Multi-view fusion of technical drawings for a conceptual 3D reconstruction using deep-learning

Andrea Carrara[1]
[1]Lehrstuhl für Computergestützte Modellierung und Simulation, TU München, Arcisstr. 21, 80333 München, Deutschland
E-mail(s): andrea.carrara@tum.de

Abstract: Technical drawings, often in the form of floor plans, are required to design, define, and execute a building project. Still, it is becoming increasingly common to develop 3D models of the building defined in floor plans since it allows to perform simulations, spot structural flaws, and easily visualize the final result. The modeling task is not automatic; it requires less time and effort if the drawings are in a vector format, but often it is the case that the floor plan is rasterized, which means that the drawing is transformed into an image from the vector format in which it was produced, all the semantic information contained in the drawing are lost in the pixel transformation. Retrieving the lost semantic information frequently involves professionals in reconstructing the floor layout; it is time-consuming and prone to errors. Automatically generating models from floor plans is an in-depth researched topic. Although various researchers have addressed the topic, most studies' complexity remains low, focusing on parsing pre-processed or simplified datasets far from the actual floor plans used in the construction industry. The difficulty of creating adequate datasets partially explains this, annotating floor plans requires a significant amount of time, and the drawings are often copyrighted. The central lack in the research is creating a building 3D model based only on the main view, not considering cross-sections and the annotations of the measurements. Those are all essential information in the actual floor plans used in the construction industry that is always omitted in research. The task of automatically creating a realistic 3D model from multiple view floor plans is a novel research topic that has the potential to broaden the field of floor plan analysis. This research aims to identify a research gap and present two conceptual AI-based pipelines for completing this novel task for raster and vector data.

Keywords: Artificial Intelligence, Design Automation, Graph Neural Network, Multi-Source Linking

1 Introduction

A technical drawing, also known as an engineering drawing, is a thorough, accurate diagram or plan explaining how an object works or is built. Engineers, electricians, and contractors use these drawings as guidelines while constructing or repairing products and buildings. Therefore, technical drawings are a standard tool shared among engineering fields to convey knowledge using a common graphical language; they bridge communication between the designers and the producers.

Technical drawings are required to design, define, and execute a building project. Many types of architectural drawings are used in the construction process; each type represents specific details of the building. The floor plan, for example, shows the layout of the rooms and the placement of doors and windows; floor plans are a special kind of top view in which the roof is removed to show the rooms' disposition offering the wall layout. It is one of the most important tools to represent knowledge in building, especially in residential construction projects. In contrast, cross-section drawings show a vertical cut of the building providing vertical information about the building.

Complex projects contain many technical drawings to represent all the particles of the building or infrastructure in the finest detail. Professional architects and engineers create the drawings.

Combining the information from all the data from different sources is necessary to have the best comprehension of the project.

Traditionally all the drawings were designed and drawn on paper. Still, recently with the development of technology, with the introduction of computer-aided design (CAD) software, the standard for technical drawings changed from paper to computer-based drawings: the digitalization of the process allows to improve the accuracy of the drawing, reducing the time and errors. Consequently, the state-of-the-art practice in the construction industry is to design drawings in digital form with a CAD system and create 3D semantically rich BIM models to perform checks, simulations, and evaluations. This is straightforward in creating a new building, but it is necessary to deal with drawings on paper when the project is about an already existing building.

In this scenario, the conventional technique is to have experts reconstruct the drawings in digital form based on the paper version. It is time-consuming, error-prone, and has a large margin of automation potential.

Automatically generating models from floor plans is an in-depth researched topic. Initially, this was primarily accomplished with heuristic-based systems, but with the advancement of artificial intelligence, learning-based systems have taken over, outperforming prior methods. This is because, while floor plans have a standard fundamental structure depicting doors, walls, and windows in the case of buildings, the representation design of these elements might vary between projects.

This paper aims to highlight the research gap in technical analysis and define the problem statement. Two data pipelines based on the neural network are presented. The systems are designed to solve the problem of the most common representations of floor plans: raster and vector format.

2 Related Work

Former methods utilize heuristic-based methods derived from domain knowledge to extract the information from the floor plans and classic computer vision methods such as Hough Transformation to detect lines. Hough Transformation is a feature extraction technique used in computer vision. The HT transforms a difficult global detection task in image space into a more straightforward local peak detection problem in parameter space [1].
In “A System to Detect Rooms in Architectural Floor Plan Images” [2] a system to detect rooms in architectural floor plan images is described, they base their approach on the Hough Transformation, but they try to overcome the drawbacks by applying it on the result of image vectorization.
In Automatic reconstruction of 3D building models from scanned 2D floor plans [3], Lucile Gimenez et al. tackle the problem of constructing 3D building models from 2D plans using a topological approach. They start with a pre-processing phase of binarization and noise cleaning from the scanned plans. Subsequently, a separation of the input image in the text image containing textual information and the geometry image containing geometry elements like lines and arcs is performed.
More recently, many data-driven approaches were proposed for architectural elements detection and room classification of floor plans. Dodge et al. provide a technique for analyzing floor plan images utilizing object identification, semantic wall segmentation, and OCR (Optical Character Recognition) in "Parsing Floor Plan Images" [4]. They use a fully connected network for wall segmentation and Faster R-CNN to detect windows and doors.
Liu et al. in “Raster-to-Vector: Revisiting Floorplan Transformation” [5] focus on the vectorization of raster floor plan images with a learning-based approach but maintain an aspect of rule-based in the process by implementing an integer programming mechanism that refines the primitives extracted from the deep learning network.
Kim et al. use a style transfer technique to automatically deal with the diverse architectural styles and construct a simplified form of the floor plan image in "Application of Style Transfer in the Vectorization Process of Floorplans" [6]. The integrated format for the style transfer is a simplified geometrical representation of walls and openings. The network attempts to learn the representation of the standard simplified geometries in the floor plan, ignoring the details intrinsic to the type of plan.
Kalervo et al. in “CubiCasa5K: A Dataset and an Improved Multi-Task Model for Floor plan Image Analysis” [7] released a new dataset for automatic parsing of floor plan images containing 5000 floor plans annotated by experts. The authors present a multi-task network based on an encoder-decoder system to address the vectorization problem of the novel dataset.
In "FloorPlanCAD: A Large-Scale CAD Drawing Dataset for Panoptic Symbol Spotting" [8] Fan et al. present a new dataset it includes 15000 floor plans ranging from residential to commercial structures. The paper’s primary strategy is to use Graph Neural Networks to analyze vector data. Hence the graph must be constructed from the CAD drawings. They combine graph convolution based on the message passing mechanism with Faster R-CNN for the instance symbol spotting tasks.

3 Multi view technical drawing analysis and automatic data linking

3.1 Problem definition

The elements in different floor plans are represented differently depending on the perspective; the goal is to automatically link the items in multiple formats to acquire the most information for representation. The requirement originates from the large number of technical drawings used to depict engineering projects; the analytical focus is frequently on a single element, but examining the entirety of perspectives allows a deeper examination of the entire project. In the case of technical drawings, the process differs depending on the input source; two primary forms of representation are usually used: raster data constituted of pixels and vector data. For each type of input, one pipeline is provided, noting commonalities, advantages, and limitations.

3.2 Raster Data analysis

Raster data is a pixel representation; each pixel has a value that represents the pixel's color. The typical workflow for this type of data includes pre-processing the drawing, binarizing the colors, and cleaning the picture of noise. This approach frequently enhances the results of the pipeline's subsequent stages, allowing the learning algorithm to perform better.

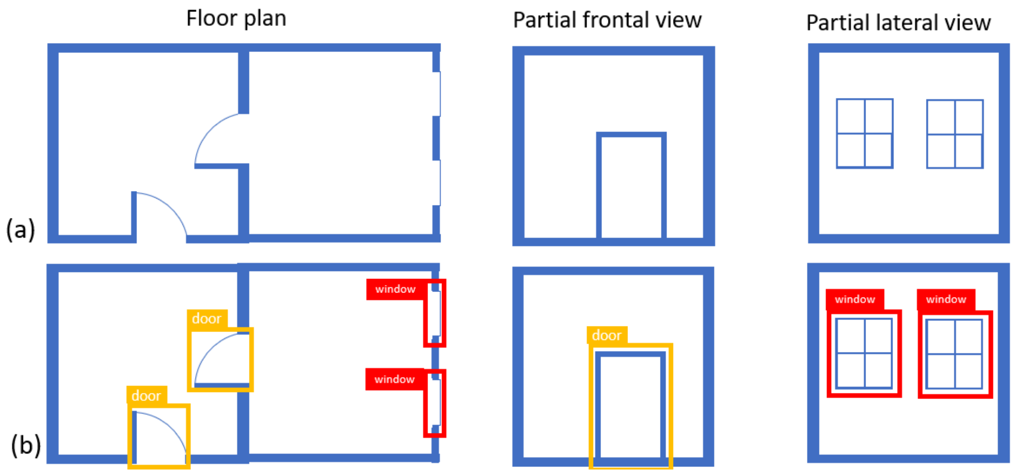

Figure 1: Object detection on multi view in technical drawings

The next stage in the processing is the detection of single elements based on the kind of view of the artwork. The floor plan needs to detect the walls and architectural components such as windows and doors from the top perspective; similarly, this categorization may be extended to other viewpoints of the structure.

For the raster data, two techniques are possible: semantic segmentation for walls and object detection on architectural parts or an entire object detection strategy. The goal of wall detection as a semantic segmentation task is to distinguish between pixels of walls and pixels of the background. Binary classification is performed on all pixels in the image, allowing for a general approach to any wall and shape because the neural network learns based on the proximity of pixels, and no geometrical function is used to fit the pixels. The deep learning model used in this assignment is often U-net or any autoencoder with skip connections, with input data in the same format as output data. Convolutional layers take features from the original input learning to represent it in dense space,. In contrast, deconvolutional recover information from the dense space to rebuild the original input or a detection mask defined in the dataset. In the case of the U-net design, certain skip connections are given to enhance the reconstruction loss when information is sent from the convolution layer to the deconvolution layer of the same dimension. The process of recognizing windows and doors on the picture is handled by region proposal networks like YOLO [9] or Fast R CNN [10].

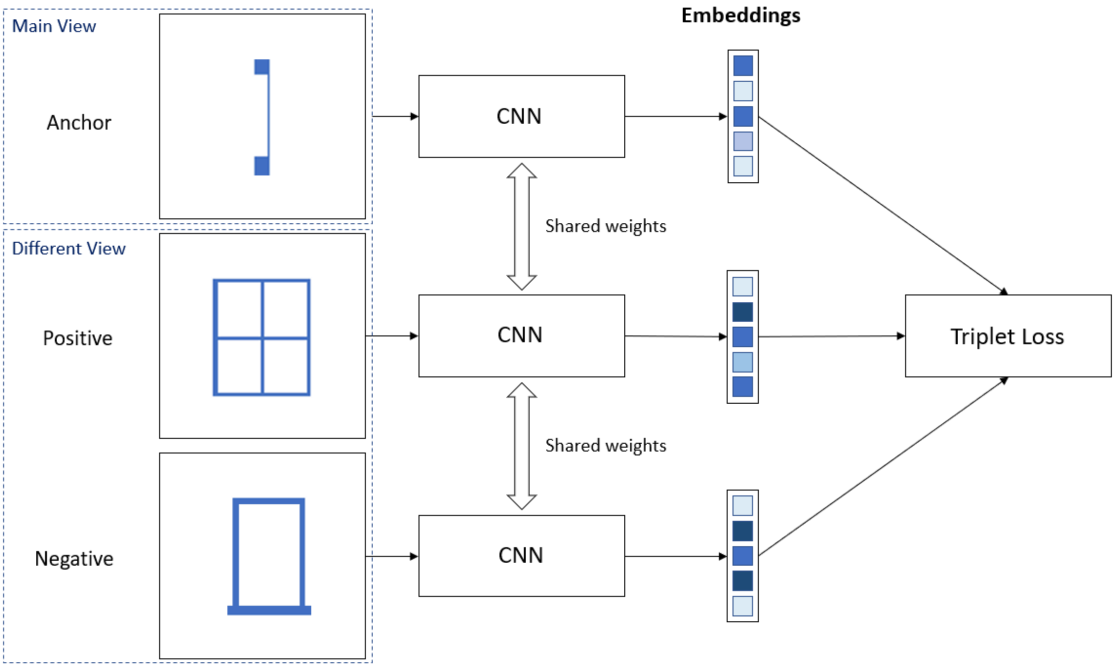

Figure 2: The triplet loss from multi view technical drawings

The alternative is to treat each element in the drawing as an element and perform object detection on the entire image, which means that either all of the walls are segmented and detected, or the distinction occurs at junctions. When the junctions are detected, the walls are connected to the points in a manner similar to graphs. This technique allows for the formation of a singular neural network, but the system loses the ability to analyse any type of wall shape.

The connecting between drawings is expressed as a similarity problem. The suggested method is based on triplet loss [11], which creates a differentiable loss function to compute image similarity. The fundamental concept is to use a Convolutional Neural Network with shared weights to transform an anchor, a positive and a negative picture into an embedding space, and then compute the distance between them automatically based on the training phase. The reference picture is the anchor, the positive image is another image that is equal to the anchor but in another view, and a negative image is a different object in another view. The triplet loss minimizes the distance to the positive while maximizing the distance to the negative. When performed on multiple instances the distances are

fine-tuned to express optimally the similarities in a multidimensional space.
Finally, the goal is to have objects from various perspectives that are near the same embedding space. Combining the triplet loss in semantic/object recognition in a multi-task learning approach might enable the training of an end-to-end neural network that identifies and automatically combines walls, windows, and doors, learning the relations between entities in the floor plan and different views.

3.3 Vector data analysis

Vector data is more popular in current drawing systems than raster data since the information is stored in line segments specified by mathematical functions rather than pixels. The various aspects of the design are frequently represented in distinct layers in CAD systems that can reflect the semantic value; however, this is not enforced by any standard and so it is not always viable to extract semantic information from it.
The vector data may be transformed into a graph format, where nodes represent vector segments and edges are constructed based on geometrical elements in the drawing (connectivity, parallelism, incidence of lines).

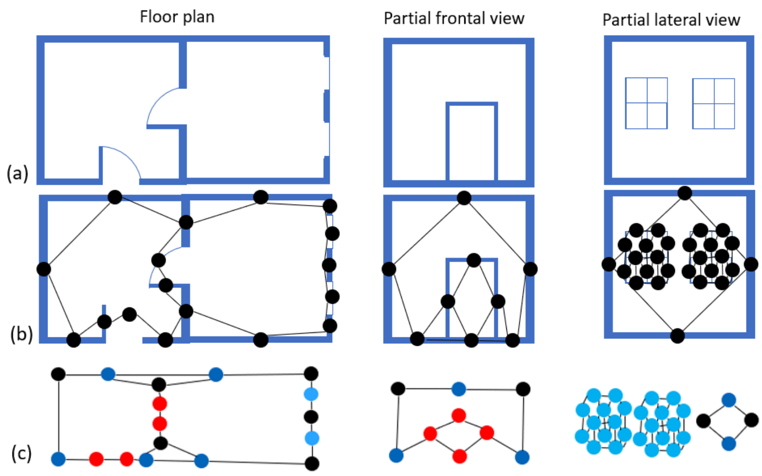

Figure 3: Subfigure (a) shows the input multi view technical drawings. Subfigure (b) depicts the transformation of vector data into the graph based on proximity and the subfigure (c) shows the node classification done by Graph Neural Network

The aggregate function combines information from nearby nodes and produces a message, while the update function combines the target node's embedding with the message to update the target node's new latent embedding vector.
In the update function, a set of weights W is introduced to enhance the embeddings by backpropagation of the loss function, and a nonlinear function is applied to the update function. These aggregate and update layers are stacked similarly to CNN convolutional layers, allowing the algorithm to learn longer node and edge relationships.
A graph connection problem can be used to define the multi-view connectivity job. The graph is built as a one-to-one translation of segments into graphs, and the node classification seeks to forecast each line into a class made up of distinct nodes. Following the initial classification, it is feasible to cluster

close nodes with the same label into a higher conceptual representation of the floor plan. Aggregating the nodes again, it is possible to obtain a representation of the layout in rooms and, if possible, the whole floor plan representation.

This enables the transformation of each graphic into a knowledge tree that begins with a single node and provides a more detailed representation of the architectural entities at each level. The drawings are connected at various levels, and a neural network may be trained to learn the graph's adjacency matrix, learning to automatically link components across drawings based on graph inter connectivity.

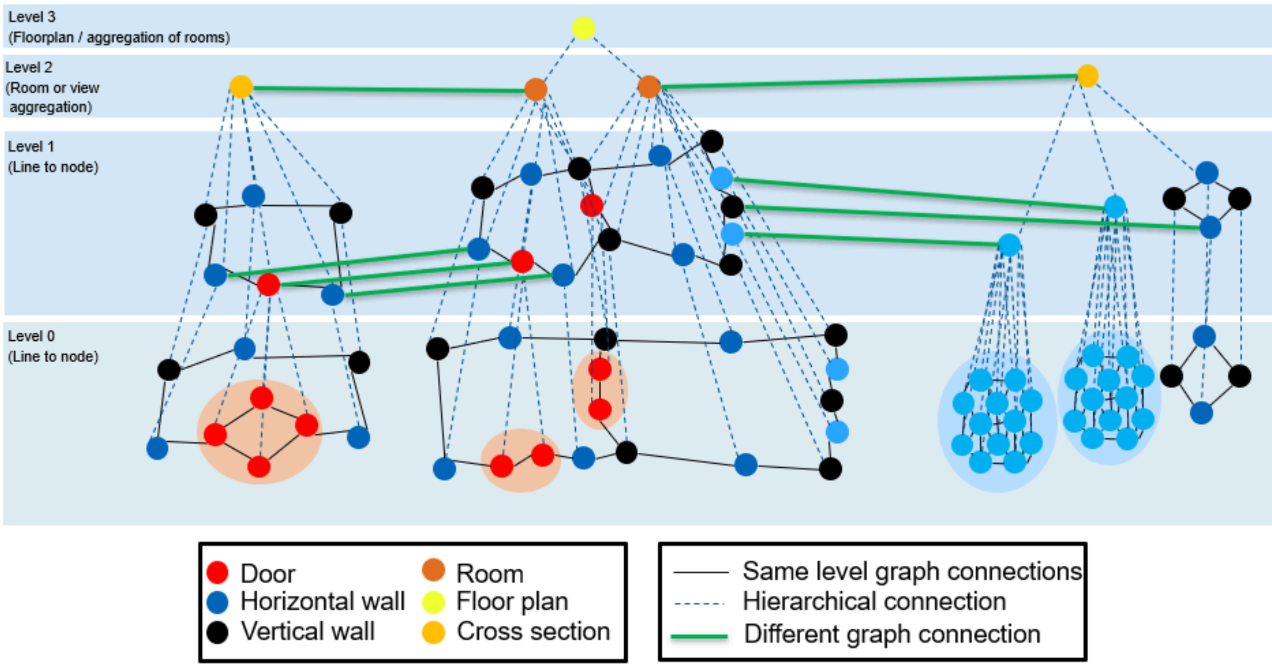

Figure 4: Hierarchical representation of technical drawings and multi view linking problem definition

This technique has possible implementation difficulties since it is hard to incorporate all of the tasks into a single network because the objective is constituted of node and edge classification. Also, it is important to specify or implement the hierarchy heuristic in the network. In any case, the suggested method has the capacity to accurately describe a technical drawing's hierarchical knowledge and link elements in a multi-view input.

4 Final Remarks

The report identifies a research gap in the field of technical drawing information extraction. To address this novel task, two data-driven techniques are offered; the approaches are theoretical due to the lack of available datasets for this sort of work. The emphasis is on data connectivity; nevertheless, transforming different perspectives in the 3D representation remains an open task.

References

[1] J. Illingworth and J. Kittler, “A survey of the hough transform”, *Computer vision, graphics, and image processing*, vol. 44, no. 1, pp. 87–116, 1988.

[2] S. Macé, E. Valveny, H. Locteau, and S. Tabbone, “A system to detect rooms in architectural floor plan images”, 2010, pp. 167–174, ISBN: 9781605587738. DOI: 10.1145/1815330.1815352.

[3] “Automatic reconstruction of 3d building models from scanned 2d floor plans”, *Automation in Construction*, vol. 63, pp. 48–56, Mar. 2016, ISSN: 09265805. DOI: 10.1016/j.autcon.2015.12.008.

[4] “Parsing floor plan images”, 2017.

[5] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, “Raster-to-vector: Revisiting floorplan transformation”, vol. 2017-October, 2017. DOI: 10.1109/ICCV.2017.241.

[6] “Application of style transfer in the vectorization process of floorplans”, vol. 114, Aug. 2018, ISBN: 9783959770835. DOI: 10.4230/LIPIcs.GIScience.2018.39.

[7] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, “Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis”, vol. 11482 LNCS, 2019. DOI: 10.1007/978-3-030-20205-7_3.

[8] Z. Fan, †. L. Zhu, H. Li, *et al.*, “Floorplancad: A large-scale cad drawing dataset for panoptic symbol spotting”. [Online]. Available: https://floorplancad.github.io/..

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2015. DOI: 10.48550/ARXIV.1506.02640. [Online]. Available: https://arxiv.org/abs/1506.02640.

[10] R. Girshick, *Fast r-cnn*, 2015. DOI: 10.48550/ARXIV.1504.08083. [Online]. Available: https://arxiv.org/abs/1504.08083.

[11] E. Hoffer and N. Ailon, *Deep metric learning using triplet network*, 2014. DOI: 10.48550/ARXIV.1412.6622. [Online]. Available: https://arxiv.org/abs/1412.6622.