Deep Floor Plan Analysis for Complicated Drawings Based on Style Transfer

Seongyong Kim[1]; Seula Park[2]; Hyunjung Kim[3]; and Kiyun Yu[4]

Abstract: This paper presents a novel approach to retrieve indoor structures from raster images of complicated floor plans. We extract the building elements in the floor plan and process them into a vectorized form to provide indoor layout information. Unlike conventional approaches, the proposed model is robust when recognizing rooms and openings surrounded by obscuring patterns, including superimposed graphics and irregular notation. To this end, we integrate various floor plan formats into a unified style using conditional generative adversarial networks prior to vectorization. This style-transferred plan that follows the unified style represents the room structure intuitively and is readily vectorized due to its concise expression. Raster-to-vector conversion is conducted with a combinatorial optimization in junction units of the layout. The experimental results demonstrate that when implemented with complex drawings, our model is comparable to existing methods in the detection and recognition of rooms and provides a much better score in one-to-one matches. **DOI: 10.1061/(ASCE)CP.1943-5487.0000942.** *© 2020 American Society of Civil Engineers.*

Author keywords: Indoor spaces; Floor plan analysis; Style transfer; Conditional generative adversarial.

Introduction

Recent advances in information technology have made it possible to acquire and process information within indoor environments, thus allowing for existing services to be expanded for indoor use (Liu et al. 2017b; May and Williams 2017). This has had a positive impact on facilities management (Yalcinkaya and Singh 2014), indoor localization (Rusli et al. 2016; Xu et al. 2018; Wei and Akinci 2019), and indoor data models (Volk et al. 2014; Kang and Li 2017; Konde et al. 2018). Above all, it has increased demand to construct fundamental databases of indoor information.

Among various approaches to generating 3D models of buildings (Gimenez et al. 2015), including those that use 3D laser scanning (Tang et al. 2010; Dimitrov and Golparvar-Fard 2015), CAD plans (Chen et al. 2008), architectural sketches (Fernández-Pacheco et al. 2012), and mobile applications (Sankar and Seitz 2012; Froehlich et al. 2017), this paper makes use of floor plans because they provide the most basic source of information for existing buildings. When approaches using floor plans or scanned paper plans are automated, models of indoor spaces can be created practically and efficiently because such plans are relatively inexpensive and easy to acquire. Due to the high accessibility of floor plans, Google Indoor Maps and OpenLevelUp have allowed users to build indoor maps using their own floor plans, but not with much detail. As such, automatic floor plan analysis is needed to make use of the practical availability of floor plans that exist in various formats. However, this is a challenging problem in a situation where (1) notation of different floor plans varies significantly according to the ordering institutions or the design offices, and (2) floor plans archived as raster images are usually characterized by complex, fuzzy architectural drawings.

Researchers in floor plan analysis have generated building models from scanned floor plans. They aim to convert raster images into vector models, and conventional approaches have set up rules based on local features within specific plans; the rules are thus highly dependent on floor plan format (Macé et al. 2010; Ahmed et al. 2011; Gimenez et al. 2015). To reduce the dependency on specific formats while addressing a larger variety of formats, researchers (de las Heras et al. 2014) have applied learning algorithms that are trained on the geometry of the patterns directly from the datasets (Dodge et al. 2017; Liu et al. 2017a; Jang et al. 2020). However, floor plan analysis has still been confined to a few concise formats that are simplified and abridged versions of architectural drawings. In addition, there is a trend toward a learning-based approach, and researchers tend to target simpler formats due to the ease of collection and labeling, even with an inability to address complicated formats.

Our goal is to extend the applicability of floor plan analysis to more complex and diverse formats (Fig. 1). We do so by reconstructing building models that represent room structure in a vector format from an EAIS dataset (Jang et al. 2020), which is composed of complicated, diverse architectural drawings. Wall and opening primitives can be reliably obtained from overlapping graphics and irregular patterns with our novel approach that converts multiple formats of floor plans into a unified format using a conditional generative adversarial network prior to vectorization. In the vectorization phase, we extract junction candidates by means of deep networks and find an optimal combination of the junctions and their connections based on combinatorial optimization. By confining the objective of the deep learning networks to style transfer rather than to extract geometric details, the deep learning networks can

[1]Reserch Fellow, Institute of Construction and Environmental Engineering, Seoul National Univ., Seoul 08826, Korea. ORCID: https://orcid.org/0000-0002-0774-6791. Email: syoi@snu.ac.kr

[2]Ph.D. Candidate, Dept. of Civil and Environmental Engineering, Seoul National Univ., Seoul 08826, Korea. ORCID: https://orcid.org/0000-0002-6059-4731. Email: seula90@snu.ac.kr

[3]Lecturer, Dept. of Civil and Environmental Engineering, Seoul National Univ., Seoul 08826, Korea. ORCID: https://orcid.org/0000-0003-4894-6906. Email: urbanistar@snu.ac.kr

[4]Professor, Dept. of Civil and Environmental Engineering, Seoul National Univ., Seoul 08826, Korea (corresponding author). Email: kiyun@snu.ac.kr

Note. This manuscript was submitted on March 18, 2020; approved on August 6, 2020; published online on December 17, 2020. Discussion period open until May 17, 2021; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, © ASCE, ISSN 0887-3801.

Fig. 1. (Color) Our proposed method can reconstruct walls in areas with overlapping graphics or nonuniform patterns, thus allowing room structures to be recovered even from complicated drawings: (a) input floor plan images and our results; (b) style-transferred plans; and (c) vectorized floor plans. See Fig. 12 for the vector-graphics representation.

recognize room structures, even from messy and ambiguous graphics. This paper allows for floor plan analysis to be accomplished with complex formats that were previously inaccessible and introduces a pragmatic novel annotation framework that can be used for deep learning.

Related Work

Research on floor plan analysis is specific to particular datasets because existing algorithms commonly rely heavily on certain patterns within floor plan notations. Thus, before reviewing relevant studies, we examined public datasets (Table 1). In general, floor plan datasets, except traditional CVC datasets, do not have a formal, unified format; instead, there are various formats within a general theme. This is because with the development of learning algorithms, each dataset expanded the range of formats it contained in order to increase the quantity. Each floor plan dataset has limitations concerning either quantity or notation complexity, and researchers opt to utilize the datasets that are suitable for their own purposes.

For such datasets to be useful for floor plan analysis, there must be pixel-wise annotations for objects such as walls, openings, and rooms. However, there are few public datasets because it is difficult for floor plans to be invariably labeled due to ambiguity in notation and the need for high-level expertise for object recognition (Macé et al. 2010; de las Heras et al. 2014). Even though several practical tools have been developed to conveniently annotate floor plans (Rendek et al. 2004; de las Heras et al. 2015), it is difficult to do so because there is no way to guarantee the same annotations from different annotators, especially for complicated floor plans.

Floor plan analysis is a combination of sequential processes that generate building models by automatically extracting meaningful information from rasterized floor plans. The overall procedures for preprocessing, line detection, wall and room recognition, and symbol recognition are well-organized (Gimenez et al. 2015). In this paper, based on the viewpoint that floor plan analysis can be applied universally through learning algorithms (de las Heras et al. 2014), we review the relevant studies in the flow from a rule-based to a learning-based approach.

In general, the process of floor plan analysis can be categorized as graphic separation, pattern recognition, and floor plan vectorization. Graphic separation, as preprocessing for pattern recognition, extracts essential graphic elements from floor plans. In pattern recognition, building elements such as walls and openings are recognized by graphic patterns—process is the core of the floor plan analysis that researchers aim for. Floor plan vectorization is used in a narrow sense in that building elements are converted into an intact vector form with respect to buildings rather than simply changing the data format to a vector. Table 2 summarizes the research that is widely referred to in floor plan analysis.

Conventional floor plan analysis using a rule-based approach is characterized by the use of specific geometric patterns in floor plan notation for object recognition. Typically, preprocessing of separating graphic elements was carried out as a first step, and especially in CVC, graphic elements were readily separated by distinguishing between lines of different thicknesses. Then, (Macé et al. 2010) extracted walls by searching for parallel contours of thick lines based on Hough transform and detected rooms as convex regions after recursive decomposition. To improve performance, (Ahmed et al. 2011) found the contours of walls after subdividing the graphics further into thick, medium, and thin lines. Also, to detect rooms,

Table 1. Floor plan datasets

Dataset (references)	Annotation (quantity)	Theme	Property
CVC (Macé et al. 2010; de las Heras et al. 2014)	Walls, doors, windows, rooms without type (122)	Simplified architectural drawings	• High-resolution architectural drawings with uniform notations • Mixture of four formats; but most are biased in one format, BlackSet that expresses a wall with thick black lines (more than 75%) • Format diversity = very low (only 4) • Notation complexity = low-medium • Information amount = medium
Rakuten (Liu et al. 2017a)	Walls, openings, room types, icon types (815)	Simplified real estate floor plans including color	• RGB color that represents the type and boundaries of the rooms • Simplified layouts mostly in a rectangular shape with stereotyped symbol and icons • Walls of uniform thickness and highlighted expressions • Format diversity = medium • Notation complexity = low • Information amount = high
Rent3D (Liu et al. 2015; Zeng et al. 2019)	Walls, opening, room types (232)	Simplified real estate floor plans	• Relatively irregular layouts including round-shaped ones • Walls with nonuniform thickness, but still highlighted • Format diversity = medium • Notation complexity = low • Information amount = high
EAIS (Jang et al. 2020)	Walls, doors (319)	Architectural drawings	• Various drawings including architectural, structure, electrical, plumbing plans with different notations • High complexity due to messy and overlapping graphics • Walls with irregular notation and unclear boundaries • Format diversity = very high • Notation complexity = very high • Information amount = low

Table 2. Representative research in floor plan analysis

Category	Paper	Datasets	Graphic separation	Pattern recognition	Floor plan vectorization
Rule-based	Macé et al. (2010) and Ahmed et al. (2011)	CVC (partial)	$\vee$	$\vee$	—
	Gimenez et al. (2016)	CVC (partial)	$\vee$	$\vee$	$\vee$
Learning-based	de las Heras et al. (2014)	CVC	$\vee$	$\vee$	—
	Dodge et al. (2017)	CVC/Rakuten	—	$\vee$	—
	Zeng et al. (2019)	Rent3D/Rakuten			
	Liu et al. (2017b)	Rakuten	—	$\vee$	$\vee$
	Jang et al. (2020)	EAIS			

Note: $\vee$ indicates whether the research covers each process.

gaps between the walls were closed by actively spotting doors and windows by means of SURF, and empirical thresholds were additionally used for insufficient cases.

Gimenez et al. (2016) went one step further to floor plan vectorization, the output of which was refined and imported into IFC format. After separation of graphics using Tombre et al. (2002), various types of building elements were detected based on structural rules designed in a form, like assigning the wall to two parallel lines within a certain distance of each other. Finally, 3D building models were generated by properly assembling the vectorized building elements. Overall, these approaches using rule-based algorithms depend heavily on notation and on empirical parameters, and they perform well in specific formats but have limitations in coping with others.

The learning-based approach, which directly learns geometric patterns from floor plan datasets, has been applied to various floor plan formats. In the early learning approach (de las Heras et al. 2014), graphic separation was required, followed by the coarse segmentation of each object based on a bag of visual words. Then, the segmented building elements were refined and vectorized in order to detect room structures. However, because the model was trained directly from the datasets, the analysis still had to be applied separately on data that used the four notations in CVC.

As deep learning is applied to floor plans analysis, the role of the learning algorithms has expanded, and graphic separation can be omitted with raw floor plan images used directly to train the models. Dodge et al. (2017) built models that coped with several formats at once via deep networks, thus increasing the versatility of floor plan analysis. The key was to detect walls using fully convolutional networks (Long et al. 2015) and deep learning networks specialized in segmentation and to visualize them in 3D by extruding the raster outputs. The analysis targeted CVC and Rakuten datasets, but for CVC, K-fold cross validation had to be used because of the limited quantity of data. Zeng et al. (2019) further developed such analysis and improved its performance by designing a network architecture that referred to the room boundary as attention when extracting rooms and their types. In addition, unlike in previous studies, the researchers targeted different types of floor plans including walls of uneven thicknesses or curved walls, but they still did not cover vectorization of the raster outputs in their scope.

In a learning-based approach, floor plan vectorization is generally accomplished by refining and processing the output of pattern recognition based on learning models. To vectorize floor plans, Liu et al. (2017a) extracted semantic information from basic geometric patterns using ResNet (He et al. 2016) and generated candidates for the junction and wall primitives. Then, they formulated integer programming to find the optimal primitive pair that correctly represented an indoor structure and finally reconstructed the final building models consisting of vectorized walls and openings. Even though learning algorithms have been extended to floor plan vectorization, their approach was specific to Rakuten data, which consist of an abridged version of a floor plan expressed using color-coded information and simplified walls with a uniform pattern. Jang et al. (2020) also covered floor plan vectorization, specifically targeting an EAIS dataset composed of complicated floor plans. Through segmentation using deep networks, walls and doors were detected from various complex formats of the drawings. Then, the extracted wall segments were vectorized by means of skeletonizing and several postprocessing tasks based on corners. For the door segments, the proper position in a vectorized wall was searched and filled in with a line. By targeting complicated floor plans, the researchers followed existing methods using segmentation and contained ad-hoc processing and heuristic parameters in the vectorization.

Overall, a learning-based approach, especially one based on deep networks, has steadily been expanded due to its versatility in addressing various formats. However, most studies have targeted datasets that are limited in quantity and notation complexity, and they also included specific processing steps that could not be generalized to other formats. Thus, our objective is to develop a learning-based model that can cover floor plan vectorization in a universally applicable way, even targeting complicated floor plans.

Analysis of Complicated Floor Plans

Conversion to a Unified Style via Style Transfer

In order to (1) recognize significant primitives from complicated and overlapping graphics, and (2) acquire a uniform level of detail (LOD) from varied types of drawings in EAIS, we transformed the format of each floor plan into a simple, unified format through a style transfer technique prior to vectorization (Fig. 2). We designed our own floor plan format, namely a unified style, that is intended for style transfer and can integrate and represent diverse types of drawings.

The unified style was carefully designed considering our objective to reconstruct room structure and convert floor plans into this style by means of learning-based generative models. When designing the unified style, we placed a great deal of importance on uniform LOD and a resemblance to the original notations. We considered the following factors: usability as indoor information, commonality for diverse types of drawings, and ease of vectorization. As a result, our unified style represented walls and openings in a simplified form, with the primary goal of representing indoor spaces based on the room unit.

Fig. 3 depicts the features of the unified style by comparing the wall of the original floor plan and the unified style. In terms of preventing significant deformation of the walls and preserving the overall room structure, the floor plan is sufficiently simplified and abridged. Specifically, in the unified style, spaces that are insignificant in terms of representing the room structure are omitted. The walls are roughly expressed as trimmed and flattened shapes rather than with precise geometry, while the openings are expressed by following the original arc shapes. Due to this expression, even when starting with complicated floor plans with ambiguous notational criteria, the unified style can represent rooms intuitively and clearly and is robustly annotated by distinct annotators in terms of presenting room structure (Fig. 4).

In order to transfer the styles of the floor plans into designated styles, we utilized the conditional generative adversarial network (Isola et al. 2017), which aims to convert input images into a style of annotation pairs. Ever since Goodfellow et al. (2014) presented the generative adversarial network, there has been tremendous development in generative models and neural style transfer (Gatys et al. 2015; Jing et al. 2019). Especially notable among various models for style transfer based on generative adversarial network (Kim et al. 2017; Zhu et al. 2017), the conditional generative adversarial network has the characteristic of translating images into the desired style using annotation pairs, and it also has the strength

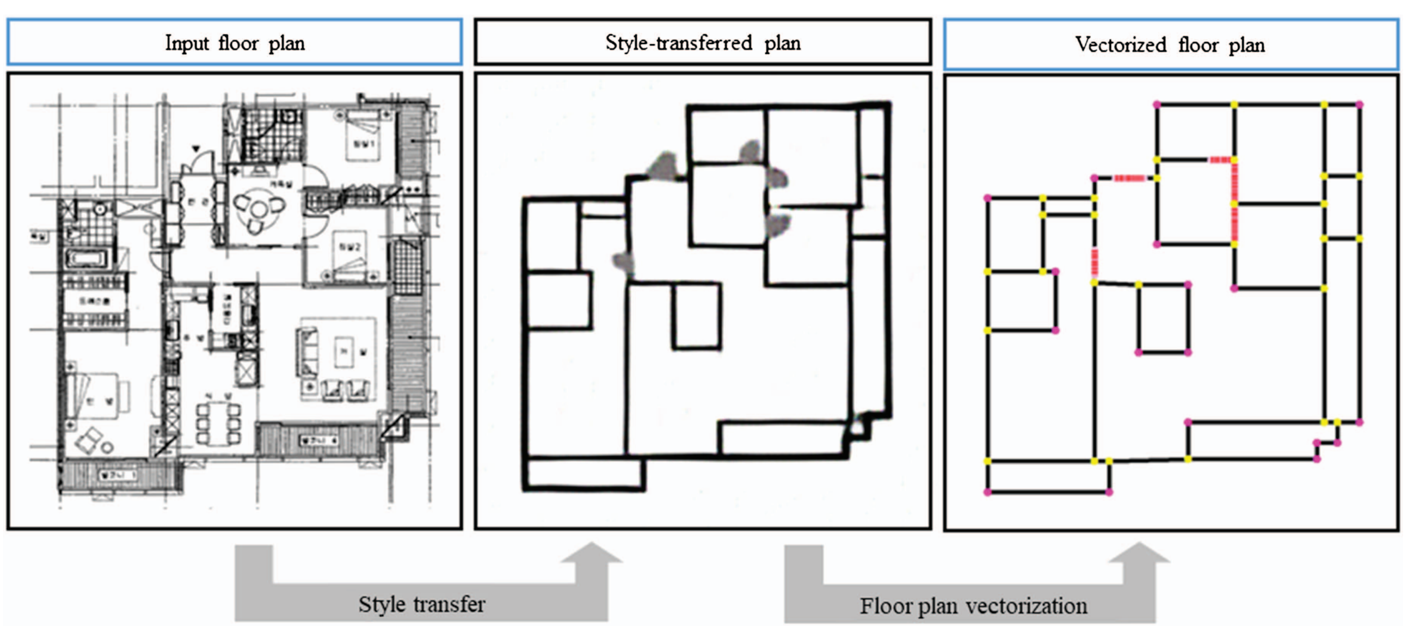
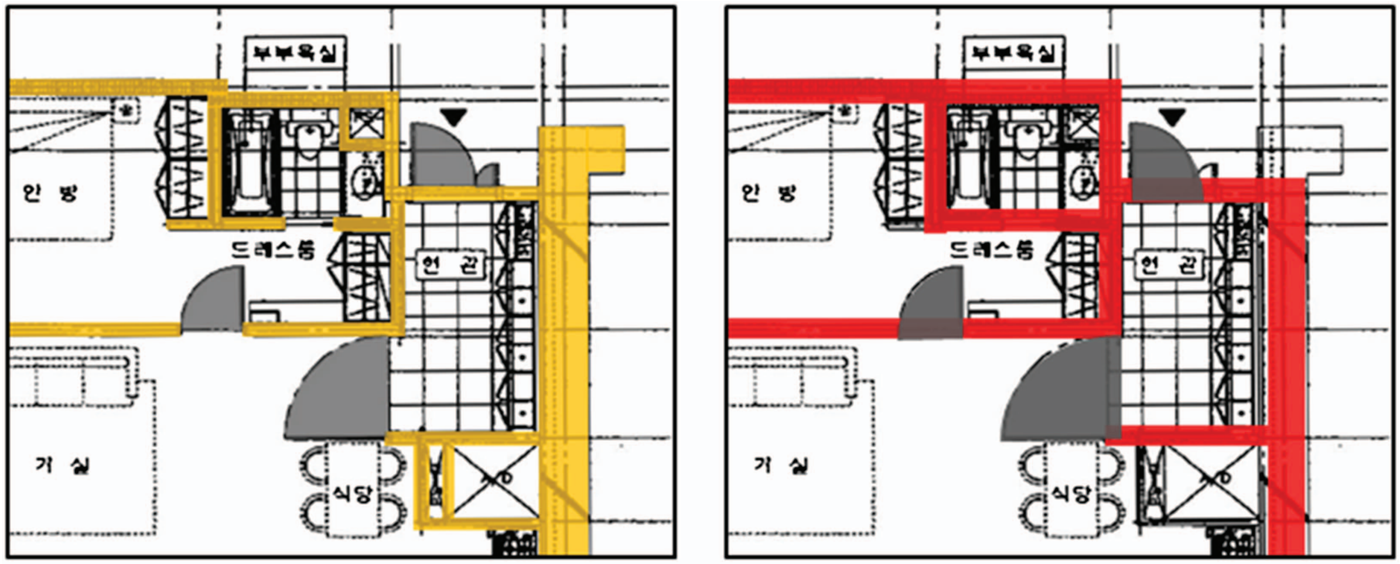
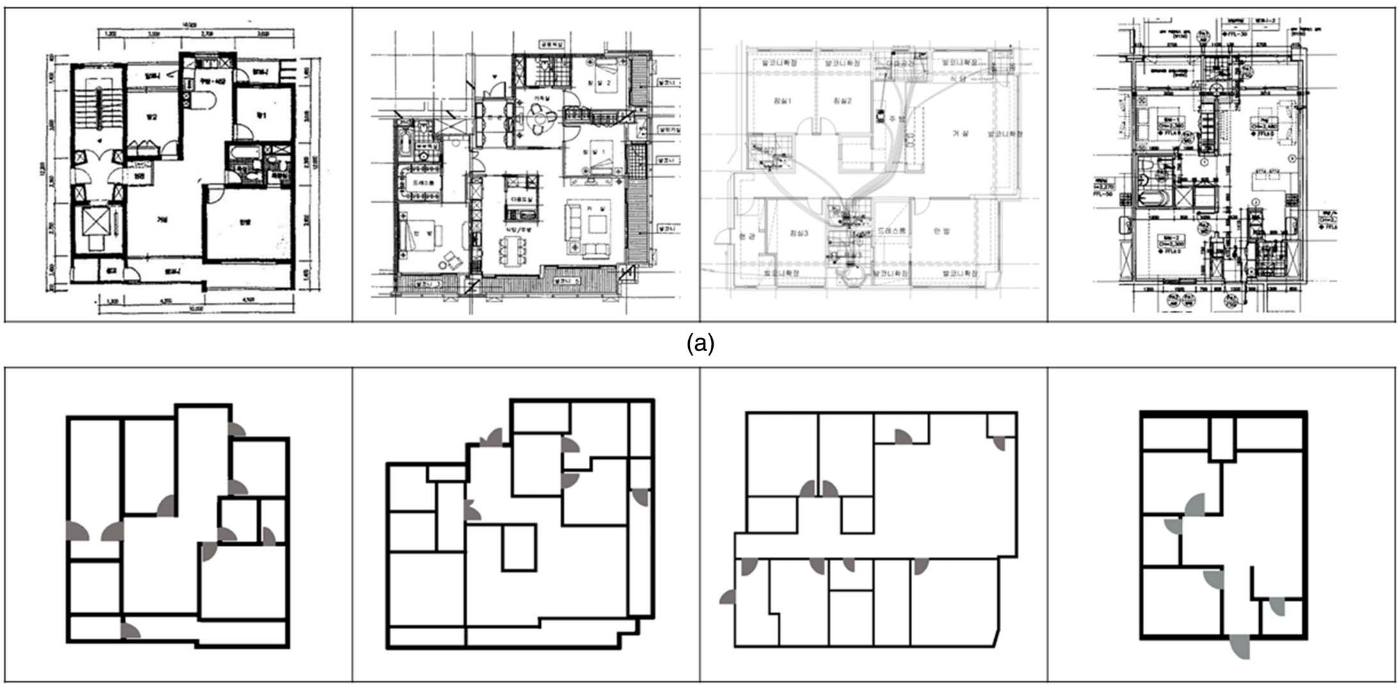


Fig. 2. (Color) Design process for analyzing complicated floor plans.



(a) (b)

Fig. 3. (Color) Comparison of (a) exact geometry of walls; and (b) walls in the unified style.



(a)

(b)

Fig. 4. Examples of the unified style: (a) floor plans; and (b) corresponding annotations.

to preserve the underlying structure of the images. These characteristics align well with our objective of converting floor plans into the designated style while preserving room structure. A detailed description of the deep networks used is depicted in the "Extraction of Preliminary Materials via Multitask Deep Networks" section.

Floor Plan Vectorization via Combinatorial Optimization

For floor plan vectorization, we essentially borrowed an idea introduced by Liu et al. (2017a), which is a stratified representation that creates high-level building models through sequential conversion over low-level geometries. The core of the idea involves combinatorial optimization that searches for a proper combination of candidate wall and junction primitives by assembling the junction and semantic information extracted using deep learning networks. We followed their approach because it performed well with simple notation and is compatible with our style transfer using deep learning networks.

We remodeled the stratified representation to be universally applicable even for floor plans that are not drawn in detail and lowered the barrier to utilizing deep networks by mitigating the

Table 3. Comparison of the number of annotation classes between Liu et al. (2017a) and the present method

Annotations	Liu et al. (2017a)	Present method
Junction	21 types	5 types (I, L, T, X, O)
Object	22 types (12 rooms + 10 icons)	3 types (wall, opening, background)

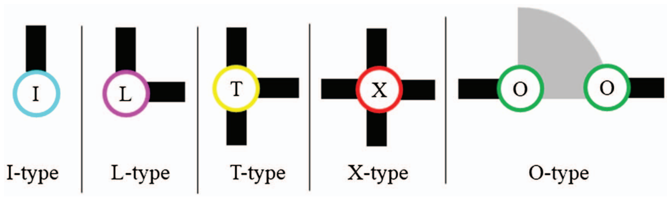


Fig. 5. (Color) Annotation of junction types.

difficulties with annotation. For example, we targeted only objects that are typically contained in architectural drawings and significantly lowered the number of classes in annotation (Table 3). Specifically, in the case of a junction, instead of the original 21 classes with orientation, we use five classes that only indicate the degree of connectivity. I-, L-, T-, and X-type indicate the number of connected wall segments (1–4, respectively), and O-type indicates the two ends of an opening (Fig. 5).

By reducing the junction classes from 21 to 5, we lost directional information of junctions and should endure the computation burden at the combinatorial optimization phase, and in return, we could obtain practicality at the deep network phase. In detail, this significantly reduced the annotating labor and had an effect of increasing the number of data per class, requiring less training floor plan. In addition, the lack of directional information means that a high degree of freedom is guaranteed at the optimization phase, so there is room for improvement in the building elements incorrectly extracted from the deep networks. At the same time, the information lost from reduced classes was fully complemented at the optimization phase by reflecting the style-transferred plan into the optimization formula. As such, we modified the stratified representation in the direction of practicality and then implemented floor plan vectorization on our style-transferred plans using combinatorial optimization. The programming modifications including a new objective function and relaxed constraints are depicted in the "Floor Plan Vectorization via Combinatorial Optimization" section.

Implementation of the Proposed Method

The process designed to analyze complicated floor plans, which are composed of sequential steps of style transfer and floor plan vectorization, is implemented in two stages: multitask deep networks and combinatorial optimization (Fig. 6). By deploying multitask deep networks, we simultaneously achieved both objectives of transferring the styles of the floor plans and extracting the geometric features to be used to generate primitive candidates. Based on these outputs, a style-transferred plan and the primitive candidates, such as junctions and walls, we used combinatorial optimization to find an optimistic primitive combination that represented a structure that was similar to the style-transferred plan while satisfying the constraints related to the floor plan layout. Specifically, for each

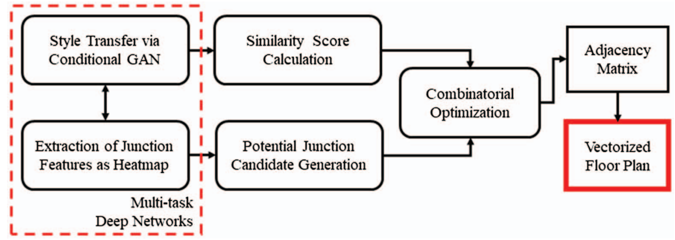


Fig. 6. (Color) Proposed process.

primitive candidate, a similarity score that compares it to the style-transferred plan is calculated and used to formulate an objective function that maximizes the similarity scores. Then, we visualized this final result expressed as an adjacent matrix after simple postprocessing. In the "Extraction of preliminary materials via multitask deep networks" section, we address the first stage, multitask networks, including network architecture and training techniques. We describe the detailed configuration of the combinatorial optimization in the "Floor Plan Vectorization via Combinatorial Optimization" section.

Extraction of Preliminary Materials via Multitask Deep Networks

By default, the junction annotations were worked on in the unified style, a concise version of the plans. In other words, in the annotation phase, junction extraction proceeded sequentially after the floor plan was converted to the unified style. However, in the phase of deep network training, our designed networks provided both outputs simultaneously through end-to-end training. In this manner, the deep networks could complementarily improve the performance of the style transfer and junction extraction while also significantly reducing redundant computations used to extract low-level features from the floor plans.

We utilized multitask deep learning networks to perform the following tasks: (1) style transfer that converts the formats of the floor plans into a unified style, and (2) extraction of junction features that indicate the position and type of each junction in the form of a heatmap. To this end, we constituted a network structure by fusing the conditional generative adversarial networks (Isola et al. 2017) with the junction-layer conversion from (Liu et al. 2017a). Both outputs were used as preliminary materials for subsequent vectorization.

The overall architecture of the deep networks is illustrated in Fig. 7. From the perspective of conditional generative adversarial networks, the generator is extended to output additional junction maps in parallel, while from the perspective of the stratified floor plan representation, the per-pixel room classification is substituted into the style-transferred plan. The frame of the networks is based on each source, but the generator, which is present on both sides, follows the junction layer, which is suitable for intricate tasks due to the existence of deeper layers. In detail, the generator network is based on ResNet (He et al. 2016) and is modified in a direction that raises the resolution of the output (Bulat and Tzimiropoulos 2016) (Fig. 8). The ensuing scoring layers at the back are stacked in the proper forms for each purpose (integrated style and junction maps). As for the discriminator, the convolutional patch generative adversarial networks classifier is used, and the required parameters such as patch size are borrowed from Isola et al. (2017).

To optimize the combined networks for different purposes, the objectives of converting to the integrated style and creating junction maps are numerically compiled. When training the style transfer

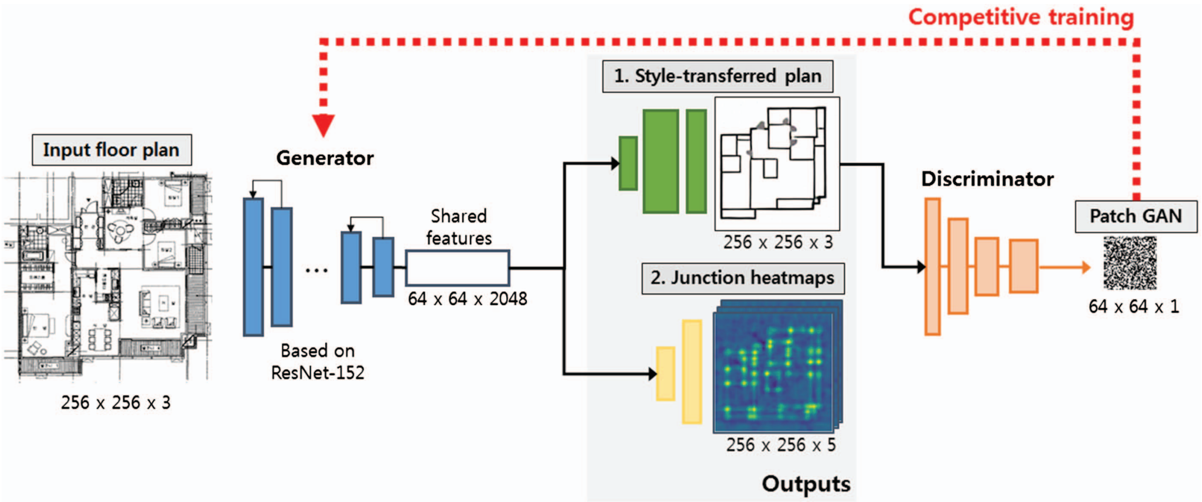


Fig. 7. (Color) Overall network architecture.

	Type	Filters	Sizes	Output
	Convolutional	64	7 x 7 / 2	128 x 128
	Pooling		3 x 3 / 2	64 x 64
3 x	Convolutional	64	1 x 1	
	Convolutional	64	3 x 3	
	Convolutional	256	1 x 1	
	Residual			64 x 64
8 x	Convolutional	128	1 x 1	
	Convolutional	128	3 x 3	
	Convolutional	512	1 x 1	
	Residual			64 x 64
36 x	Convolutional	256	1 x 1	
	Convolutional	256	3 x 3	
	Convolutional	1024	1 x 1	
	Residual			64 x 64
3 x	Convolutional	512	1 x 1	
	Convolutional	512	3 x 3	
	Convolutional	2048	1 x 1	
	Residual			64 x 64
	Shared features:			2048 x 64 x 64

Fig. 8. (Color) Architecture of the generator; note that the shaded parts are modifications from the original ResNet-152 to improve the resolution of the output.

alone, the objective of the minimax problem is solved by alternating each gradient descent step of the divided max and min problems. For conversion to the junction map, an objective of the regression is minimizing pixel-wise sigmoid cross-entropy loss. We assembled the minimization part of the two objectives while updating only the weights in relevant layers. Overall, like a fractal structure, when each learning step for the style transfer trains in turn, the minimization problem is also subdivided into two alternating steps. Thus, the generator, instead of being updated, just fools the discriminator, and it additionally reflects the loss for the junction maps.

The training detail is as follows. The pixel size of the input and output is fixed at 256×256. The input images are resized to this resolution while maintaining their aspect ratio, which includes the core importance of the floor plan, and then for data augmentation, techniques like random cropping, random rotation in 90-degree units, and random flip are carried out. We carried out the same process on the corresponding junction maps. When training was complete, the results from the deep learning networks were both the style-transferred plan as a raster image and the junction maps as stacked heat maps. These are delivered to the next section.

Floor Plan Vectorization via Combinatorial Optimization

By means of integer programming, we assembled intermediate products generated by the deep networks. Each product is incomplete to embody the vector structure, and the products complemented each other to construct the building models of the floor plan. Specifically, we built primitive candidates of the walls and openings from the junction maps and then spotted a correct subset of them while referring to the style-transferred plans. To arrive at the optimal solution of the integer programming, we used a Gurobi solution, which took under 3 s even for a case that had the most candidates in EAIS.

Candidate Generation

We first set a global threshold to determine alignment. Uncertainties of the exact locations of the junctions accumulate from human error in annotation. We assigned five pixels (from a 273×273 image) to the tolerances for this type of error, which is half the size used in Liu et al. (2017a), in order to elaborate the complicated structure of the EAIS data. For instance, we considered the junctions within the global threshold to be overlapping, and we regarded the primitives made by the junctions aligned within the threshold as being parallel to the axial direction.

Prior to generating the candidates, we transformed the junction maps into a junction set that contained pixel coordinates and a class for each junction (Fig. 9). To convert the heat maps to point sets, we applied nonmaximum suppression that left only the highest probability output in the neighborhood. Based on the junction set, candidates for the wall and opening primitives are generated when their potential existence is qualified in the style-transferred floor plan. In detail, each primitive can be formed by two junctions when the nearby pixels of the generated primitive are primarily classified as the corresponding class in the style-transferred floor plans.

Integer Programming

We borrowed the base frame of integer programming from Liu et al. (2017a) but adjusted it to apply to the style-transferred plan. The overall details of the variables, an objective function, and the constraints are given in Fig. 10.

J. Comput. Civ. Eng., 2021, 35(2): 04020066

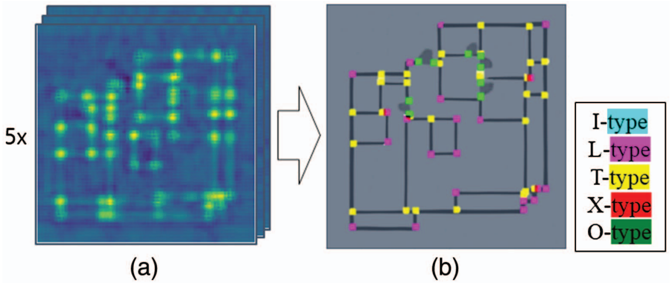


Fig. 9. (Color) Generation of junction candidates: (a) junction heatmaps; and (b) visualized junction candidates superimposed on the style-transferred plan. The legend shows the color used for each junction type.

The variables are confined to walls and openings, and binary indicators of the junctions and primitives are defined for each. For instance, '1' indicates the presence of a junction, and '0' indicates the absence of a junction. Since our combinatorial optimization faithfully adheres to the basic conversion of the style-transferred plan into a vector format as it was originally shaped, the objective function is formulated to retrieve the indoor structure, similar to the style-transferred plan, while using the junctions extracted from deep networks as much as possible. Specifically, the objective function is to maximize the linear combination of all variables. The weights of the junctions are a degree of connected primitives that allow as many intersecting primitives as possible to be used to concisely express the floor plan; the same structure can be expressed in fewer junctions. The weights of the primitives include a similarity score that allows a comparison with the style-transferred plan, quantified by the proportion of classified pixels within the global threshold. Note that this similarity score is newly calculated for the combinatorial optimization and is not relevant to the loss used for updating styles in multitask deep networks.

The constraints are as follows. (1) The loop constraint is confined to closures at the whole building level, instead of rooms, because we only had wall information, and in order to keep buildings closed, there must be more junctions than walls. (2) The connectivity constraint controls the compatibility between the junctions and the primitives in the same way as one-hot encoding. When a junction does not exist, the primitives connected to it must not exist either. In addition, each junction must be connected to a certain number of primitives corresponding to its classes. (3) The mutual exclusion constraint prevents overlapping geometry, so the junctions or primitives that are spatially close within the global threshold cannot be chosen at the same time. (4) The opening constraint enforces that openings can be selected only when wall primitives are close.

After solving for the integer programming, we took the solution that indicated the floor plan structure through a combination of junctions and primitives. Then, we conducted postprocessing for good alignment of the walls and openings. We gathered the

Binary variables

- $\boldsymbol{J^{wall}}, \boldsymbol{P^{wall}}$ for walls
- $\boldsymbol{J^{open}}, \boldsymbol{P^{open}}$ for openings

Objective function

$$Max.\,Z = \sum_{\substack{wall,\\ open}} \left[\sum_j D_j \cdot J_j + \lambda \sum_p \omega_p^{sim} \cdot P_p \right]$$

Constraints

(1) Loop

$$\sum_p P_p^{wall} \leq \sum_j J_j^{wall}$$

(2) Connectivity

$$\sum_j A_{ij} = D_i \cdot J_i \text{ for } \forall i$$

(3-1) Mutually exclusive of junctions

$$\frac{1}{n} \sum_j M_{ij}^J J_i \leq (1 - J_i)\, for\, \forall i$$

(3-2) Mutually exclusive of primitives

$$\frac{1}{m} \sum_j M_{kj}^P P_k \leq (1 - P_k)\, for\, \forall k$$

(4) Opening

$$P_o \leq \sum_{k \neq o} M_{ok}^{Open} P_k \;\; for \;\; \forall \;\; o$$

- n, m = Total number of junctions and primitives
- λ = Ratio of junction score to primitive score
- **D** = Degree of junctions' class, where D_i is number of connected lines in i^{th} junction
- ω^{sim} = Similarity score for the style-transferred plan for each primitive
- **A** = Adjacency matrix, where A_{ij} is 1 if i^{th} and j^{th} junctions form the primitive, 0 otherwise
- $\mathbf{M^J}, \mathbf{M^P}$ = Mutually exclusive matrix for junction and primitive, respectively, where M_{ij}=1 if i^{th} and j^{th} components are overlaid, 0 otherwise

Fig. 10. Details of variables, objective function, and constraints.

junctions in a straight line by searching for connected walls that were not regarded as diagonal, and we adjusted their position to the average location. We also moved the openings to stick to the closest walls. Eventually, we constructed building models that represented the integrated style in a vector format.

Evaluation

Floor Plan Dataset

Since our objective is to recognize room structure from complicated formats, we targeted the EAIS dataset (Jang et al. 2020). Compared with other datasets, such as CVC, Rakuten, and Rent3D, which are typically composed of abridged versions of plans, EAIS contains various types of architectural drawings characterized by complicated and fuzzy styles. To be specific, the walls in EAIS floor plans are expressed in nonuniform patterns, and the boundaries are unclear due to overlapping graphics and noise (Fig. 4). To the EAIS dataset, in addition to the 319 original floor plans, we added images until there were 450 from the same data source (Jang et al. 2020) to increase quantity and format diversity. We manually annotated the floor plans ourselves because we proposed a new annotation, namely the unified style. To train-test the split ratio, we split our updated EAIS data into 400 images for training and 50 images for testing while keeping diverse types evenly distributed. The test images were newly selected because there is no test set used on EAIS data to evaluate the vector results.

In addition, we utilize BlackSet in CVC as a reference dataset since it has been widely used in conventional fields. BlackSet consisting of abridged architectural drawings in a single and simple format has a clear and consistent notation expressing the wall with black thick lines. Compared to EAIS in the analysis of the floor plan, building elements could be easily detected and recognized in BlackSet (no format diversity; no overlaid graphic; no ambiguous expression; see Table 1).

Table 4. Evaluation of the room-detection task. Ours and Ours* indicate the results of the style-transferred plan and the vectorized floor plan, respectively

	EAIS		CVC (BlackSet)
Metrics	Ours	Ours*	(de las Heras et al. 2014)
DR (%)	88.37	87.87	94.76
RA (%)	90.90	89.96	94.26
One to one rate (%)	80.89	81.32	57.68
One to many count	0.26	0.28	1.34
Many to one count	0.5	0.48	2.24

Quantitative Evaluations

The evaluations of the floor plan analysis were dominantly performed through a wall segmentation task or a room-detection task (de las Heras et al. 2015). In this paper, we assessed the results of both the style-transferred plan and the vectorized floor plan by means of a room-detection task since our goal was to reconstruct indoor structures based on the room unit. On the other hand, the wall segmentation task, which includes a pixel-wise evaluation of geometric walls based on the Jaccard index (Everingham et al. 2010), was inappropriate in our case because our raster results, the style-transferred plan, targeted a unified style that represented walls in a simplified expression, not with a precise geometry. Moreover, fundamentally for floor plan analysis, good performance in segmentation tasks did not guarantee good vector results. Thus, provided that we assessed the final result of the vectorized floor plan, it was trivial to evaluate the raster results, which were intermediate products used for vectorization.

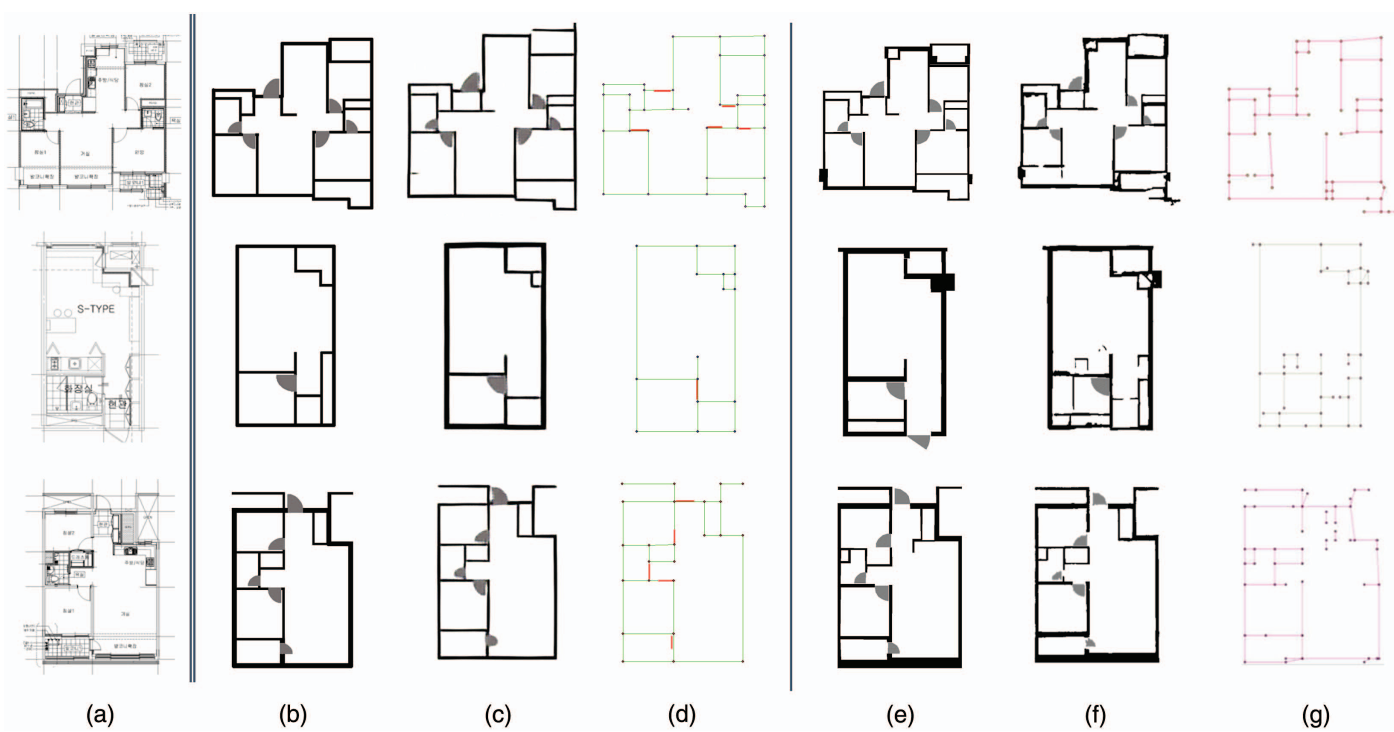


Fig. 11. (Color) Visual comparison of outputs produced by our method (b–d) and by Jang et al. (2020) (e–g); annotation, raster result, and vector result, repectively. [Images in (f) and (g) adapted from Jang et al. 2020.]

Fig. 12. (Color) Results of our method for various types of complex floor plans in EAIS. (a–c) An input floor plan, our vectorized floor plan, and the corresponding 3D model. In the vectorized floor plan, the walls and openings are represented by lines, respectively. The legend for the junction colors is shown in Fig. 9.

The evaluation protocol for the room-detection task resulted in a detection rate (DR) and recognition accuracy (RA) based on the match-score table (Phillips and Chhabra 1999), which was characterized by reflecting the exact matches (one to one) as well as the partial matches (one to many and many to one). Calculation of the DR and RA with respect to the room-detection task were detailed by de las Heras et al. (2014). The predicted rooms for each raster and vector results were regarded as connected components and closed polygons, respectively, and the corresponding ground truth was prepared. When calculating the match-scores of the predicted room and ground truth, the acceptance and rejection thresholds that were used to determine whether each pair matched were 0.5 and 0.1, respectively, which were the same as those used previously for CVC data (Macé et al. 2010; Ahmed et al. 2011; de las Heras et al. 2014).

Table 4 shows an evaluation of the room-detection task for our proposed method using EAIS data. The evaluation of the style-transferred plan (ours) and vectorized floor plans (ours*) is nearly identical, showing that the room structure retrieved in our raster results is well-conveyed to the vector results. See Figs. 11(c and d) to compare both results.

There is no relevant research that could be used for a direct comparison, as handling a complicated floor plan has rarely been covered in floor plan analysis. Also, there is no shared annotation for complicated floor plan datasets, which are fundamental barriers for learning-based approaches to compare each other. Even so, to verify our performance, we refer to the room-detection task on CVC and use it as baseline performance, which has been updated over the years in the conventional field of floor plan analysis. Compared with de las Heras et al. (2014) on CVC data, our method shows

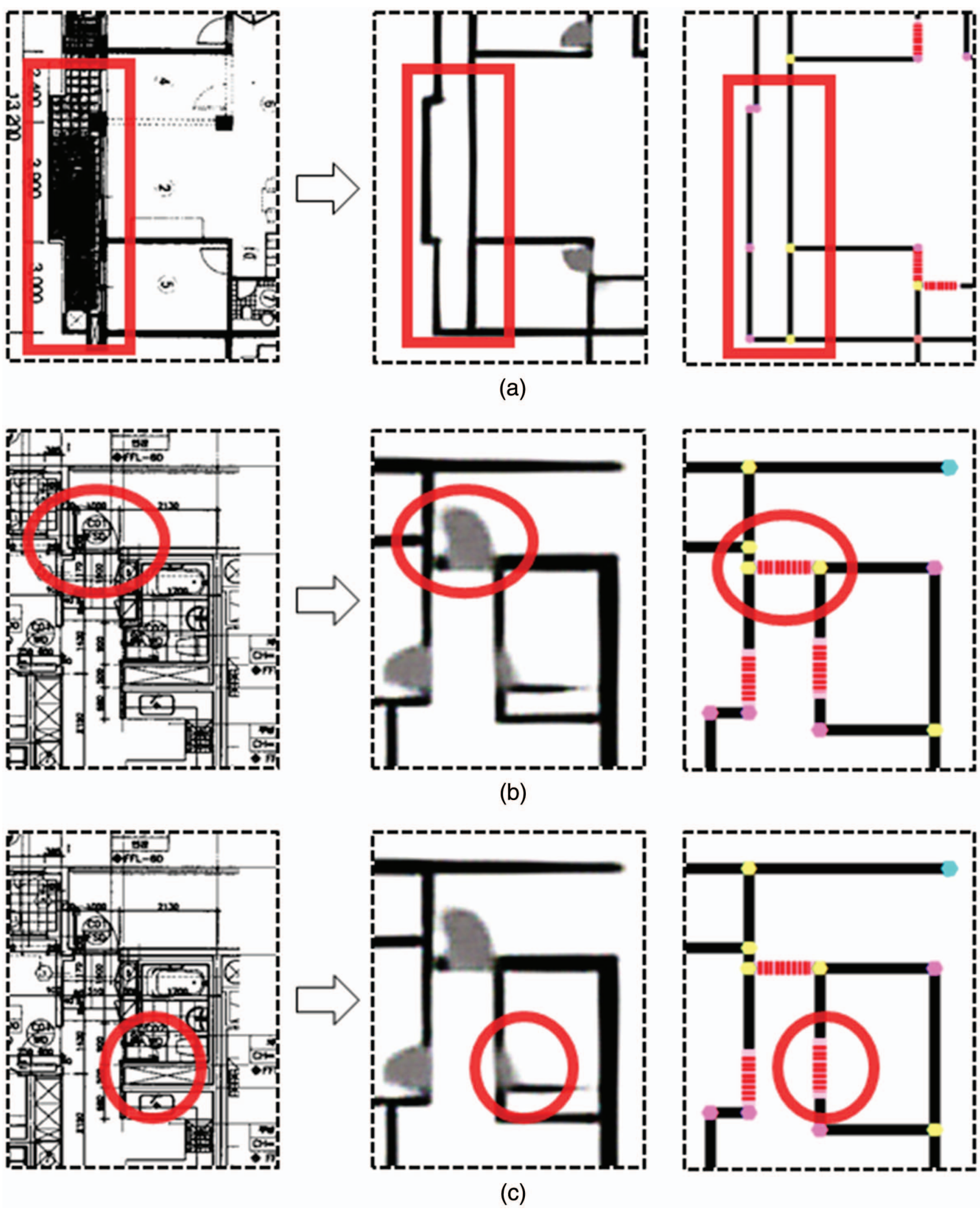


Fig. 13. (Color) Examples of our results using challenging graphics: (a) walls with irregular patterns; (b) opening with overlapping notations; and (c) opening not detected as neat arc shapes.

slightly lower performance in terms of DR and RA (5%–6%), despite targeting complicated floor plans. Besides, ours significantly improves the number of exact matches (one-to-one rate), demonstrating its superiority in terms of constructing an indoor structure without particular postprocessing steps, such as merging and splitting predicted rooms.

Qualitative Comparison

We compared our method with that in (Jang et al. 2020), which targeted EAIS data. Although Jang et al. (2020) performed vectorization of the floor plan, they evaluated only the segmentation task quantitatively while showing only a few examples of the vector results. Thus, we included their examples in our test images and compared the overall outputs qualitatively (Fig. 11).

First, comparing the raster results [Figs. 11(c and f)] shows that our style transfer is superior to the approach based on segmentation in terms of extracting walls in the form of neat and straight lines with clear boundaries. When the format of the floor plan is complicated and there are symbols similar to those that indicate walls, the approach based on segmentation naturally offers reduced performance, whereas our method still robustly generates the room layouts, preserving the basic expression of the unified style. This robustness is due to our deep learning networks being trained while paying attention to style-loss, and it is much more noticeable when applied to various complex floor plans (Fig. 12).

Figs. 11(d and g) show the vectorized floor plans from each study. Jang et al. (2020) dealt with vectorization separately from segmentation, so we can see that performance decreases when converting the raster to a vector. However, in our method, the raster outputs from the deep networks already reflect the simplification steps, so we can convert the shape of the raster to the vector almost as is without losing the room structure. Also, while the vector results in Fig. 11(g) have several slight problems, such as uneven walls and meaningless closed spaces, our method neatly vectorizes the floor plan by means of the constraints in the optimization.

Discussion

More results of applying our method to various types of complex drawings are available in Fig. 12. To intuitively visualize our room structure results, we generated 3D representations of the building models using our vectorized floor plans. Under each example, we display statistics about the correctness of each junction, opening, and room estimation. The notation shows the total count of an exact match/ground truth/prediction.

To highlight the advantages of using our method, we cropped and zoomed in on the results, especially for graphics that were considered challenging for floor plan recognition (Fig. 13). Fig. 13(a) indicates walls with messy and irregular patterns and Figs. 13(b and c) indicate openings with overlapping notation. Our method based on style transfer is characterized by the production of an output that preserves the overall shape of the layout, thus showing good performance even with complicated graphics. Moreover, even when the deep networks failed to detect an opening as an intact arc-shape due to ambiguous graphics, the corresponding opening junctions could still be detected, so that this missing information could be supplemented in the vectorization step [Fig. 13(c)].

By processing floor plans from other data sources, we verified that our model could be generalized and applied to the new formats that were not used for training. Fig. 14 shows our results of the style-transferred plans for the four formats of CVC. Except for very distinctive styles different from the format samples of EAIS, our model retrieved the overall structure of the interior, although it did not detect all the detailed walls. Unfortunately, since BlackSet, which accounts for over 75% of CVC, cannot be covered by our trained model through EAIS, direct quantitative comparisons have been difficult to perform fairly (Our model detects the outer wall of BlackSet well, which is expressed similarly to other drawings, while it is difficult to distinguish the inner wall that uses the unique notation of the BlackSet). Still, such generalized application suggests that floorplan analysis can be performed through learning drawings of similar styles, not exactly the same format, especially when the restoration of the entire indoor structure is prioritized.

Finally, we discuss some limitations that make it difficult for our results to be used directly as indoor information. In some situations, topological information is not fully reconstructed when generating building models. As it is dependent on learning, our method implicitly performs a simplification of the layouts, which means that

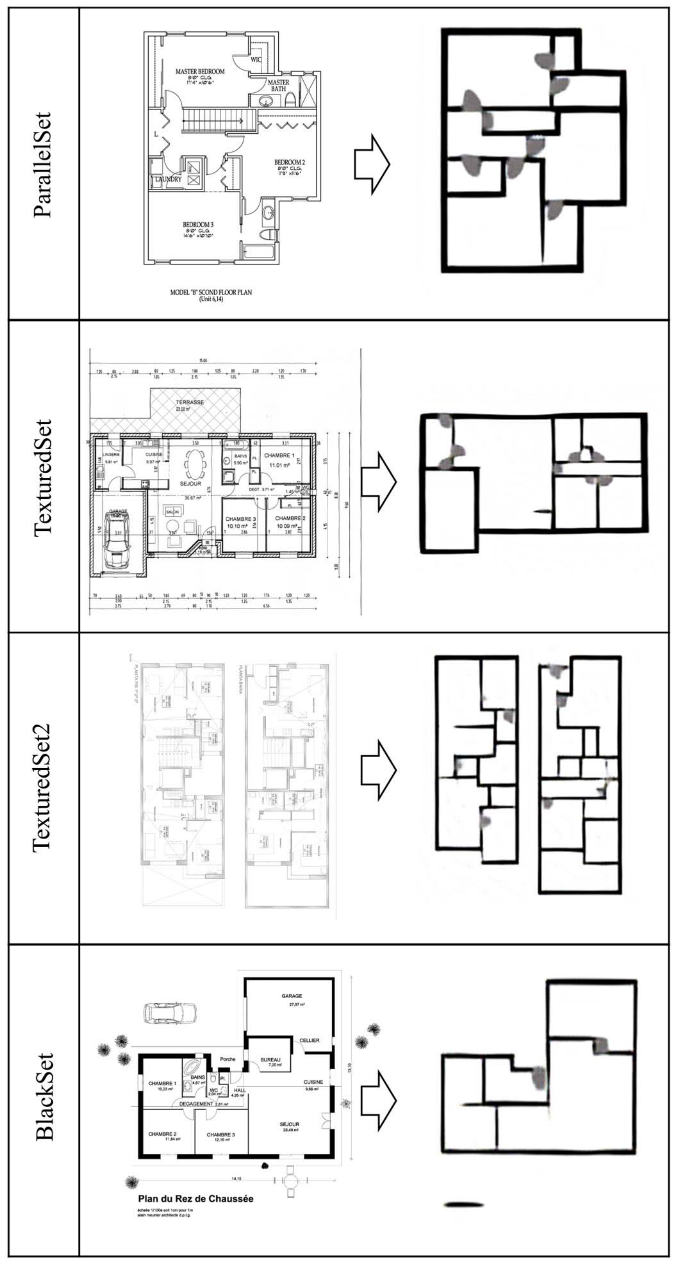


Fig. 14. Results on the four formats of CVC. For the cases of BlackSet with distinctive wall notation and TexturedSet2 containing two buildings at once, the performance is rather low.

omission or closing of the rooms can occur. For example, rooms without openings are generated at times, yielding an incomplete topology for building models. To address this issue, we believe there is room for improvement in the method itself. After coarse recognition of the room structure, connectivity between rooms can be obtained by taking advantage of locally available information and of contextual knowledge in order to reconstruct intact building models.

In addition, the method is presently limited to addressing layouts within the Manhattan assumption, although it could be extended to those that include diagonal walls. We present results only including walls with axial directionality since there is not enough variation in type and quantity in the EAIS dataset to train deep networks on diagonal walls. However, the vectorization based on our junction types that are not relevant to wall orientations could be directly adapted to diagonal layouts.

Conclusion

This paper proposes a novel method of recognizing floor plan elements to address various types of complex formats. Our method results in a vectorized floor plan that can be used to generate indoor models, including CityGML and IndoorGML. There are three key contributions of this work. First, we significantly extend the coverage of floor plan analysis into complex cases with a new approach based on style transfer. Second, we enable the vectorization of floor plans via deep learning models by designing a multitask network that performs style transfer and junction extraction. Third, we introduced a new annotation, namely the unified style, that robustly represents room structure, even from diverse and complicated floor plans. Also, the unified style makes labeling tasks much easier, thus allowing for deep networks to be applied to floor plans practically. In the end, we evaluate our method using the EAIS dataset. The results show the superiority of our method in terms of room-detection tasks, especially when targeting irregular patterns or overlapping graphics. In the future, we plan to add connectivity between rooms in our vector results so that more complete indoor information can be generated in terms of topology.

Data Availability Statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request (list items)

- Our trained models and results.
- EAIS floor plan dataset with our annotations.

Acknowledgments

This research was supported by a grant (19NSIP-B135746-03) from the National Spatial Information Research Program funded by the Korea Ministry of Land, Infrastructure and Transport.

References

Ahmed, S., M. Liwicki, M. Weber, and A. Dengel. 2011. "Improved automatic analysis of architectural floor plans." In *Proc., 2011 Int. Conf. on Document Analysis and Recognition*. New York: IEEE.

Bulat, A., and G. Tzimiropoulos. 2016. "Human pose estimation via convolutional part heat map regression." In *Proc., European Conf. on Computer Vision*. Berlin: Springer.

Chen, X., S. B. Kang, Y. Xu, J. Dorsey, and H. Y. Shum. 2008. "Sketching reality: Realistic interpretation of architectural designs." *ACM Trans. Graphics* 27 (2): 1–15. https://doi.org/10.1145/1356682.1356684.

de las Heras, L. P., S. Ahmed, M. Liwicki, E. Valveny, and G. Sánchez. 2014. "Statistical segmentation and structural recognition for floor plan interpretation." *Int. J. Doc. Anal. Recogn.* 17 (3): 221–237. https://doi.org/10.1007/s10032-013-0215-2.

de las Heras, L. P., O. R. Terrades, S. Robles, and G. Sánchez. 2015. "CVC-FP and SGT: A new database for structural floor plan analysis and its groundtruthing tool." *Int. J. Doc. Anal. Recogn.* 18 (1): 15–30. https://doi.org/10.1007/s10032-014-0236-5.

Dimitrov, A., and M. Golparvar-Fard. 2015. "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems." *Autom. Constr.* 51 (Mar): 32–45. https://doi.org/10.1016/j.autcon.2014.12.015.

Dodge, S., J. Xu, and B. Stenger. 2017. "Parsing floor plan images." In *Proc., 2017 15th IAPR Int. Conf. on Machine Vision Applications*, New York: IEEE.

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. "The Pascal visual object classes (VOC) challenge." *Int. J. Comput. Vision* 88 (2): 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Fernández-Pacheco, D. G., F. Albert, N. Aleixos, and J. Conesa. 2012. "A new paradigm based on agents applied to free-hand sketch recognition." *Expert Syst. Appl.* 39 (8): 7181–7195. https://doi.org/10.1016/j.eswa.2012.01.063.

Froehlich, M., S. Azhar, and M. Vanture. 2017. "An investigation of Google tango® tablet for low cost 3D scanning." In *Proc., Int. Symp. on Automation and Robotics in Construction*, Taipei, Taiwan: The International Association for Automation and Robotics in Construction.

Gatys, L. A., A. S. Ecker, and M. Bethge. 2015. "A neural algorithm of artistic style." Preprint, submitted August 26, 2015. http://arxiv.org/abs/1508.06576.

Gimenez, L., J. L. Hippolyte, S. Robert, F. Suard, and K. Zreik. 2015. "Review: Reconstruction of 3D building information models from 2D scanned plans." *J. Build. Eng.* 2 (Jun): 24–35. https://doi.org/10.1016/j.jobe.2015.04.002.

Gimenez, L., S. Robert, F. Suard, and K. Zreik. 2016. "Automatic reconstruction of 3D building models from scanned 2D floor plans." *Autom. Constr.* 63 (Mar): 48–56. https://doi.org/10.1016/j.autcon.2015.12.008.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. "Generative adversarial nets." In *Proc., Advances in Neural Information Processing Systems 27 (NIPS 2014)*. Red Hook, NY: Curran Associates.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep residual learning for image recognition." In *Proc., 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 770–778. New York: IEEE.

Isola, P., J. Y. Zhu, T. Zhou, and A. A. Efros. 2017. "Image-to-image translation with conditional adversarial networks." In *Proc., 2017 IEEE Conf. on Computer Vision and Pattern Recognition*. New York: IEEE.

Jang, H., K. Yu, and J. H. Yang. 2020. "Indoor reconstruction from floor plan images with a deep learning approach." *ISPRS Int. J. Geo-Inf.* 9 (2): 65. https://doi.org/10.3390/ijgi9020065.

Jing, Y., Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. 2019. "Neural style transfer: A review." In *Proc., IEEE Transactions on Visualization and Computer Graphics*. New York: IEEE.

Kang, H. K., and K. J. Li. 2017. "A standard indoor spatial data model—OGC indoor GML and implementation approaches." *ISPRS Int. J. Geo-Inf.* 6 (4): 116. https://doi.org/10.3390/ijgi6040116.

Kim, T., M. Cha, H. Kim, J. K. Lee, and J. Kim. 2017. "Learning to discover cross-domain relations with generative adversarial networks." In *Proc., 34th Int. Conf. on Machine Learning*, 1857–1865. Brookline, MA: Microtome Publishing.

Konde, A., H. Tauscher, F. Biljecki, and J. Crawford. 2018. "Floor plans in city GML." In *Proc., ISPRS Annals of Photo Grammetry, Remote Sensing and Spatial Information Sciences*, 25–32. Göttingen, Germany: Copernicus Publications.

Liu, C., A. G. Schwing, K. Kundu, R. Urtasun, and S. Fidler. 2015. "Rent3D: Floor-plan priors for monocular layout estimation." In *Proc.,* *2015 IEEE Conf. on Computer Vision and Pattern Recognition*, 3413–3421. New York: IEEE.

Liu, C., J. Wu, P. Kohli, and Y. Furukawa. 2017a. "Raster-to-vector: Revisiting floor plan transformation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*. New York: IEEE.

Liu, X., X. Wang, G. Wright, J. C. P. Cheng, X. Li, and R. Liu. 2017b. "A state-of-the-art review on the integration of building information modeling (BIM) and geographic information system (GIS)." *ISPRS Int. J. Geo-Inf.* 6 (2): 53. https://doi.org/10.3390/ijgi6020053.

Long, J., E. Shelhamer, and T. Darrell. 2015. "Fully convolutional networks for semantic segmentation." In *Proc., IEEE Conf. on Computer Vision and Pattern Recognition*, 3431–3440. New York: IEEE.

Macé, S., H. Locteau, E. Valveny, and S. A. Tabbone. 2010. "A system to detect rooms in architectural floor plan images." In *Proc., 9th IAPR Int. Workshop on Document Analysis Systems*. New York: Association for Computing Machinery.

May, M., and G. Williams. 2017. *The facility manager's guide to information technology.* Houston: International Facility Management Association.

Phillips, I. T., and A. K. Chhabra. 1999. "Empirical performance evaluation of graphics recognition systems." *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (9): 849–870. https://doi.org/10.1109/34.790427.

Rendek, J., G. Masini, P. Dosch, and K. Tombre. 2004. "The search for genericity in graphics recognition applications: Design issues of the QGAR software system." In *Document analysis systems VI: DAS 2004*. Berlin: Springer.

Rusli, M. E., M. Ali, N. Jamil, and M. M. Din. 2016. "An improved indoor positioning algorithm based on RSSI-Trilateration technique for internet of things (IOT)." In *Proc., 2016 Int. Conf. on Computer and Communication Engineering*. New York: IEEE.

Sankar, A., and S. Seitz. 2012. "Capturing indoor scenes with smart phones." In *Proc., 25th Annual ACM Symp. on User Interface Software and Technology.* New York: Association for Computing Machinery.

Tang, P., D. Huber, B. Akinci, R. Lipman, and A. Lytle. 2010. "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques." *Autom. Constr.* 19 (7): 829–843. https://doi.org/10.1016/j.autcon.2010.06.007.

Tombre, K., S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch. 2002. "Text/graphics separation revisited." In *Lecture notes in computer science*. Berlin: Springer.

Volk, R., J. Stengel, and F. Schultmann. 2014. "Building information modeling (BIM) for existing buildings—Literature review and future needs." *Autom. Constr.* 38 (Mar): 109–127. https://doi.org/10.1016/j.autcon.2013.10.023.

Wei, Y., and B. Akinci. 2019. "A vision and learning-based indoor localization and semantic mapping framework for facility operations and management." *Autom. Constr.* 107 (Nov): 102915. https://doi.org/10.1016/j.autcon.2019.102915.

Xu, W., L. Liu, S. Zlatanova, W. Penard, and Q. Xionge. 2018. "A pedestrian tracking algorithm using grid-based indoor model." *Autom. Constr.* 92 (Aug): 173–187. https://doi.org/10.1016/j.autcon.2018.03.031.

Yalcinkaya, M., and V. Singh. 2014. "Building information modeling (BIM) for facilities management—Literature review and future needs." In *Proc., IFIP Int. Conf. on Product Lifecycle Management*. Berlin: Springer.

Zeng, Z., X. Li, Y. Kin, and C. W. Fu. 2019. "Deep floor plan recognition using a multi-task network with room-boundary-guided attention." In *Proc., IEEE Int. Conf. on Computer Vision*, 9095–9103. New York: IEEE.

Zhu, J. Y., T. Park, P. Isola, A. A. Efros. 2017. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In *Proc., 2017 IEEE Int. Conf. on Computer Vision*, 2223–2232. New York: IEEE.