Artificial Intelligence, Social Computing and Wearable Technologies, Vol. 113, 2023, 290–296
https://doi.org/10.54941/ahfe1004201

Auto3DBuilder: An Automatic 3D Building Modeling Tool From 2D Drawings

Amartuvshin Narangerel, Minjin Myagmarjav, Woong Hee Lee, and Dongwoo Lee

AST Holdings, Gyonggi-do Province, Suwon city, Republic of Korea

ABSTRACT

This paper presents a novel 2D to 3D building modeling tool seamlessly integrated into our Smart City Lab platform. The tool, designed for simplicity and user-friendliness, enriches existing LoD1 buildings without requiring modeling expertise. Leveraging Grasshopper as a backend and utilizing RestAPI interfaces, it transforms 2D architectural drawings into detailed 3D models. We delve into the methodology, showcasing how users can effortlessly generate intricate building models. The potential applications extend to urban planning, simulations, and innovative technologies like AR, VR, and the metaverse. Acknowledging current limitations, we outline prospects for future advancements, including accommodating more sophisticated building designs and enhancing interoperability for extensive data integration. This tool marks a significant step towards extensive and interactive urban development and planning.

Keywords: Building automatic modeling, Digital twin, Urban platform, Smart city lab

INTRODUCTION

In recent years, numerous nations and local governments have been actively investing in Smart City initiatives and development of Urban Digital Twins creating a surge of advancement in urban technologies, one of such is Urban digital twin platform aka Smart city platform. This platform operates as an interactive arena where urban stakeholders actively participate, testing new innovations, conducting in-depth evaluations of the current urban landscape, and performing diverse simulations in dynamic urban-scenarios delivering fundamental information in urban development, planning and decision-making (Biljecki *et al.*, 2015). Notable examples of such cases include the Digital Twin of the City of Zurich (Schrotter and Hürzeler, 2020) and Virtual Singapore, among others.

The increasing demand for Urban Digital Twin platforms underscored the necessity for high-quality data pertaining to the virtual urban environment. This is particularly critical because the level of detail in buildings and infrastructures significantly impacts the precision of simulating real-world scenarios within digital twin platforms. However, a substantial number of 3D building models, frequently falling under the classification of Level of Detail

1 (LoD1) models that lack crucial specific building features, are actively employed within urban digital twin platforms. This limitation stems from historical practices, where city municipalities primarily dealt with 2D drawings for building permissions, especially for smaller-scale constructions. Budget constraints and traditional methods further reduced the imperative for comprehensive 3D modeling, relegating it to select public structures with larger budgets. Recent technological advancements have empowered digital twin platforms to analyze and visualize urban landscapes with detailed 3D models. Furthermore, continuous progress is underway in developing tools designed to generate highly detailed 3D building models, typically categorized as LoD3 or higher.

In this paper, we introduce an essential feature of our Smart City Lab platform: an automatic 2D to 3D modeling. This function is designed to be a user-friendly tool for various stakeholders who have little or no modeling expertise, aiming to enrich the platform's existing LoD1 buildings. While the initial Smart City Lab platform is a web platform which is developed on the Cesium engine, our tool will leverage Grasshopper—an algorithmic 3D modeling software for Rhinoceros3D—as a backend, seamlessly connected with the platform through RestAPI interfaces.

The paper includes brief overview of tools that generates 3D building models from various data sources. Then it delves into the implementation and methodology of our tool in depth. Lastly, it concludes with result and discussions relating limitations, future directions.

RELATED WORKS

Generating 3D building models is typically a labor-intensive task due to variations in floor plans and building drawings arising from differing drawing standards and styles, necessitating skilled modelers and dedicated software. Depending on the purpose and requirements, various types of 3D building models are generated, each following the specific framework of the software or system used. A common workflow for generating detailed building models from architectural drawings typically involves a two-phased consecutive generation method: analyzing the architectural drawing and then generating the building components in three dimensions.

Over the past few decades, substantial efforts have been invested in creating 3D building models using architectural floor plan images (Yin, Wonka and Razdan, 2008; Lim, Janssen and Stouffs, 2018). Advancements in technology have led to the utilization of lidar data and other scanned point cloud data to create more detailed 3D building models (Haala and Kada, 2010; Ochmann, Vock and Klein, 2019). Additionally, side views and aerial images play a vital role in capturing the overall shape of buildings (Yang, Zhao and Sun, 2018). Given the complexity of building layouts, researchers often focus on parsing floor plan information from image files using image-processing and artificial intelligence techniques (Dodge, Xu and Stenger, 2017; Dong *et al.*, 2021).

METHODOLOGY

The tool operates on a request-based system, enabling users to generate 3D building models by uploading 2D floor plans through the dedicated platform interface. This process consists of two phases. In the initial phase, users are required to modify the 2D CAD floor plans into a Drawing Exchange Format file (.dxf file) and prepare the essential groundwork for the 3D building model generation. Subsequently, in the second phase, the platform's backend uses the prepared.dxf floor plan files to generate the 3D building models.

The preparation of the 2D floor plan as a unified.dxf file is a crucial step, necessitated by the diverse formats and styles in which building drawing files are typically drafted. This preparation process is carefully designed to spare users from the arduous task of redrawing the entire building. Instead, it focuses on converting critical lines and annotations into predefined layers with a minimal amount of additional drafting, following a prescribed user-manual and.dxf template shown in Figure 1, where the original building layout (left side) and the modified version (right side) are visually represented. The input 2D building layout file must encompass essential building drawing elements to ensure the generation of a comprehensive 3D building model, complete with slabs, windows, doors, roof, and interior walls.

The second phase, dedicated to 3D building model generation, capitalizes on Grasshopper's flexibility and interoperability, functioning as a robust backend server. It seamlessly integrates with the existing Smart City platform and NAS servers through RestAPIs. The comprehensive system architecture is visually presented in Figure 2 in the next page. In Figure 2, the Smart City platform backend is depicted on the left-hand side in a blue background, while the Network Attached Storage (NAS server) is highlighted in pink at the right-bottom. The automatic 3D building modeling tool itself is represented in green at the right-top. The blue arrows connecting the blocks symbolize API data transmission, while the purple lines indicate file transmission within the image.

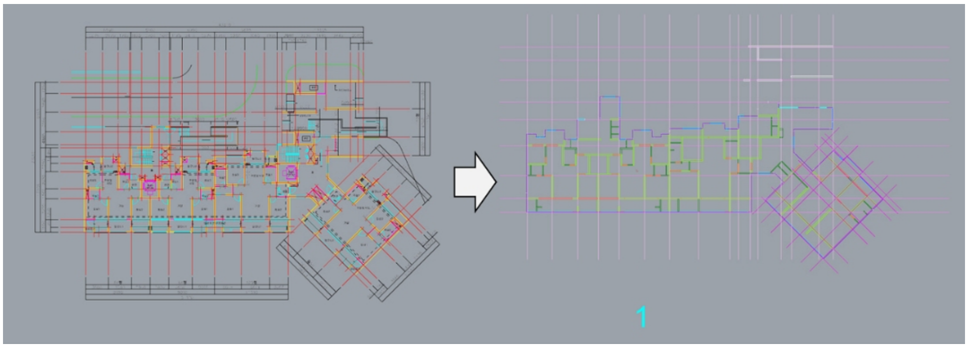

Figure 1: Modification of the architectural drawing. Simplified drawing at the right-hand side with the original at the left.

The process is set in motion as the users upload the 2D building layout file and provide additional detailed information via the platform. The uploaded.dxf files are stored in the NAS, and essential data, including users' supplementary information, is stored in the Database (DB) highlighted in cyan background in Figure 2.

Subsequently, the API module of the Grasshopper script queries the database and retrieves the first item from the request list into the Grasshopper environment. The script carefully checks and confirms the presence of the minimum required information in the.dxf file and accurately extracts layouts from designated layers. If the data is deemed sufficient, Grasshopper then seamlessly generates various building elements, in the order of external walls, floor slabs, roof structures, entrance doors, windows, columns, stairs etc. The combined building model is then exported to the appropriate NAS repository represented in pink in Figure 2. The platform DB list, alongside the dedicated scheduling list, is meticulously updated through Post and Patch requests.

In cases where the input.dxf file lacks critical information, such as the external wall layer, the 3D model generation process is concluded by updating the Database list and transmitting a precise error message. In contrast, upon successful generation, the user's request list is promptly updated, enabling users to seamlessly place the 3D model on the platform, which can later be utilized for various urban simulations.

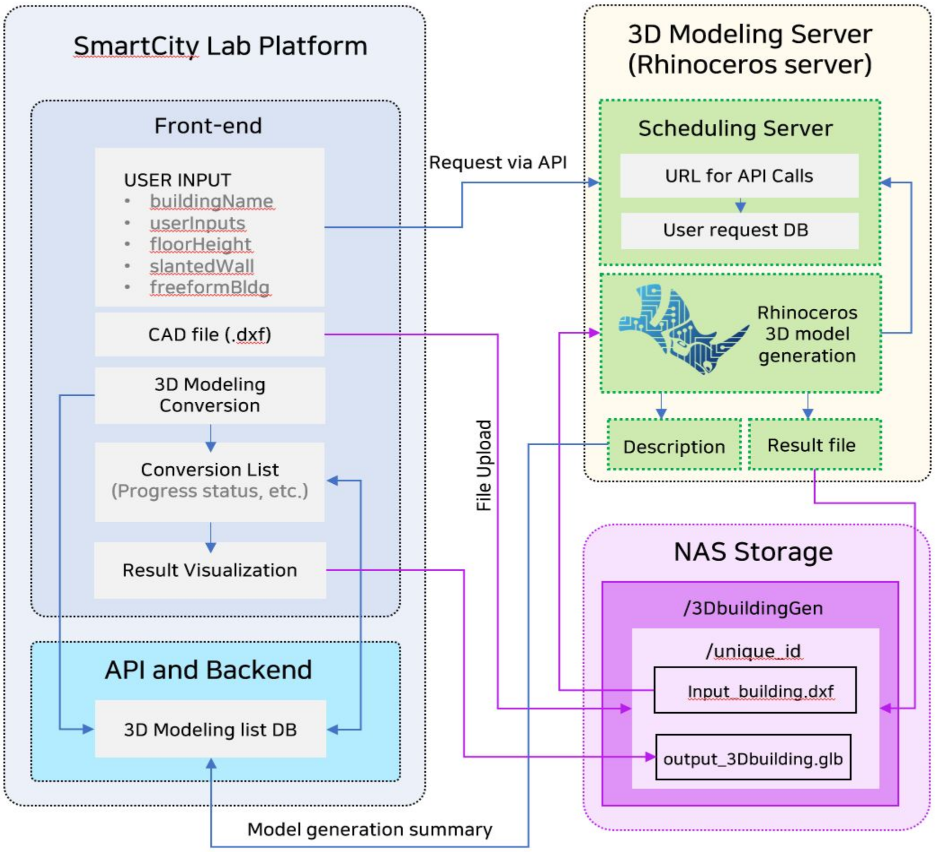

Figure 2: System architecture diagram of 3D model generator.

RESULT AND CONCLUSION

The tool underwent rigorous testing using multiple architectural drawings, ranging from simple single-story buildings without windows to relatively complex office and apartment buildings. An example of such an apartment building is illustrated in at the right-hand side of Figure 3. Depending on the intricacies of the 2D layout drawing and the computational power of the PC, the tool consistently demonstrated the ability to generate models within a matter of minutes. Smaller buildings were typically generated in about 10 seconds, while larger apartment buildings required around 5 to 10 minutes.

One significant advantage of our tool is its availability as a service within the platform. Users are relieved of the burden of modeling; instead, they can swiftly test their pre-converted drawings and iterate over their ideas efficiently. Moreover, the tool features an extensive array of error messages, providing clear indications of any issues with the input 2D drawing. This empowers users to make necessary modifications in the building drawing file and re-upload it for processing. The platform interface, depicted on the left-hand side in Figure 3, complements this user-friendly experience.

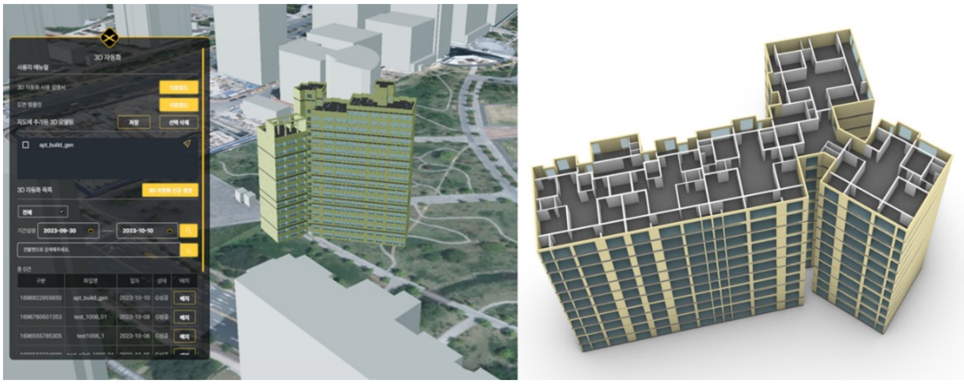

Figure 3: Example of generated 3D building model, platform interface at the left, and generated glb file at the right.

Additionally, our tool continuously evolves, with the Grasshopper script and user manual being regularly updated. This ensures that complex cases are documented in the database for ongoing development and refinement.

Furthermore, our tool holds immense potential, paving the way for expanding the platform's capabilities. It allows for simulations integrating indoor building structures, complemented by diverse representation methods (Narangerel *et al.*, 2018) for crucial applications like pathfinding for emergency escape (Zhou *et al.*, 2020), robot navigation (Pauwels *et al.*, 2023), and building energy prediction (Roudsari, Pak and Smith, 2013). The applications of these intricate building models extend beyond simulation, encompassing augmented reality (AR), virtual reality (VR), and potentially metaverse applications. These technologies significantly enhance the understanding of urban landscapes, offering novel perspectives on urban plans and experiences.

In future endeavors, we are committed to addressing certain limitations and enhancing the tool's adaptability. Presently, the tool is most effective when applied to buildings with straightforward and simplistic designs and structures. However, to be able to generate complex designs, intricate facades (Narangerel, Lee and Stouffs, 2017), and free-form (Castro Pena *et al.*, 2021) shapes we are considering the utilization of deep learning or machine learning algorithms, based on or current DB and grasshopper algorithms. Implementing these enhancements will significantly broaden the tool's usability, particularly for users with only a building drawing in image format.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & communication Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00412, Operation of Smartcity lab based on Digital Twin).

REFERENCES

Biljecki, F. et al. (2015) 'Applications of 3D city models: State of the art review', ISPRS International Journal of Geo-Information. MDPI AG, pp. 2842–2889.

Castro Pena, M. L. et al. (2021) 'Artificial intelligence applied to conceptual design. A review of its use in architecture', Automation in Construction. Elsevier B. V.

Dodge, S., Xu, J. and Stenger, B. (2017) 'Parsing floor plan images', in 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA). IEEE, pp. 358–361.

Dong, S. et al. (2021) 'Vectorization of floor plans based on EdgeGAN', Information (Switzerland), 12(5).

Haala, N. and Kada, M. (2010) 'An update on automatic 3D building reconstruction', ISPRS Journal of Photogrammetry and Remote Sensing, 65(6), pp. 570–580.

Lim, J., Janssen, P. and Stouffs, R. (2018) 'Automated Generation of BIM Models from 2D CAD Drawings', in, pp. 61–70.

Narangerel, A. et al. (2018) 'FBSMAP: The Spatial Representation Method for Intelligent Semantic Service in Indoor Environment', in, pp. 587–596.

Narangerel, A., Lee, J.-H. and Stouffs, R. (2017) 'Thermal and Daylighting Optimization of Complex 3D Faceted Façade for Office Building', in SharingofComputableKnowledge! Rome, pp. 209–218.

Ochmann, S., Vock, R. and Klein, R. (2019) 'Automatic reconstruction of fully volumetric 3D building models from oriented point clouds', ISPRS Journal of Photogrammetry and Remote Sensing, 151, pp. 251–262.

Pauwels, P. et al. (2023) 'Live semantic data from building digital twins for robot navigation: Overview of data transfer methods', Advanced Engineering Informatics, 56.

Roudsari, M. S., Pak, M. and Smith, A. (2013) 'Ladybug: a Parametric Environmental Plugin for Grasshopper To Help Designers Create an Environmentally-Conscious Design', 13th Conference of International building Performance Simulation Association, pp. 3129–3135.

Schrotter, G. and Hürzeler, C. (2020) 'The Digital Twin of the City of Zurich for Urban Planning', PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 88(1), pp. 99–112.

Yang, J., Zhao, J. and Sun, Q. (2018) '3D model retrieval using constructive-learning for cross-model correlation', Neurocomputing, 275, pp. 1–9.

Yin, X., Wonka, P. and Razdan, A. (2008) 'Generating 3D Building Models from Architectural Drawings: A Survey', IEEE computer graphics and applications, 29(1), pp. 20–30.

Zhou, X. et al. (2020) 'Accurate and Efficient Indoor Pathfinding Based on Building Information Modeling Data', IEEE Transactions on Industrial Informatics, 16(12), pp. 7459–7468.