefore, the raster file(s) had to be converted into a point cloud using the Python package "rasterio" (Rasterio, 2024), in order to be eventually parsed as a text file through the Grasshopper Python interpreter. Other alternatives were tried, but to no avail.

Regarding the ("surrounding") buildings, the ESRI shapefile format was the most common data format encountered. However, the built-in Grasshopper "Import SHP" functionality was found to be limited as it does not keep the field values associated with each geometry. In order to generate simple LoD1 buildings by extruding the footprints by a given height, this is a rather strong limitation. For this reason, the Grasshopper plugin "Local Software" (Local Software, 2024) had to be used to overcome said limitation. Alternatively, GIS data could be retrieved according to the CityGML standard. In this case, again, no native support is available, therefore the GML file was parsed with various Grasshopper Python components and scripts developed ad hoc to extract the required geometries.

Regarding the automatic generation of the 3D urban scene by merging/fusing the different datasets – as required by R9) – this could be achieved only partially, as it strongly depends on the quality and type of the GIS input data and the target geometry representation (B-rep or voxel-based). In general, a B-rep model of the 3D urban scene is required to be watertight for simulation purposes, in particular for the wind ones. This can be achieved, for example, by means of a CDT (Constrained Delaunay Triangulation). For this reason, the method proposed by Pađen et al. (2022) was adapted and implemented within Grasshopper. However, testing on the different datasets showed that it tends to be computationally intensive, therefore conflicting with requirement R10) that asks for short computation times during the whole pipeline. For this reason, a simplified, but faster, approach was also developed to obtain a B-rep model of the 3D urban scene. However, speed comes at the cost of robustness, in that sometimes this method cannot solve issues as in the case of overlapping tiles. Nevertheless the developed pipeline can create a B-rep representation of the 3D urban scene that can be generally used for the simulation purposes, although one last transformation is required in the case of the terrain TIN: for wind simulation an additional step is needed and it involves providing the terrain with a certain thickness by extruding it in the negative z direction by a certain value. Only now the B-rep geometry model will comply with the Butterfly requirements allowing for the wind simulation to be executed. Finally, a potential limitation of the B-rep models lies in their resolutions. Too dense models (i.e. with too many small triangles) may lead to extremely long simulation times, or even crash the simulation engine when the Grasshopper memory limit is reached (see later). This calls for some user-defined simplification of the input geometry, e.g. in terms of the terrain TIN obtained from the (raster-derived) point cloud.

The voxel-based approach can in part overcome some of the limitations of the B-rep one, because it can easily convert various input data into a 3D urban scene that satisfies the needs of (near) real-time environmental analysis and simulation (or when the B-rep method is not applicable or suitable). The voxelization approach has worked with different configurations of data from the 5 selected countries. The challenge consists however in choosing the right voxel size, as a too fine voxel resolution could lead to long simulation times or, more frequently, to crashes of Rhinoceros. For this reason, a series of tests at different resolutions and for different sizes of the study area (expressed as number of tiles) were carried out, in order to measure the associated simulation time – also considering requirement R10) on the "short times" – and gather some experience based on empirical evidence. As can be seen from Table 3, the bottleneck is very often the system memory. When the 3D grid resolution is too high, the RAM memory limit of Rhinoceros (2 GB) is reached, crashing it. Alternatively, frequent interaction between memory and disk cache can significantly slow down the simulation speed. As suggested also by Garcia-Sánchez *et al.* (2021), setting the 3D grid resolution at 2m×2m×2m allows to obtain acceptable compromise between quality of the results and simulation time. Still, this excludes a fully automatic solution, as the user should nevertheless first perform some tests to adjust the resolution according to the 3D urban scene size.

Table 3. Comparison of simulation times for wind simulation depending on the size of the study area (expressed as number of tiles of size 128m×128m and the 3D grid resolution).

| Number of tiles | 3D grid resolution (m×m×m) | Simulation time (s) |
|---|---|---|
| 1 | 4×4×4 | ≈0 |
| 1 | 2×2×2 | 14.0 |
| 1 | 1×1×1 | 109.9 |
| 4 | 4×4×4 | 6 |
| 4 | 2×2×2 | 49.3 |
| 4 | 1×1×1 | Out of memory |
| 9 | 4×4×4 | 15.2 |
| 9 | 2×2×2 | 116.0 |
| 9 | 1×1×1 | Out of memory |

### 7. Conclusions and outlook

This paper has presented the methodology and the implementation results of a prototypic tool for Rhinoceros/Grasshopper that allows importing data from the GIS domain into a 3D modelling software typically used by AEC practitioners. The main idea, and the main use case, is to allow for environmental analyses (namely solar and wind simulation) of a "new" building, designed and modelled in Rhinoceros/Grasshopper, in the urban context where it is planned to be built. In other words, the "surrounding" urban scene should be also considered and included in the simulation process.

The idea started from an existing tool, developed in-house at Royal HaskoningDHV that allows importing data from the Netherlands, and led to its extension in order to work with data from different countries, and to the addition of the capability to perform additional environmental analyses. A series of data and software requirements were identified and discussed between both GIS and AEC practitioners.

During the data evaluation stage, it has become evident that the GIS data availability and accessibility varies greatly across the different countries, therefore introducing additional challenges and reducing the *ideal* goal of a universally applicable tool. The lack of standardisation between countries often requires manual data retrieval strategies that hinder the automation of data integration, making it less viable for end-users such as users outside of the GIS community who may have a limited understanding of geodata.

When it comes to modelling the 3D urban scene required for the environmental simulations, two strategies were envisioned, implemented and tested: a TIN-based and a voxel-based one. Both have advantages and disadvantages. For example, the TIN method allows to model more accurately the geometries of the 3D urban scene. However, it is also heavily dependent on the quality and interoperability between input datasets, often requiring complex and time-expensive data transformation and integration operations. On the other hand, the voxel-based method is more fault-tolerant. However, it requires the user to set a suitable voxel size, as too fine voxel resolution, and/or a too large 3D urban scene may become problematic and lead to long simulation times or software crashes. Nevertheless, experiments have shown that a 3D grid resolution of 2m seems to be an acceptable compromise, at least for the type of preliminary environmental analysis carried out at this stage of the project.

In the end, the experience collected has allowed us to define a methodology and implement a prototypic tool that has been tested with data from 5 different countries, and for which both solar and wind simulations have been carried out. Figure **7** provides an overview of the results obtained with the different datasets used to test the tool. All resources are available as open source software in a GitHub repository (SynProj, 2023).

In future, besides an overall code overhaul and restructuring (ideally in C# instead of Python, as it is much faster), it would be
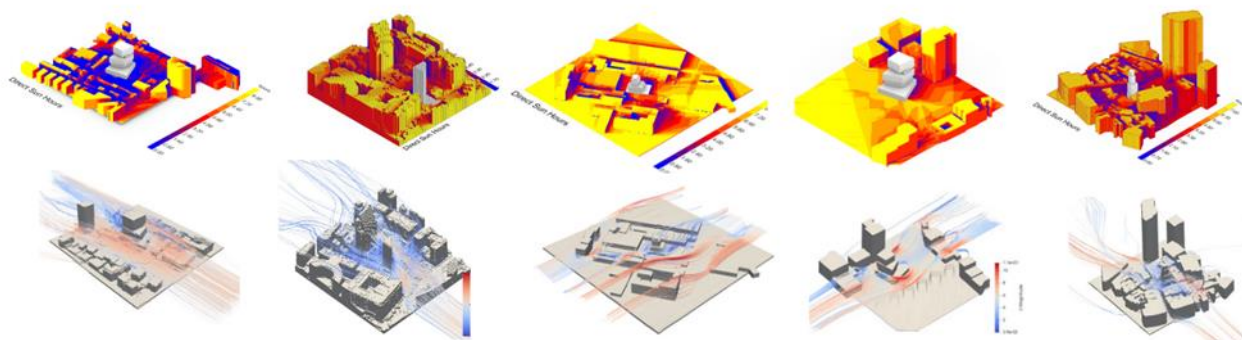


Figure 7. Overview of the solar [top] and wind [bottom] simulation results carried out with datasets from the selected 5 different countries, using the same tool for Rhinoceros/Grasshopper developed in the work described in this paper.

interesting to try the tool with data from other countries, in order to further test it and gather further insights. Additionally, a further step in the implementation should include those urban objects that, at this stage, have been left out, namely 3D vegetation (e.g. trees), land use, transportation, and so on.

## References

Agugiaro, G., García González, F.G., Cavallo, R., 2020. The city of tomorrow from... the data of today. *ISPRS Int. Journal of Geo-Information*, MDPI, 2020, 9(9), 554. https://doi.org/10.3390/ijgi9090554.

Blocken, B., 2015. Computational fluid dynamics for urban physics: importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations. *Building and environment*, 91, 219–245.

Clemen, C., Hendrik, G., 2019. Level of Georeferencing (LoGeoRef) using IFC for BIM. *J. of Geodesy, Cartography and Cadastre*. 2019(10), pp. 15-20.

Donkers, S., Ledoux, H., Zhao, J., Stoter, J., 2016. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LoD3 buildings. *Transactions in GIS*, 20(4), 547–569.

Garcia-Sánchez, C., Vitalis, S., Paden, I., Stoter, J., 2021. The impact of level of detail in 3D city models for CDF-based wind flow simulations. *ISPRS Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* , XLVI-4/W4-2021, 67–72.

Garramone, M., Moretti, N., Scaioni, M., Ellul, C., Re Cecconi, F., Dejaco, M. C., 2020. BIM and GIS integration for infrastructure asset management: a bibliometric analysis. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, VI-4/W1-2020, 77–84, https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-77-2020.

Gröger, G., Plümer, L., 2012. CityGML—Interoperable semantic 3D city models. *ISPRS J. Photogramm. Remote Sens.* 2012, 71, 12–33.

Jusuf, S.K., Mousseau, B., Godfroid, G., Soh Jin Hui, V., 2017. Integrated modeling of CityGML and IFC for city/neighborhood development for urban microclimates analysis. *Energy Procedia*, 2017(122), pp. 145-150, ISSN 1876-6102. https://doi.org/10.1016/j.egypro.2017.07.329

Ladybug, 2024, Ladybug tools homepage. https://www.ladybug.tools (last accessed on 12 April 2024)

Ledoux, H., Arroyo Ohori, K., Peters, R., Pronk, M., 2022. Computational modelling of terrains. Self-published.

Local Software, 2024, Local Software Food4Rhino page. https://www.food4rhino.com/en/app/local-software (last accessed 12 April 2024)

Noardo, F., Ellul, C., Harrie, L., Overland, I., Shariat, M., Arroyo Ohori, K., Stoter, J., 2020. Opportunities and challenges for GeoBIM in Europe: developing a building permits use-case to raise awareness and examine technical interoperability challenges. *Journal of Spatial Science*, *65*(2), 209–233. https://doi.org/10.1080/14498596.2019.1627253

Ohori, K.A., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J., 2017, Towards an integration of GIS and BIM data: what are the geometric and topological issues. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf*, IV-4/W5, 2017, pp. 1–8.

OpenFOAM, 2024, A free, open source CFD software. https://www.openfoam.com (last accessed 12 April 2024)

Padjen, I., Garcia-Sánchez, C., Ledoux, H., 2022. Towards automatic reconstruction of 3D city models tailored for urban flow simulations. *Frontiers in built environment*, 141.

Radiance, 2024, A Validated Lighting Simulation Tool. https://www.radiance-online.org (last accessed 12 April 2024)

Rasterio, 2024, Geospatial raster I/O for Python programmers. https://rasterio.readthedocs.io/en/stable/ (last accessed on 12 April 2024)

Rhinoceros, 2024. Grasshopper on-line resource. https://www.rhino3d.com/learn/?query=kind:%20grasshopper&modal=null (last accessed on 12 April 2024)

SynProj, 2023, GitHub repository, https://github.com/biscuittsai1022/Synthesis-project_1-repository_2023 (last accessed 12 April 2024)

Tsai, B.S., Huizer, L., Giampaolo, M., Monté, S., Gong, S., 2023. (Co-)development of an open-data-based tool to perform preliminary environmental analyses at district scale in different (European) countries. Delft University of Technology. http://resolver.tudelft.nl/uuid:f837eb15-b62b-4981-865c-b85bac2d0a65

Zhu, J., Wang, X., Chen, M., Wu, P., Kim, M.J., 2019. Integration of BIM and GIS: IFC geometry transformation to shapefile using enhanced open-source approach. *Automation in Construction*, 2019(106), 102859, ISSN 0926-5805.

Zhu, J., Wu, P., 2021. Towards effective BIM/GIS data integration for smart city by integrating computer graphics technique. *Remote Sensing*. MDPI AG,13(10).

Zhu, J, Wu, P., 2022. BIM/GIS data integration from the perspective of information flow. Automation in Construction, 2022(136), 104166, ISSN 0926-5805. https://doi.org/10.1016/j.autcon.2022.104166.