BATCHPLAN: A LARGE SCALE SOLUTION FOR FLOOR PLAN EXTRACTION

BURAK YILDIZ[1], JAVIER CUARTERO[2], FATEMAH MOSTAFAVI[3] and SEYRAN KHADEMI[4]
[1,3,4]Technical University of Delft.
[2]Kaan Architecten.
[1]b.yildiz@tudelft.nl , 0000-0001-9932-4221
[2]j.Cuartero@kaanarchitecten.com, 0009-0007-5981-4112
[3]f.mostafavi@tudelft.nl, 0000-0002-8047-2168
[4]s.khademi@tudelft.nl, 0000-0003-4623-3689

Abstract. The development of Building Information Modelling (BIM) has enabled new opportunities, such as standard data storage and collaborative building design. Moreover, there exist many Life Cycle Assessment (LCA) tools and Building Energy Performance (BEP) simulators that use the Industry Foundation Classes (IFC) exports of BIM platforms as input for further operational analysis. While the extracted IFC files contain numerical and tabular data from the BIM model, the visual data including floor plans and section drawings is often obtained directly from the original 3D software such as REVIT. In this study, we introduce an open-source solution, BatchPlan, for batch processing IFC files of medium- and high-rise building projects, leading to floor plan extraction on a large scale. Furthermore, we have designed a user-friendly graphical interface that allows users to select floors manually. BatchPlan is based on open-source Python packages; thus users can easily edit and adapt it to their specific requirements. The presented solution enables a scalable data generation pipeline for downstream tasks that require extensive quantitative analysis, such as machine learning models to perform material detection, volume estimation, and environmental impact prediction.

Keywords. Floor plan extraction, Industry Foundation Classes (IFC), Building Information Modelling (BIM), Architectural Technical Drawings, Big Data

1. Introduction

The emergence of Building Information Modelling (BIM) has improved building design efficiency by managing all the project data. To improve sustainable assessments of buildings in the design and evaluation phases, BIM tools are becoming more popular

ACCELERATED DESIGN, Proceedings of the 29th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2024, Volume 1, 201-210. © 2024 and published by the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Hong Kong.

in the construction industry (Santos et al., 2019) and (Yeung et al., 2023). The Industry Foundation Classes (IFC) format file is an open file format used by BIM programs, including spatial elements, materials, and shapes of a building or facility (ISO 16739-1, 2018). The majority of BIM modelling software, including Revit, ArchiCAD, and Rebro, currently supports the import and export of IFC files (Jiang et al., 2019). IFC files are intended to be platform-independent, and therefore play an information exchanger role. This would allow interoperability between various BIM programs (Liu et al., 2021), as well as between BIM and other tools such as LCA (Xu et al., 2022).

To adapt to more diverse applications, IFC files can be converted to various file types such as Green Building XML (gbXML) (Elagiry et al., 2020). Moreover, since IFC files contain machine-readable unique identifiers alongside semantics (e.g., object type or function), characteristics or attributes (e.g., materials, color, and thermal properties), relationships (e.g., locations, connections, and ownership), and objects (e.g., columns and slabs), there is an opportunity to make use of them in automated procedures (Lai et al., 2019). In addition, given the difficulties stemming from multi-resource heterogeneous BIM software, IFC files have been used in automated building performance calculations (Deng & Lu, 2023).

Despite all the possibilities, not all BIM software is free to use or open source. In an open-source software, the source code is accessible, and licensing allows to modify and extend the software's functionalities. As a result of such transparency, users can improve and customize the software according to their own needs. Accordingly, this study introduces BatchPlan, a large-scale solution for floor plan extraction of buildings' IFC files. The proposed solution is developed to address the challenge of analysing big data during a data-driven design process, utilizing the conventional method of data storage in the building industry in an accelerated manner. The main contributions of this study are therefore as follows:

- Introducing an open-source, customizable, and flexible solution for floor plan extraction in large scale
- Enabling visual (floor plan maps), geometrical (room- and unit-wise coordinates), and numerical (area, volume, etc.) outputs data format
- Batch-processing large IFC files
- Facilitating large-scale quantitative analysis and ML-based workflows

The remainder of the papers is divided into three main subjects: 1) discussing related work regarding the floor plan extraction and the available open-source solutions using IFC files, 2) investigating the development of BatchPlan and the underlying dataset, 3) presenting the features of BatchPlan in comparison to other solutions for the task of floor plan extraction.

2. Related Work

2.1. FLOOR PLAN EXTRACTION

The plan, elevation, and section views of a construction structural drawing correspond to the different engineering data including dimensions, construction entities, and annotations, respectively (Björk & Laakso, 2010). Building plans represent a building on a horizontal plane at a particular floor level and convey information about architectural objects' distribution, extension, and outline (Yin et al., 2020).

The idea of extracting indoor spatial information from floor plans has gained a lot of attention in previous studies (Kim et al., 2021). The extraction of technical architectural drawings on the other hand is still often done manually. BIM software offers a user-friendly graphical interface for the manual extraction of floor plans at the scale of a single building design. However, dealing with a substantial volume of files simultaneously is not practical using this software. Also, manually extracting 2D architectural plans for all stories in large-scale projects can be a cumbersome process and is prone to errors. The mentioned limitations shed light on the advantages of automated extraction of floor plans, more preferably on a large scale.

2.2. IFC-BASED OPEN-SOURCE SOLUTIONS

There exist several solutions that take the IFC files as input and give the user the possibility to view and edit BIM files. These solutions range from toolkits to programming packages and libraries. Several available solutions are explained in this section.

IfcOpenShell is an open-source toolkit that helps in dealing with IFC files (IfcOpenShell, 2023). Viewing models, extracting attributes, moving objects, and generating 2D drawings are among the toolkit's possible functionalities. More specifically, IfcConvert is a command-line application of IfcOpenShell for converting IFC geometry into file formats such as OBJ, DAE, GLB, STP, IGS, XML, SVG, H5, and IFC itself. IfcOpenShell has been implemented for IFC-based BIM reconstruction in previous studies (Pan et al., 2023).

Xbim is an open-source BIM toolkit that supports the IFC file format (xbim Toolkit, 2023). Building Information Models (BIMs) can be read, created and viewed using Xbim. In addition to geometric and topological operations, there is also support for visualization of IFC files. Xbim Utilities, an example application of the toolkit allows performing bulk functions on IFC files. Another application, Xbim Exchange, makes the conversion of IFC files to COBie format possible.

Other than toolkits with multiple applications, there are few related solutions offering more limited functionalities. IFC.js is an open-source library that allows for viewing IFC models in web browsers (Viegas, 2023). Through its role as a bridge between different software, including commercial options, it demonstrates how open-source solutions can improve interoperability. In addition, BIMvision is a free IFC model viewer (Datacomp IT Sp. z o.o., 2023), allowing for viewing a virtual model from Revit, Archicad, or other CAD systems. Cutting the plane of the selected model face, edge, or the global axis is among the possibilities that BIMvision offers.

Most of the tools and solutions have been developed mainly with the purpose of either viewing or converting IFC files, whereas BatchPlan is developed to not only process big IFC files, but also provide other visual, geometrical, and numerical outputs which will be used in many design-related downstream tasks. This is an urgency to be addressed given the fast-growing development of data-driven methods in architectural design discipline and industry. Hence, the creation of workflows and pipelines for generating extensive and machine-readable datasets is crucial. Example measures are taken in the field for the shift from conventional databases (Pizarro et al., 2022) to ML-ready datasets (Van Engelenburg et al., 2023). Implementing BatchPlan, a large-scale solution for floor plan extraction of buildings, not only helps accelerate the process of outputting visual and numerical data from IFC files but also contributes to developing proper datasets for large quantitative analysis and ML-based tasks.

3. BatchPlan

We introduce BatchPlan as a robust large-scale floor plan extraction tool designed to be highly customizable, extensible, and pluggable in various capacities. The design decisions are meticulously crafted, particularly for the processing of extensive BIM data stored in IFC files. Customization is facilitated through the provision of user-defined element filters and styling. Filter functions serve the purpose of selectively excluding elements from the final outputs, allowing users to specify the elements they wish to exclude. Styling functions, on the other hand, are employed to define the appearance of elements, encompassing attributes such as colour coding in the generated outputs. Both filter and styling functions operate on an IFC element and its corresponding OpenCASCADE Boundary Representation (BRep) geometry. The filter functions return a Boolean value, while the styling functions provide the desired element style. Users have the flexibility to enhance BatchPlan by supplying a user-defined output formatter to tailor the output to their preferred format. An example output of this formatter can be a list of geometrical details of each room in WKT (well-known text) format. Furthermore, BatchPlan seamlessly integrates into other programs or data pipelines through its Application Programming Interface (API). The architectural framework of BatchPlan along with the data flow within it are shown in Figure 1.

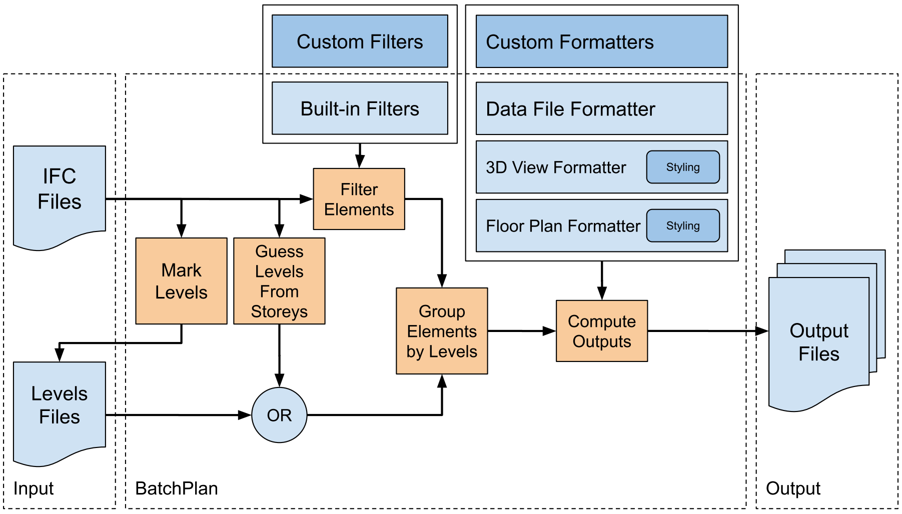


Figure 1. BatchPlan architecture and data flow of one single project from input to output. All orange boxes indicate main computing points where only level marking can be done using BatchPlan's Graphical User Interface (GUI) and others using Command Line Interface (CLI). Dark blue areas are

where user-defined codes can be injected.

BatchPlan accepts one mandatory and one optional input. The mandatory input comprises IFC files, while the optional input involves level files. These level files are formatted as comma-separated vector (CSV) files and encompass heights at which BatchPlan computes floor plans. Users can generate these files conveniently through BatchPlan's Graphical User Interface (GUI). Alternatively, one may manually create a CSV file or employ a script to populate it with the desired heights. This deliberate design choice offers users the flexibility to extract floor plans at various heights on a scalable basis. The level files are optional since BatchPlan can use storey information to automatically calculate the heights. The data flow within BatchPlan, illustrated in Figure 1, unfolds as follows:

- IFC files given in command line arguments are read.
- IFC "Cutting" planes on which floor plans are generated are determined by employing heights from either the corresponding level files or storey information.
- The filters, either provided or selected by user, are executed on the elements.
- The remaining elements, post-filtering, are categorized by levels. This categorization involves grouping elements with a non-empty intersection with the same plane into a common bucket.
- The intended outputs are computed for each of the buckets, created at the previous step, using the selected and/or provided output formatters, subsequently saved to output files.

BatchPlan is implemented in the Python programming language, and its source code (Yildiz et al., 2023) is made available under the MIT License, known for its permissive nature as an open-source license. The selection of Python as the programming language aimed at facilitating ease of modification and contributions for future users. The code leverages two primary open-source packages, namely IfcOpenShell (IfcOpenShell, 2023) and pythonocc (pythonocc-core, 2023). IfcOpenShell serves the purpose of reading and processing IFC files, while pythonocc is utilized for identifying intersections between planes and elements. Additionally, built-in formatters are implemented using pythonocc.

3.1. DATASET

For the development and the test phase of the BatchPlan, we curated a dataset from KAAN Architecten (KAAN Architecten, 2023) projects. Six distinct residential projects were selected to encompass various design stages, ranging from early design to project construction, including building permit documentation and design development stages.

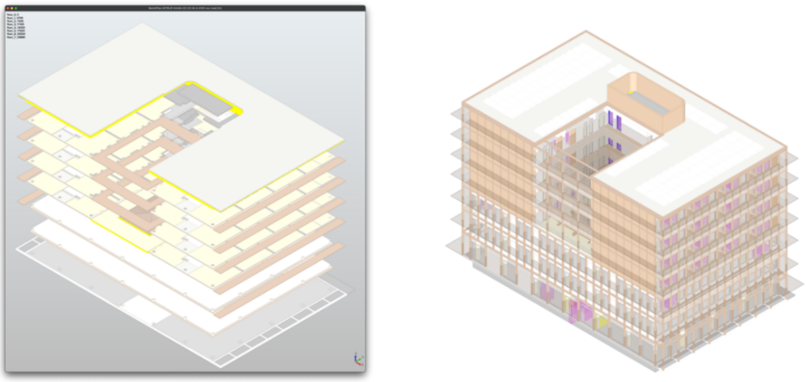
To ensure consistency and facilitate meaningful comparisons among dataset assets, a decision has been made to focus on both medium- and high-rise housing architectural examples. These projects have been chosen to address different scenarios in which the evolution of the geometric (detailed) aspects of architecture could potentially impact the results of BatchPlan or its processes. Details of the selected projects are provided in Table 1.

Table 1. Details of the selected projects from the KAAN Architecten dataset.

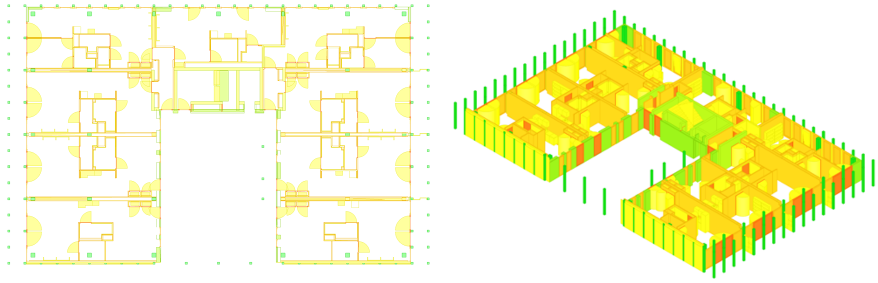
Project Name	Category	Design Start	Design End	Gross Floor Area (m2)	Number of Storeys	Number of Elements
SPOT	Construction	2017	2021	40,000	35	47,560
Lumiere	Design	2018	-	50,000	46	33,865
Zalmhaven	Built	2015	2018	49,000	24	27,900
Overhoeks	Construction	2017	2019	11,400	9	11,720
Blok O	Design	2022	-	150,000	9	10,270
Strijp S	Design	2018	2023	4,800	7	5,340

4. Results

BatchPlan is designed to efficiently process large-scale data while maintaining ease of customization. To assess its capabilities, the KAAN Architecten buildings dataset has been employed. Sample outputs of the BatchPlan's GUI are presented in Figure 2, highlighting the features of the proposed solution, from marking the levels at which the floor plans are desired to be extracted, to customized 2D or 3D color-coding of the outputs depending on a certain criterion.



(a) (b)



(c) (d)

Figure 2. A sample GUI view and sample outputs of BatchPlan. (a) BatchPlan GUI for marking the levels at which the floor plans are extracted, (b) 3D view of the whole building with planes indicating the levels where the floor plans are extracted, (c) Customized color-coding of a floor plan using an external data source (i.e., carbon footprint values). (d) Customized color-coding of the 3D view of the single floor plan in (c).

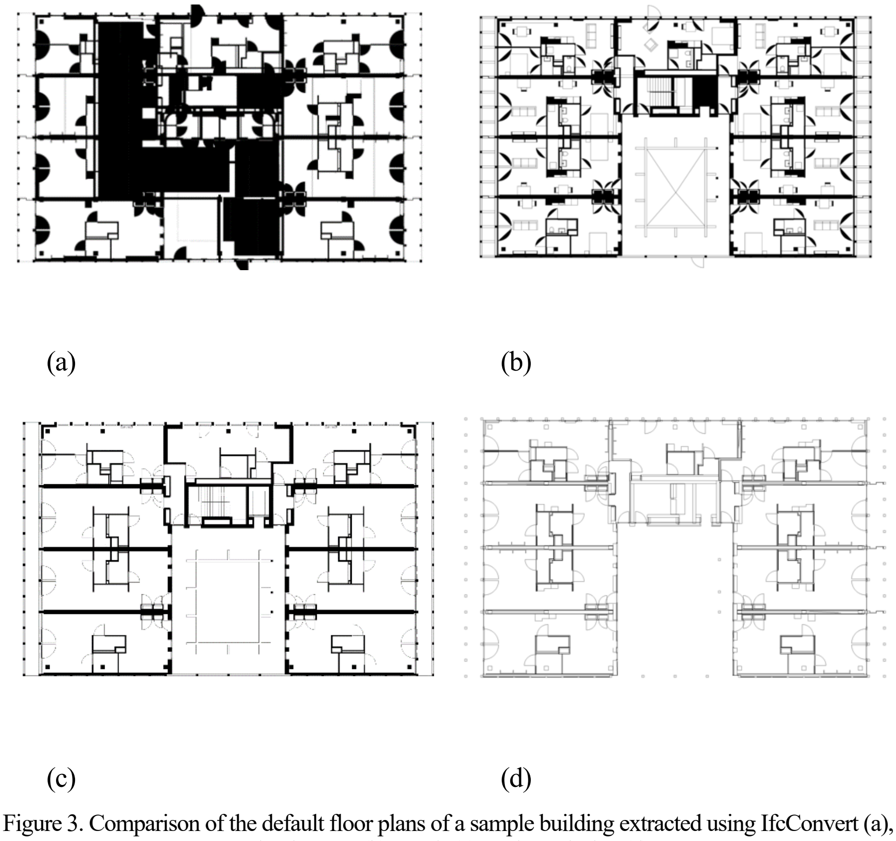
A comparative analysis of BatchPlan's features with those of selected software and tools capable of floor plan extraction was also conducted. The results of these comparisons are presented in Table 2. The MIT license has been chosen for BatchPlan, affording users the right to modify, customize, and redistribute it under both open source and proprietary licenses without hesitation. This approach is advantageous for both the open-source community and the industry, serving non-profit as well as for-profit purposes. One distinctive feature of BatchPlan is its customization capability, allowing users to provide their own styling functions to alter the styling of floor plans or 3D output views. These styling functions can also incorporate external data sources to manipulate the appearances of the elements. For example, users can create a heatmap by coloring the elements according to their respective environmental footprint.

Table 2. Comparison of the software and tools related to floor plan extraction

Program	IfcConvert	BlenderBIM	Revit	BatchPlan
License	LGPL-3.0-or-later	GPL-3.0-or-later	Proprietary	MIT
Source Code	Open source	Open source	Closed source	Open source
User Interface	CLI	GUI	GUI	CLI
Customizable	Not easy	Not easy	Limited	Easy
Extensible	No	Not easy	Not easy	Easy
Scalable	Yes	No	No	Yes
Platform-Independent	Yes	Yes	No	Yes

Another feature of BatchPlan is its extensibility, signifying that in addition to built-in output formatters such as rendered images and 3D views, users can introduce customized output formatters. These output formatters are functions that accept a set of IFC elements and their corresponding OpenCASCADE BRep representations, generating customized outputs such as data files and rendered images. The data files may encompass associated properties of the elements, such as their volume and locations. Another example of a customized formatter could be one that produces rendered images with bounding boxes around elements.

Sample floor plans have been extracted using the software and tools listed in Table 2, as illustrated in Figure 3. IfcConvert produces floor plans in the form of vectorized images, particularly in the Scalable Vector Graphics (SVG) format. Achieving a satisfactory floor plan in IfcConvert necessitates setting specific parameters and modifying the output SVG file. BlenderBIM, equipped with a user-friendly GUI, requires manual effort for extracting floor plans for each building and its individual levels. Revit, being a proprietary software, also entails manual intervention for floor plan extraction. In contrast, BatchPlan offers the advantage of extracting floor plans for an entire set of projects with a single command, streamlining the process significantly.



(a) (b)

(c) (d)

Figure 3. Comparison of the default floor plans of a sample building extracted using IfcConvert (a), BlenderBIM (b), Revit (c), and BatchPlan (d).

5. Findings and Discussion

Floor plans serve as the orthogonal projection of a building level, depicting building elements such as walls, doors, and windows to convey the architect's design. With the development of data-driven design methods, the need for processing big architectural data arises. In this study, an open-source, customizable, and extensible solution for floor plan extraction, called BatchPlan, is introduced. The development of the proposed solution was presented, and the results were compared to the similar available solutions. The presented solution can serve the research on big data resulting from large amount of existing IFC files. Moreover, to test the features of the BatchPlan, the solution was applied to six medium- and high-rise housing projects.

The development of BatchPlan facilitates flexibility, while concurrently prioritizing the efficient processing of large-scale data. Since BatchPlan offers processing and outputting large amounts of data, it serves further downstream tasks in which big data is needed. For instance, a possible application is developing a diverse collection of floor plans to create a dataset, enabling the training of an AI-driven model. The outputs of this study will not only assist designers in generating floor plans at the desired building levels but also enable researchers to perform quantitative analyses on district or urban scales.

Besides the mentioned features, BatchPlan has some limitations, mostly depending on the familiarity of the users with various interfaces. Although BatchPlan has a GUI for level marking, it is executed through its Command Line Interface (CLI). While the CLI offers advantages, particularly when running the program for batch processing on High-Performance Computing (HPC) clusters, some users may find it less convenient. Further developments of the proposed solution include enhancing the GUI tailored towards the architectural design process and allowing other types of output formats.

Acknowledgements

We thank our collaborators from the KAAN Architecten firm who provided the required data of building projects that assisted the research. This study was funded by NWO (Dutch Research Council) under the file number of GOCI.KIEM.02.017.

References

Björk, B. C., & Laakso, M. (2010). CAD standardisation in the construction industry — A process view. Automation in Construction, 19(4), 398–406. https://doi.org/10.1016/J.AUTCON.2009.11.010

Datacomp IT Sp. z o.o. (2023). BIMvision. https://bimvision.eu/

Deng, X., & Lu, K. (2023). Multi-level assessment for embodied carbon of buildings using multi-source industry foundation classes. Journal of Building Engineering, 72. https://doi.org/10.1016/j.jobe.2023.106705

Elagiry, M., Bourreau, P., Charbel, N., & De Angelis, E. (2020). IFC to Building Energy Performance Simulation: A systematic review of the main adopted tools and approaches. https://www.researchgate.net/publication/345983333

IfcOpenShell. (2023). The open source IFC toolkit and geometry engine. https://ifcopenshell.org/

ISO 16739-1. (2018). Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.

Jiang, S., Jiang, L., Han, Y., Wu, Z., & Wang, N. (2019). OpenBIM: An Enabling Solution for Information Interoperability. Applied Sciences 2019, Vol. 9, Page 5358, 9(24), 5358. https://doi.org/10.3390/APP9245358

KAAN Architecten. (2023). KAAN Architecten - Netherlands based architectural firm. https://kaanarchitecten.com/

Kim, H., Kim, S., & Yu, K. (2021). Automatic Extraction of Indoor Spatial Information from Floor Plan Image: A Patch-Based Deep Learning Methodology Application on Large-Scale Complex Buildings. ISPRS International Journal of Geo-Information, 10(12). https://doi.org/10.3390/ijgi10120828

Lai, H., Deng, X., & Chang, T.-Y. P. (2019). BIM-Based Platform for Collaborative Building Design and Project Management. Journal of Computing in Civil Engineering, 33(3), 05019001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000830/ASSET/A68C72D6-F281-4755-8C15-EA811AA6B442/ASSETS/IMAGES/LARGE/FIGURE11.JPG

Liu, L., Li, B., Zlatanova, S., & van Oosterom, P. (2021). Indoor navigation supported by the Industry Foundation Classes (IFC): A survey. Automation in Construction, 121, 103436. https://doi.org/10.1016/J.AUTCON.2020.103436

Pan, Z., Yu, Y., Xiao, F., & Zhang, J. (2023). Recovering building information model from 2D drawings for mechanical, electrical and plumbing systems of ageing buildings. Automation in Construction, 152, 104914. https://doi.org/10.1016/J.AUTCON.2023.104914

Pizarro, P. N., Hitschfeld, N., Sipiran, I., & Saavedra, J. M. (2022). Automatic floor plan analysis and recognition. Automation in Construction, 140(April), 104348. https://doi.org/10.1016/j.autcon.2022.104348

pythonocc-core. (2023). GitHub - tpaviot/pythonocc-core: Python package for 3D CAD/BIM/PLM/CAM. https://github.com/tpaviot/pythonocc-core

Santos, R., Costa, A. A., Silvestre, J. D., & Pyl, L. (2019). Integration of LCA and LCC analysis within a BIM-based environment. Automation in Construction, 103, 127–149. https://doi.org/10.1016/J.AUTCON.2019.02.011

Van Engelenburg, C., Khademi, S., Mostafavi, F., Standfest, M., & Franzen, M. (2023). Modified Swiss Dwellings: A ML-ready Floor Plan Dataset of Residential Building Complexes. https://doi.org/10.4121/e1d89cb5-6872-48fc-be63-aadd687ee6f9.v1

Viegas, A. G. (2023). IFC.js Open source IFC library. https://github.com/IFCjs

xbim Toolkit. (2023). making building information flow. https://docs.xbim.net/

Xu, J., Teng, Y., Pan, W., & Zhang, Y. (2022). BIM-integrated LCA to automate embodied carbon assessment of prefabricated buildings. Journal of Cleaner Production, 374, 133894. https://doi.org/10.1016/J.JCLEPRO.2022.133894

Yeung, J., J Hahn Menacho, A., Marvuglia, A., Navarrete Gutiérrez, T., Beach, T., & Rezgui, Y. (2023). An open building information modelling based co-simulation architecture to model building energy and environmental life cycle assessment: A case study on two buildings in the United Kingdom and Luxembourg. Renewable and Sustainable Energy Reviews, 183, 113419. https://doi.org/10.1016/J.RSER.2023.113419

Yildiz, B., Mostafavi, F., Cuarteo, J., & Khademi, S. (2023). Carbon Image Project. https://github.com/byildiz/BatchPlan

Yin, M., Tang, L., Zhou, T., Wen, Y., Xu, R., & Deng, W. (2020). Automatic layer classification method-based elevation recognition in architectural drawings for reconstruction of 3D BIM models. Automation in Construction, 113. https://doi.org/10.1016/j.autcon.2020.103082