Transforming IFC Models into a Structured 3D Format for Multimodal Localization from 2D Images in a Web-Based Tool

Robin Ov[1], Ayman Soultana[1], Angelina Aziz[1] and Roberto Perez Martinez[2]
[1]Ruhr University Bochum (RUB), Bochum, Germany
[2]Fraunhofer Heinrich Hertz Institute (HHI), Berlin, Germany
E-mail(s): robin.ov@ruhr-uni-bochum.de, ayman.soultana@ruhr-uni-bochum.de, angelina.aziz@ruhr-uni-bochum.de, roberto.perez.martinez@hhi.fraunhofer.de

Abstract: Accurate localization within built environments based on 2D images remains a fundamental challenge in construction informatics, robotics, and augmented reality applications. Existing approaches, such as SPVLoc (Semantic Panoramic Viewport Matching for 6D Camera Localization), have demonstrated reliable image-based localization by leveraging structured 3D representations. However, many localization pipelines primarily rely on point clouds and strcture-from-motion, which lack the semantic richness of Building Information Modeling (BIM) data. This study investigates the transformation of Industry Foundation Classes (IFC) models into a Structured3D format, a representation that combines 3D geometric primitives with semantic relationships, to improve localization accuracy using a multimodal approach and its integration into a web-based tool. Structured3D provides localization algorithms with contextual information beyond raw geometrics, enabling more robust pose estimations in complex indoor environments. The conversion from IFC models into a Structured3D format is implemented within a web-based tool that incorporates an IFC viewer, allowing users to interactively visualize and process the Structured3D data. The localization workflow integrates image-based and geometric data, enabling precise pose estimation. Camera positions obtained from the localization process are stored within the IFC model as documentation points, providing a structured record of captured viewpoints for further analysis. The approach is evaluated in real-world building environments, considering varying lighting conditions, occlusions, and perspective changes. The results indicate that IFC-derived Structured3D data, combined with web-based integration, enhance localization accuracy and usability. Future research will explore machine learning services to automatically detect building components and estimate their position within this pipeline.

Keywords: IFC, Structured3D, localization,SPVLoc, BIM, multimodal approach, 2D-to-3D matching, web-based tool

Published in the conference proceedings of the 36. Forum Bauinformatik 2025, Aachen, Germany, DOI: will/be/assigned/by/editor

1 Introduction

Accurate indoor camera localization within architectural environments is a fundamental challenge in fields such as navigation, augmented reality (AR), building management and robotics. Traditional 2D-to-3D matching techniques, from 2D camera images to locations in a 3D model, often rely on point clouds or on CAD-based models, which are geometrically detailed, but lack semantic notations for a comprehensive understanding of the scene. This lack of semantic information hinders effective localization, especially in complex indoor settings.

Research has shown the potential of integrating semantic information in localization pipelines. For example, Structured3D provides a dataset of synthetic annotated indoor scenes in which both geometric primitives and their semantic interpretation are described within a single framework [1]. This allows for detailed modeling of complex indoor environments, capturing both the arrangement of objects and their defined grouping. In turn, Structured3D has shown improved performance on Machine Learning tasks requiring holistic understanding, such as room layout estimation.

However, while Structured3D provides a powerful framework for scene representation, its reliance on synthetic scenes limits its applicability to real-world architectures. In practice, semantically rich models of buildings are ideally stored in Industry Foundation Classes (IFC), a standard format used in Builidng Information Modeling (BIM). However, the structure of IFC is complex and difficult to directly apply to ML tasks [2]. As a result, we propose a converter that transforms IFC models into the Structured3D format. By extracting IFC entities and their geometric primitives and reencoding them in the Structured3D schema, our tool enables a structured semantic representation of real-world architectural environments, enabling compatibility with ML tasks. We will also apply our converted S3D json files to SPVLoc, which is a camera localization method, based on the S3D framework[3].

2 Background

Indoor camera localization is a key task in computer vision that involves a range of 2D to 3D localization techniques, leveraging point clouds, CAD models, and semantically annotated datasets. The following section proves a detailed overview of the background on indoor camera localization.

Point cloud camera localizations like PICCOLO make use of both colored point clouds and a query panorama image of a scene for localization[4].This approach enables localization by matching the query image to the point cloud using a gradient descent algorithm, without requiring neural network training or the collection of ground-truth image poses. Other methods using point clouds along with image queries for camera localization include line-based 2D-3D pose estimation approaches that leverage geometric features in untextured point clouds, by extracting line segments [5]. Or i3dLoc, which adresses inconsistencies between query images and point clouds, due to conditions such as lighting or weather [6]. Their method leverages a Generative Adversarial Network to extract condition-invariant geometric features to match these query images with point clouds. But all of these methods share the limitation of the requirement of pre-scanning all rooms on which localization should be done, which can be costly, especially in large buildings.

Consequently, the advantages of utilizing a common format for capturing environment information become apparent. One widely adopted standard in architecture is the Industry Foundation Classes

(IFC) format, which is a general-purpose data format designed to enable interoperability across different software [7]. But the feature of utilizing IFC for different software comes at the cost of a complex structure, which are often underutilized in specific use cases. As a result, extracting relevant information can be inefficient and challenging, requiring conversion methods to tailor the data to the targeted application.

While IFC offers a comprehensive standard for building data, its complexity can limit its direct applicability in certain tasks. To better address specific needs, several specialized formats have been developed, such as CityGML for urban-scale 3D city representations, GeoJSON for geospatial data exchange or point clouds for detailed 3D scanning and mapping [8]–[10]. For representing indoor environments in camera localization, we use the Structured3D json format, which addresses the lack of comprehensive information for training ~~a~~ machine learning models [1]. The format of Structured3D makes use of geometric primitives such as **planes**, **lines** and **junction points**. The relationships of these primitives are also included in the format by matrices, defining their connectivity to define larger geometric objects along with their semantic interpretation. Environments in Structured3D format can currently be generated from floorplan images [11]. But the conversion can include errors due to low image qualities. Using IFC files comes with the benefit of being less vulnerable to errors, as IFC files contain detailed classes specifically defined for structural elements, such as walls, doors and windows. In previous studies, the Structured3D format has been used by SPVLoc to perform camera localization by combining the format with images [12]. Structured3D allows SPVLoc to avoid the need of detailed textured models or dense point clouds, cutting on costs. The pipeline involves utilizing a perspective RGB images to be localized to rendered semantic panoramas from Structured3D. Important to note is, that SPVLoc does not require fine-tuning for unseen Structured3D environments, enabling a quick and scalabe deployment which was successfully demonstrated in an ML-based visual fire safety inspection pipeline [11]. Similar 3D-based localization, such as Inloc rely on dense, image-derived 3D models without semantic abstractation and require detailed scans of the environment [13]. Other methods focus on specific application domains, such as Lost Shopping!, which performs camera localization from images within large shopping malls, requiring query images and floor plans as input [14]. Or Rent3D, which focusses on estimating room layouts from single images using floor plan priors, but do not perform global localization [15]. In contrast, SPVLoc offers a general-purpose solution that can operate across diverse indoor environments without scene-specific adaptation.

3 Methodology

The following sections outline the data format of Structured3D and the conversion process of ifc to Structured3D, which enables the localization process by SPVLoc.

3.1 Technical Background

IFC is a standardized object-based schema defined by entity hierarchies. The entities inherit properties of their respective parent entities, such as IfcWall inherits from IfcBuildingElement. The geometry are defined by boundary representations, such as faces, edges, and vertices. The format of Structured3D is a JSON dictionary, which makes use of geometric primitives such as **junction**, **lines** and **planes** points to define their relationships as seen in Figure 1.

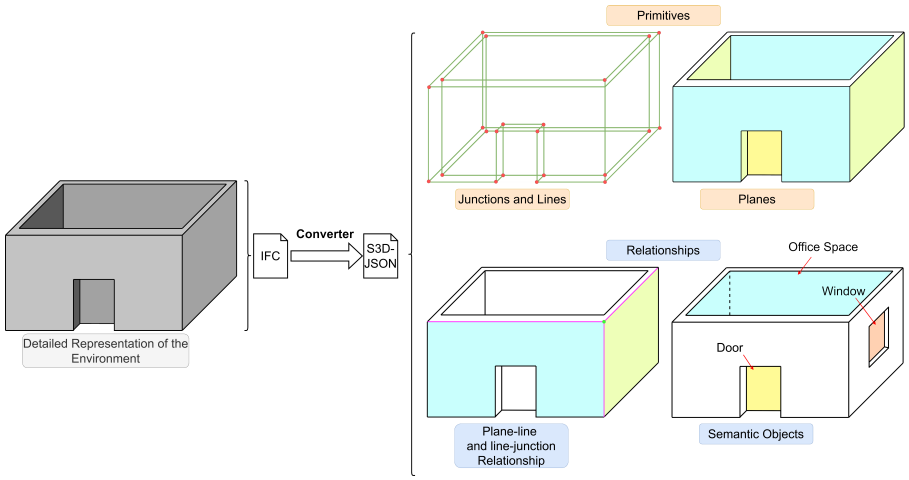

Figure 1: Overview for the IFC to S3D Converter

As shown in Figure 1, the **junction points** are defined as individual 3D coordinates. While the **lines** are defined by the coordinate of a junction point and the direction to the other connected junction. **planes** are defined by their normal and offset. The **lineJunctionMatrix** is of dimension LxJ, where L and J represent the lines and junctions, respectively. Each entry is of binary value, such that lineJunctionMatrix$_{lj} \in \{0, 1\}$. The value defines if the corresponding line and junction is connected. The **planeLineMatrix** follows the same structure as the lineJunctionmatrix, but it instead defines if the plane is bounded by the corresponding line. The **semantics** are defined by a tuple of planes, with the according name of the object. The initially proposed Structured3D format also contains cuboids and manhattan, however they were omitted in this work, as they are not utilized by the SPVloc pipeline (s3d quelle).

3.2 Conversion of IFC

As a first step, the IFC model's units are converted to millimetres to standardize all geometric measurements. The Conversion Process is divided in four main sections, extraction of lines and junctions, extraction of planes, extraction of semantics and openings, and finally the generation of the line-junction and plane-line matrix, as they can be seen in the previous Figure 1.

3.2.1 Extraction of Lines and Junctions

The first step involves parsing the IFC geometry to extract junction points and lines from the geometric mesh representations of building elements. The IFC building elements include IfcWall, IfcDoor, IfcWindow, IfcColumn, and IfcSpace. Each object is converted into a mesh using the ifcopenshell library, after which the vertices and edges are extracted. The 3D coordinates of the Vertices are

registered as junctions. The edges between the vertices are registered as lines. Objects that fill openings of walls, such as IfcDoor and IfcWindow, are saved seperately, as they are later used in the semantics section to annotate opening objects.

3.2.2 Extraction of Planes

The second step focuses on extracting planes from the building geometry. For each IfcWall, polygonal faces are identified and transformed into global coordinates. Planes are then defined using the face normal and offset, computed from three non-collinear points. Only vertical faces of walls are defined as planes, due to the horizontal faces not being visible from inside the rooms. The IfcRelVoidsElement relationship is used to identify openings, such as doors and windows, and associate them with the appropriate wall geometry. This entity in the IFC schema relates an opening element (such as an IfcOpeningElement) to its host wall. During parsing, each IfcRelVoidsElement is checked to ensure the voided element is an IfcWall, and a mapping from opening ID to wall ID is created. Once the corresponding wall is found, its associated vertical planes are retrieved. Next, the geometric mesh of the IfcOpeningElement is extracted, and its vertices are converted into millimetre-scale coordinates. From the bounding box of the opening, the shortest axis is determined, which is assumed to represent the depth or thickness of the opening (i.e., the direction across the wall). This axis is used to split the opening geometry into two parallel surfaces (front and back), which represent the two visible faces of a door or window opening in the wall. For both faces, a convex hull is computed over the corresponding face vertices to determine the lines of the polygon. Line directions are determined by traversing the loop, generating 3D lines defined by their starting junctions and directions. These lines are then evaluated against candidate wall planes to identify which wall plane each corresponding opening face lies on. This step is crucial, as Structured3D expects wall planes to include the door lines, which are assigned later in the planelinematrix. Finally, these front and back faces of the opening are assigned a new plane with the same parameters as their corresponding wall plane. Each vertex is registered as a junction, and the edge direction vectors are stored as lines. As a post-processing step, the normals of each plane are adjusted to face outward from the corresponding IfcSpace, which is the room they belong to.

3.2.3 Semantics

In the next step, the semantic structure of the IFC is extracted. This includes identifying all rooms and assigning their corresponding wall planes to each of them. Doors and windows are also added as seperate semantic elements. Each door or window is assigned the same ID as the room it connects to, meaning they are treated as part of that room's description. Before generating the Plane-Line and Line-Junction matrices, a post-processing step is applied. The floor and ceiling are added as planes and included in each room's semantic data and since IFC models typically define only one large floor and ceiling per level, these are divided so that each room is assigned its own portion.

3.2.4 Plane-Line and Line-Junction Matrix

The final components needed for the Structured3D JSON are the Plane-Line and Line-Junction matrices. The Plane-Line matrix specifies which lines lie on which planes. For each object, all lines are evaluated against each of its planes, and lines on the plane are marked 1; otherwise 0. For

doors and windows, lines must also be registered on the corresponding wall planes if they align. The Line-Junction matrix links each line to its defining junctions. For every line, the two nearest junctions corresponding to its start and end are identified within the object it belongs to and marked in the matrix.

With all the IFC information extracted, the Structured3D JSON is created. It includes sections for all junctions, planes, semantics, and the matrices. This JSON can now be used by SPVLoc to perform localization. Next, the results and a look of the implementation in the web-based tool will be shown.

4 Results

This section presents the implementation and evaluation of the IFC to Structured3D conversion pipeline integrated with SPVLoc. The results demonstrate the conversion interface and an example of localization performance using real building data.

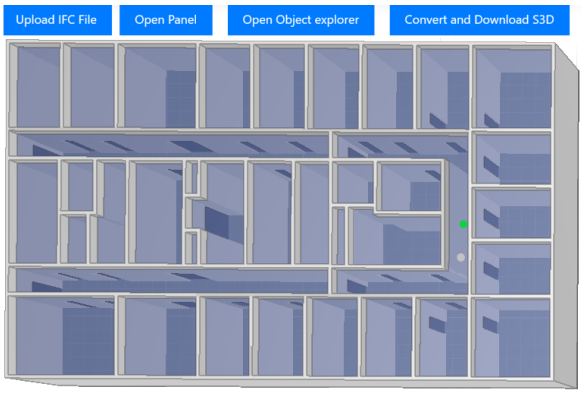

Figure 2: Web based tool for uploading an IFC file and converting it to Structured3D

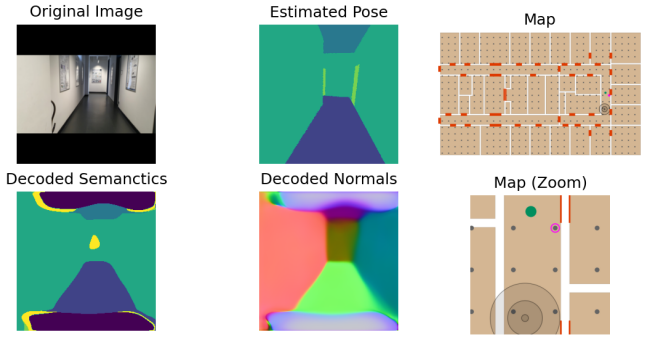

Figure 3: Exemplary result from SPVLoc

Figure 2 shows the website interface developed for the conversion pipeline. It allows users to upload an IFC file, explore the object hierarchy and properties, and convert the file into a Structured3D JSON format. For this work, an IFC model of the IC6 floor at Ruhr University Bochum was used as input. The resulting Structured3D JSON file was then used in SPVLoc, along with corresponding images, to perform localization within the environment. Figure 3 shows an exemplary result of this pipeline, detailing each stage of the SPVLoc localization process. The top-left image shows the original RGB input image captured in the hallway. This image is processed through the SPVLoc pipeline to infer semantic segmentation labels and surface normals, depicted in the decoded semantics and decoded normals views, respectively. The estimated pose is shown in the top-middle image, and uses these intermediate to predict the camera orientation and location relative to the building's Structured3D layout. The rightmost images illustrate the localization from a top-down view of the environment. The green dot indicates the predicted camera position, the pink dot shows the estimated orientation of the camera, and the grey dot marks the ground truth position derived from precise annotations. A zoomed-in view of the map is also included to focus on these points of interest. These results highlight

how the converted Structured3D data serves as spatial and semantic reference, enabling accurate and interpretable localization within the built environment.

5 Discussion

While the conversion of IFC model to the Structured3D format enables accurate localization, several practical challenges were encountered. The planeLineMatrix and lineJunctionMatrix, encode binary entries which can lead to large json files due to their quadratic growth as the number of planes, lines, and junctions increases. To mitigate file size, index-based tuples have been implemented to represent connections more compactly. ~~While the open-source version of SPVLoc does not utilize this format, the optimization is intended to support future use cases.~~

Another limitation arises from the use of mirrored or improperly aligned IFC models, which can resukt in inaccurate localizations. Despite these issues, the IFC-to-S3D conversion proved effective, as Structured3D representations were consistently generated, and localization based on them were accurate. The results demonstrate that 3D data extracted from IFC can successfully be used for camera localization or other applications which require spatial reasoning.

6 Conclusion

- short summary - future outlook (repetitive structures, maybe include more objects to ifc for semantics)

References

[1] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, *Structured3d: A large photo-realistic dataset for structured 3d modeling*, 2019. DOI: 10.48550/ARXIV.1908.00222.

[2] C. Mirarchi, M. Gholamzadehmir, B. Daniotti, and A. Pavan, "Semantic enrichment of bim: The role of machine learning-based image recognition", *Buildings*, vol. 14, no. 4, p. 1122, Apr. 2024, ISSN: 2075-5309. DOI: 10.3390/buildings14041122.

[3] N. Gard, A. Hilsmann, and P. Eisert, "Spvloc: Semantic panoramic viewport matching for 6d camera localization in unseen environments", in *Computer Vision – ECCV 2024*. Springer Nature Switzerland, Dec. 2024, pp. 398–415, ISBN: 9783031734649. DOI: 10.1007/978-3-031-73464-9_24.

[4] J. Kim, C. Choi, H. Jang, and Y. M. Kim, *Piccolo: Point cloud-centric omnidirectional localization*, 2021. DOI: 10.48550/ARXIV.2108.06545.

[5] H. Yu, W. Zhen, W. Yang, and S. Scherer, *Line-based camera pose estimation in point cloud of structured environments*, 2019. DOI: 10.48550/ARXIV.1912.05013.

[6] P. Yin, L. Xu, J. Zhang, H. Choset, and S. Scherer, "I3dloc: Image-to-range cross-domain localization robust to inconsistent environmental conditions", 2021. DOI: 10.48550/ARXIV.2105.12883.

[7] P. Pauwels, D. Van Deursen, R. Verstraeten, *et al.*, "A semantic rule checking environment for building performance checking", *Automation in Construction*, vol. 20, no. 5, pp. 506–518, Aug. 2011, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2010.11.017.

[8] T. H. Kolbe, “Representing and exchanging 3d city models with citygml”, in *3D Geo-Information Sciences*. Springer Berlin Heidelberg, 2009, pp. 15–31, ISBN: 9783540873952. DOI: 10.1007/978-3-540-87395-2_2.

[9] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and S. Hagen, *The GeoJSON Format*, RFC 7946, Aug. 2016. DOI: 10.17487/RFC7946. [Online]. Available: https://www.rfc-editor.org/info/rfc7946.

[10] R. B. Rusu and S. Cousins, “3d is here: Point cloud library (pcl)”, in *2011 IEEE International Conference on Robotics and Automation*, IEEE, May 2011. DOI: 10.1109/icra.2011.5980567.

[11] A. Aziz, N. Gard, M. König, P. Eisert, J. H. Heinbach, and L. Trost, “Visual fire safety inspection framework using computer vision algorithms”, *Journal of Computing in Civil Engineering*, vol. 39, no. 5, Sep. 2025, ISSN: 1943-5487. DOI: 10.1061/jccee5.cpeng-6492.

[12] A. C. Barreiro, M. Trzeciakiewicz, A. Hilsmann, and P. Eisert, “Automatic reconstruction of semantic 3d models from 2d floor plans”, pp. 1–5, Jun. 2, 2023. DOI: 10.23919/mva57639.2023.10215746. arXiv: 2306.01642 [cs.CV].

[13] H. Taira, M. Okutomi, T. Sattler, *et al.*, *Inloc: Indoor visual localization with dense matching and view synthesis*, 2018. DOI: 10.48550/ARXIV.1803.10368.

[14] S. Wang, S. Fidler, and R. Urtasun, “Lost shopping! monocular localization in large indoor spaces”, in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Dec. 2015. DOI: 10.1109/iccv.2015.309.

[15] C. Liu, A. Schwing, K. Kundu, R. Urtasun, and S. Fidler, “Rent3d: Floor-plan priors for monocular layout estimation”, in *CVPR*, 2015.