



Article Indoor Reconstruction from Floorplan Images with a Deep Learning Approach

Hanme Jang^(D), Kiyun Yu and JongHyeon Yang *^(D)

Department of Civil and Environmental Engineering, Seoul National University, Seoul 08826, Korea; janghanie1@snu.ac.kr (H.J.); kiyun@snu.ac.kr (K.Y.)

* Correspondence: yangjonghyeon@snu.ac.kr; Tel.: +82-10-5068-2628

Received: 19 November 2019; Accepted: 19 January 2020; Published: 21 January 2020



Abstract: Although interest in indoor space modeling is increasing, the quantity of indoor spatial data available is currently very scarce compared to its demand. Many studies have been carried out to acquire indoor spatial information from floorplan images because they are relatively cheap and easy to access. However, existing studies do not take international standards and usability into consideration, they consider only 2D geometry. This study aims to generate basic data that can be converted to indoor spatial information using IndoorGML (Indoor Geography Markup Language) thick wall model or the CityGML (City Geography Markup Language) level of detail 2 by creating vector-formed data while preserving wall thickness. To achieve this, recent Convolutional Neural Networks are used on floorplan images to detect wall and door pixels. Additionally, centerline and corner detection algorithms were applied to convert wall and door images into vector data. In this manner, we obtained high-quality raster segmentation results and reliable vector data with node-edge structure and thickness attributes that enabled the structures of vertical and horizontal wall segments and diagonal walls to be determined with precision. Some of the vector results were converted into CityGML and IndoorGML form and visualized, demonstrating the validity of our work.

Keywords: floorplan image; semantic segmentation; centerline; corner detection; indoor spatial data

1. Introduction

With developments in computer and mobile technologies, interest in indoor space modeling is increasing. According to the U.S. Environmental Protection Agency, 75% of people living in cities around the world spend most of their time indoors. Moreover, the demand for indoor navigation and positioning technology is increasing. Therefore, indoor spatial information has a high utility value, and it is important to construct it. However, according to [1,2], the amount of indoor spatial data available, such as indoor navigation and indoor modelling, is very small relative to the demand. To address this, research has been conducted on generating indoor spatial information sing various data such as LiDAR (Light Detection and Ranging), BIM (Building Information Modeling), and 2D floorplans. LiDAR is a surveying method that obtains the distance from the target by aiming a laser towards it and measuring the time gap and difference in wavelength of the reflected light. Hundreds of thousands of lasers are shot at the same time and return information from certain points on the target. Point data acquired by LiDAR systems describe 3D worlds as they are; therefore, it has been widely used to create mesh polygons that can be converted into elements of spatial data [3,4]. However, because of the high cost of the LiDAR system and the time the process takes, only a few buildings were modelled this way. However, BIM—especially its IFC(Industry Foundation Classes)—is a widely used open file format that describes 3D buildings edited by various software like Revit and ArchiCAD [5]. It contains enough attributes and geometric properties of all building elements that can be converted to spatial data. Sometimes, IFC elements are projected into the 2D plane and printed as a floorplan for sharing

on sites (to obtain opinions) or making a contract that is less formal than the original BIM vector data. Floorplans have been steadily gaining attention as spatial data sources because they are relatively affordable and accessible compared to other data [6]. However, when printed on paper, the geometric point and line information vanish; only raster information remains. Several papers on the subject and methods for recovering vector data from raster 2D floorplans have been suggested to overcome this.

The OGC (Open Geospatial Consortium), the global committee for geospatial contents and services standard recently accepted two different GML models, CityGML and IndoorGML as the standard 3D data models for spatial modelling inside buildings. It is pertinent to follow a standardized data format when constructing indoor spatial data for the purpose of generating, utilizing, and converting them to other GIS formats.

Studies on extracting and modifying objects such as doors and walls from floorplan images can be classified into two categories, according to whether they extract the objects in the image using a predefined rule set or whether they extract objects using machine learning algorithms that are robust to various data. Macé et al. [7] separated images and texts with geometric elements from a floorplan using the Qgar tool [8] and Hough transform method. As their method uses well-designed rules for certain types of floorplans, low accuracy is expected for other kinds. In addition, they could be difficult to use as spatial information due to their lack of coordinates. Gimenez et al. [9] defined the category of wall segments well and classified them at the pixel level using a predefined rule set. However, as these methods have many predefined hyper-parameters, they need to be chosen before the process, and thus they lack generalizability. Ahmed et al. [10,11] separated text and graphics and categorized walls according to the thickness of their line (i.e., thick, medium, or thin). This method outperformed others in terms of finding wall pixels; however, doors and other elements were not considered and the result was in a raster form unsuitable for spatial data.

However, as these studies processed floorplans according to predefined rules and parameters, they lacked generality in terms of noise or multiple datasets. To overcome these problems, methods using machine learning algorithms appeared. De las Heras et al. [12] constructed a dataset denoted CVC-FP (Centre de Visio per Computador Floorplan), which contains multiple types of 2D floorplans, then used it to extract walls using SVM-BOVW (Support Vector Machine Bag of Visual Words) classical machine learning algorithm and extracted room structures using the centerline detection method. Although the accuracy of generated data was quite low, this research was the first to try to generate vector data from various types of 2D floorplan images. Dodge et al. [13] used FCN-2s to segment walls in addition to R-FP (Rakuten Floorplan) datasets for training to improve segmentation results. However, they did not conduct any postprocesses to construct indoor spatial information. Liu et al. [14] applied ResNet-152(Residual Network 152) to R-FP to extract the corners of walls and applied integer programming on the extracted corners to construct vector data that represented walls. However, their research was conducted on the assumption that every wall in the floorplan was either vertical or horizontal, not diagonal. Furthermore, they did not consider the elements of indoor spatial information, such as wall thickness.

In general, existing studies on constructing indoor spatial information from floorplans either use the rule-based method, which only works well for specific drawings that meet predetermined criteria; or machine learning algorithms, which have higher performance than the rule-based method because they learn patterns from data automatically and show robustness in performance against noise. Representative studies are summarized in Table 1. However, these studies discussed above also have some limitations in terms of indoor spatial information. First, information regarding walls and doors was simply extracted in the form of an image. In order to comply with the OGC standard, the objects must be represented as vectors of points, lines, and polygons. Second, studies that performed vector transformations simply expressed the overall shape of the building. According to Kang et al. [15], IndoorGML adopts the thick wall model and uses the thickness of doors and walls to generate spatial information. In addition, Boeters et al. [16] and Goetz et al. [17] suggest that geometric information should be preserved as much as possible when generating CityGML data, typically to create interior and exterior walls using wall thickness, and to maintain the relative position and distance between objects.

In this study, we aimed to generate vector data that could be converted into CityGML LoD2 (Level of Detail 2) or IndoorGML thick wall models given floorplan images. To this end, we used CNNs (Convolutional Neural Networks) to generate wall and door pixel images and used a vector-to-raster conversion method to convert wall and door images into vector data. Compared to existing work, the novelty of our study is as follows. First, we generated reliable vector data that could be converted into CityGML LOD2 or IndoorGML thick wall models, unlike in previous studies. Second, we adopted modern CNNs to generate high-quality raster data that could then be used to generate vector data. Third, to fully recover wall geometry, we extracted diagonal wall segments, as well as extract vertical or horizontal wall segments.

Title	Dataset	Method	Result
[7]	CVC-FP	Qgar project, Hough transform	Entity segmentation
[9]	CVC-FP	Predefined rule	Room detection
[10]	CVC-FP	Contour extraction	Room detection
[11]	Defined in paper	Predefined rule	Room detection
[12]	CVC-FP	SVM-BoVM, centerline detection	Room detection
[13]	CVC-FP, R-FP	Deep learning (FCN-2s)	Pixel
[14]	R-FP	Deep learning (ResNet-152)	Vector with points and lines

Table 1. Summary of the datasets, methods, and results of selected previous studies.

Note: FCN-2s: Fully Convolutional Network with stride 2; ResNet-152: Residual Network 152.

2. Materials and Methods

To generate vector data from floorplans, we divided our workflow into two processes: segmentation and vector data generation (Figure 1). The segmentation process is divided into two stages, namely training and inference stages, in which we extract wall and door images from raw 2D floorplan images. To train the CNNs, we classified datasets into three categories. These categories contained 247 images for training, 25 images for validation, and 47 for testing. Data augmentation techniques such as flip, shift, and brightness were used at the training stage, after which the CNNs were trained by being fed augmented images and constructed ground-truths in training sets. In the inference stage, we segmented walls and doors in test sets using trained models. In the process of generating vector data, we converted segmented wall and door images into vector data. The simple wall geometry was constructed using centerline detection and refined using the corner detection method on the centerline images. The thickness of each wall was calculated automatically by overlapping the image of the wall and the refined wall geometry. Finally, the wall graph and door point, which were simply extracted separately using the corner detection method, were merged. All the experiments were conducted using TensorFlow and PyQGIS library in python environment.



Figure 1. Workflow of the study, showing the input data, process, and results. Note: CNN = convolutional neural network.

2.1. Deep-Learning-Based Segmentation

While existing pattern recognition algorithms are difficult to train in an end-to-end manner (since the feature extractors that extract hand-crafted features and classifiers are separated), deep networks integrate low-, mid-, and high-level features and classifiers in an end-to-end multilayer fashion [18]. Since Krizhevsky et al. [19] demonstrated astonishing results using the ImageNet 2012 classification benchmark, CNNs have shown excellent performance in many computer vision tasks, such as image classification, semantic segmentation, object detection, and instance segmentation [20–23]. To train a CNN for segmentation, a sufficient quantity of high-quality data is required. When data are difficult to obtain, weak annotations such as image-level tags, bounding boxes, or scribbles can be used to achieve results like the results of pixel-wise annotations [24]. In recent years, this approach has reached 90% of the performance of Deeplab v1 [25], but it still falls short of the accuracy of supervised learning with pixel-wise annotations.

In this study, we used a GCN (Global Convolutional Network), Deeplab v3 plus which is an advanced version of Deeplab v1 and a PSPNet (Pyramid Scene Parsing Network) as they were found to demonstrate good performance in semantic segmentation for Pascal VOC2012, Microsoft COCO (Common Object in Context) and others. Furthermore, considering the characteristics of the floorplan with a small number of classes, we used shallower models that showed efficient performance despite the small number of parameters. We used DarkNet53 (Figure 2) as a backbone, removing the final average pooling layer, fully connected layer, and SoftMax layer; and then joined five deconvolution layers to restore the original size. After the first deconvolution layer, we added the skip connection before each deconvolution layer. In order to improve the accuracy of extracting objects, we trained the networks for walls and doors individually and constructed pixel-level annotations using LabelMe for our training data, which were trained in a fully supervised manner. We carried out data augmentation for shift, rotation, flip, and brightness change, and used the R-FP as training data, similar to Dodge et al. [12]. Simultaneously, data characteristics can affect the segmentation results. If a foreground pixel occupies only a small region compared to the background, it often causes the learning process to get trapped in the local minima of the loss function, yielding a network whose predictions are strongly biased towards the background [22]. To prevent this, we used weighted cross entropy loss and dice loss in the training process. We used batch normalization, ReLUs (Rectified Linear Units) method and trained these with ADAM (Adaptive Moment estimation) optimizers. Finally, we used ensemble techniques to improve the segmentation results.

	Туре	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
_	Convolutional	64	3 x 3 / 2	128 x 128
	Convolutional	32	1 x 1	
1 x	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
[Convolutional	64	1 x 1	
2 x	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 / 2	32 x 32
[Convolutional	128	1 x 1	
8 x	Convolutional	256	3 x 3	
	Residual			32 x 32
-	Convolutional	512	3 x 3 / 2	16 x 16
	Convolutional	256	1 x 1	
8 x	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 / 2	8 x 8
[Convolutional	512	1 x 1	
4 x	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 2. Architecture of DarkNet53.

2.2. Centerline-Based Approach

In order to convert the images of the walls into vector data, we first used the centerline detection process. Since most of the walls in the floorplan were straight lines, we represented them using polylines. We then applied corner detection algorithms to the output image. We constructed a node-edge graph using the output image, and then combined the two outputs into one to generate the wall and door outputs in vector form. We then applied the corner detection method and postprocessed the door image to generate the final door segment.

2.2.1. Centerline Detection

The final outcome required the exact coordinates of every corner of the wall segments and the edges between them. To accomplish this, the gap between the position of the real corners and the position of the candidates extracted from the image should be minimized, and isolated corners must be connected properly. The centerline (i.e., the collection of pixels in the middle of each wall) can be used as a guideline for two reasons. First, when using the Harris corner detection method (Section 2.2.2), the locations of corners can be found by identifying the feature points in the image. However, if we use the wall image from segmentation as the input, we find that the thickness of the wall causes inaccuracies in the locations of the corners. The location of the detected corners has errors in the x or y direction by half of the wall thickness compared to the exact location of the end points of each wall. In addition, due to the thickness of the wall, multiple corner points are detected on the inside and outside of the wall at the edges.

Second, the centerline is simply a raster form of the grid, and the pixels do not fall apart (Figure 3). By applying classical image processing, we can convert the image into a node-edge graph. As described in Figure 3, each pixel can work as a node and be connected with every adjacent pixel to create edges. Consequently, the overall appearance of the building wall can be viewed, which makes connecting corners possible.



Figure 3. Centerline detection and node-edge graph.

According to Gramblicka and Vasky [26], it is very important to select the correct algorithm when performing centerline detection. We found that the walls included in the drawing were composed of rectangles with long horizontals and short verticals with many vertical intersections. The algorithm proposed by Huang et al. [27], which improved the classical centerline detection algorithm presented by Boudaoud et al. [28], had a good fit with our data and showed great performance when handling noise. It was, therefore, selected for use in this study.

2.2.2. Corner Detection

As shown in Figure 3, a node-edge graph can be a wall segment on its own. However, to express the wall with the minimum number of corners and straight lines, useless nodes should be removed. Only the nodes representing "real corners" (Figure 4) rather the edge of the wall should be included. To identify the node corresponding to the corner of the wall, we conducted the corner detection process. A corner can be defined as a "visually angular" point in the image. The methods used for corner detection can be classified into signal intensity, contour, and template methods, according to the searching algorithm. Among them, the signal intensity method is suitable for processing linear-shaped features such as walls. Zhang et al. [29] and Harris and Stephens [30] also reported that the classical algorithms based on signal intensity showed strong performance in the point searching field. Therefore, in this study, we used Harris corner detection, which is a representative signal intensity method, as our corner detection algorithm.



Figure 4. Corners of the wall: (a) corner type a; (b) corner type b.

2.2.3. Wall Segments and Thickness

To generate wall thickness attributes, the method of Jae et al. [31] was used. We used vector data as the input, which were simplified by combining the output of the Harris corner detection process and the node-edge graph from the centerline image. The vector data and the original raster image of the walls were overlapped, and the kernels moved using 1-pixel-width slides along the wall to determine wall thickness (Figure 5). Wall thickness data are stored separately, such that they can be used to create spatial information, such as CityGML and IndoorGML.



Figure 5. Restoring wall thickness using the moving kernel method.

2.2.4. Door detection

Similar to the wall image, the door image contains door objects in the form of raster pixels. It is used to create a vector object to represent the doors in a similar manner to wall image processing. Since the floorplans used in this study are very complex and diverse in nature, every door is assumed to be in the shape shown in Figure 6a to increase overall accuracy. Following Or et al. [32], assumptions regarding the shape of the doors are listed below:

- Every door consists of two orthogonal straight lines and one additional arc;
- The length of two lines are almost identical;
- The arc is always a quarter of a circle;
- In some cases, one of the two straight lines is omitted.



Figure 6. Process of door extraction: (a) raw image; (b) rasterized door, (c) extracted points; (d) final output.

We applied a process similar to that described in Section 2.2.2 to extract the corners and determine the direction and shape of each door. Only three isolated points were produced from door segmentation images (Figure 6b), without any other geometry, as shown in Figure 6c. To overcome this, we used the data to construct a virtual triangle, calculated the distances between the edges and corners of every wall, and decided on the direction. Finally, the two nearest points from the corners on the edge of the wall segment were used to make a line as a door segment, as shown in Figure 6d.

3. Results

3.1. Data Generation

From EAIS (Korean government's web system for computerization of architectural tasks like building license, construction and demolition), we downloaded floorplan images and manually annotated them for pixel-wise ground truth generation. To precisely reproduce the geometry of the architectural plan, we labeled various structural features, such as air duct shafts (AD) and access panels (AP) (Figure 7).



Figure 7. Examples of pixel-wise ground truth generation: (a) source image; (b) ground truth.

3.2. Results of Experiments

We used a total of 319 floorplan images divided into 246 training sets, 25 validation sets, and 47 test sets through random sampling for CNN training, validation, and hyperparameter tuning of Harris corner detection. The results of the experiments are described as follows. First, we demonstrate the results of extraction of the segments of walls and doors while preserving the actual wall thickness using CNN. Second, we demonstrate the results of converting the raster images into node-edge structure vector data.

3.2.1. Wall and Door Segmentation Results

The results are reported according to the model used, the loss function used, and whether the R-FP dataset was added during the training process. We did not add R-FP datasets to the door segmentation results because there are no pixel-wise annotations for the doors. All the following results are reported in test sets. We used intersection over union (IoU) to evaluate the results.

The results of wall segmentation without R-FP datasets are shown in Table 2, whilst Table 3 shows the results with R-FP datasets. When the R-FP data were used during the training process, the networks yielded better results. In addition, when we used dice loss, we found a higher IoU than weighted cross entropy loss. Among the various models used, modified DarkNet53 produced the best results.

Table 2. Results of wall segmentation without R-FP datasets. Note: WCE = weighted cross entropy; GCN = global convolutional network. PSPNet = pyramid scene parsing network.

Network	Dice Loss	WCE Loss
DarkNet53	0.679	0.6578
GCN	0.633	0.6299
Deeplab v3 plus	0.612	0.6066
PSPNet	0.566	0.5637

Table 3.	Results	of wall	segmentation	with	R-FP	datasets.
incie o.	reound	or main	ocginentation	** 1611	T C T T	aatabetb.

Network	Dice Loss	WCE Loss	
DarkNet53	0.688	0.6699	
GCN	0.657	0.6352	
Deeplab v3 plus	0.643	0.6125	
PSPNet	0.572	0.5802	

Table 4 shows the results of door segmentation. We found that similar to wall separation, using modified DarkNet53 and dice loss yielded better performance.

Network	Dice Loss	WCE Loss
DarkNet53	0.7805	0.6233
GCN	0.6532	0.5849
Deeplab v3 plus	0.5605	0.4308
PSPNet	0.7405	0.6116

Table 4. Results of door segmentation with R-FP datasets.

The door segmentation results showed better performance than wall segmentation, because in most cases the shape of the doors is uniform. However, the thickness, length, and types of walls vary in floorplan datasets. To improve wall segmentation performance, we ensembled three models that were trained with different hyperparameters, yielding an IoU of 0.7283. Table 5 shows the results of segmentation. As shown, the walls and doors are well segmented, and the results could be used as input data for vector generation.



Table 5. Wall and door segmentation results.

3.2.2. Vector Generation Results

The hyperparameters of Harris corner detection were set to blocksize = 2, ksize = 7, k = 0.05 using the validation set. Improved centerline detection method generated relatively clear and simple data as shown in Table 6. Especially, unintended branches and structure with hole due to the thick wall pixels didn't appear in our work. The results from previous works are summarized in Table 7, using the original image to the final vector result of floorplan No. 290 as an example. Pixels from wall segmentation were changed to centerline and converted to node-edge graph. Corner detection method was used simultaneously to indicate the exact location of corner points, and finally vector data were generated. Further results are provided in Table 8.



 Table 6. Centerline detection results with two algorithms.





Table 8. Generated vector outputs with original and segmentation images.





From the vector data and attribute, the CityGML LOD2 model and cell spaces of IndoorGML thick wall model were generated automatically. Every postprocess was performed with FME (Feature Manipulation Engine) by safe software, the of which results are shown in Table 9. The CityGML Lod2 model consists of wall, interior wall, ground surface, and door information. On the other hand, cell spaces of IndoorGML could be generated. To achieve this, extracted wall and door edges were combined, closing the space in the floorplan.

Table 9. GML data model.



We evaluated the results of vector data generation using a confusion matrix. To determine whether nodes were correctly extracted or not, we compared their locations with the locations of nodes in the original image (Figure 8).



Figure 8. Comparison of (a) nodes generated by ground truth with (b) extracted nodes.

The F1 score was 0.8506, which indicates that extracted nodes are appropriate for use as end points of the edge (Table 10). Table 8 demonstrates that more corners were detected than the actual number of corners, indicating that the false positive value is relatively larger than the false negative value. Many corners were often detected due to misalignment of the raster pixels on the diagonal (Figure 9a). Uncleaned noise in the segmentation image resulted in corners on the cross-section of walls and furniture being regarded as corners. For example, we did not include the outline of an indoor entity in the ground truth label (Figure 9b,c), but misclassified the pixels around the entity as a wall, leading to the generation of two incorrect horizontal vector walls (Figure 9d). If we remove unnecessary nodes of diagonal components and improve the performance of segmentation, we can expect greater accuracy.

	Prediction Actual		Positive		Negative		
	Positive	Positive	2198		267		
	Negative	Negative	386		0		
b	а	а	с	d	e		
ŕ	A / /			ר ר			
L se	××	\mathcal{N}	20	1 -			
	\sim	\sim	M I	1 1-	-		
ľ	× - \	×	R I				

Table 10. Confusion matrix of node extraction.

Figure 9. Wrong Prediction: (**a**) floorplan with diagonal wall; (**b**) original floorplan with furnishing; (**c**) ground truth pixel; (**d**) prediction pixel; (**e**) generated vector.

Finally, wall thickness, as measured by the number of pixels occupied by each wall, was added to the automatically generated vector. An example of the measurement of the thickness of each wall by the method described in Section 2.2.3 is shown in Figure 10. Of the numbers in parentheses, the first number represents wall thickness from the results of wall segmentation, and the second number represents thickness from ground truth.



Figure 10. Number of pixels (estimated, ground truth).

In addition, we defined a "thickness error index to evaluate the wall thickness result for all test sets. The equation of thickness error is as follows:

Thickness Error
$$= \frac{1}{n} \sum \frac{t_{seg} - t_{gt}}{t_{gt}}$$
 (1)

where t_{seg} is the wall thickness from the segmentation result and t_{gt} is wall thickness from the ground truth result.

The thickness error of 10 images was obtained by random sampling of test sets, producing an error value of 0.224. This suggests that we can expect an average error of 1 pixel per five wall pixels. However, this value may be an overestimate, as the error from thin walls raises the overall thickness error. In fact, the $t_{seg} - t_{gt}$ values of almost 80% of walls were lower than the estimated overall thickness error. Thus, we conclude that the wall thickness attribute is sufficiently accurate for use in generating indoor spatial data, such as in CityGML or IndoorGML.

4. Discussion

In this paper, we proposed a process for automatically extracting vector data from floorplan images, which is the basis for constructing indoor spatial information. To the best of our knowledge, this is the first study that takes OGC standards into account when extracting base data for constructing indoor spatial information from floorplan images. Our main contributions are as follows. First, we recovered vector data from floorplan images that could be used to generate indoor spatial information with CityGML LOD2 or a cell space of IndoorGML thick wall model, which was unable to be accomplished with image results from previous studies. Second, we generated high-quality segmentation results by adopting a modern CNN architecture that enabled us to generate reliable node-edge graph-formed vector data with a thickness attribute. Third, we recovered both orthogonal and diagonal lines using recently developed centerline detection algorithms.

To further demonstrate the contributions of our work, some methodologies were briefly compared with our work. Segmenting entities in floorplan images was not an easy task; therefore, generating fine vector data had been challenging and rarely tried before the centerline detection method was applied to the output of learning-based algorithms by [11]. However, since the segmentation result still had some noise and a bumpy shape, the output of the centerline detection method did not look clean; it included many branches caused by the characteristics of centerline detection method itself that had to be removed. This made it more difficult to extract nodes from it in a practical manner. We first overcame these problems by improving the segmentation accuracy using a modern deep learning technique. Although a deep learning technique was already applied in the previous work to segment the entities, the previous work did not consider generation of vector data and focused only on the image process itself. Furthermore, the FCN-2s model in that work was based on FCN, the earliest and most primitive network architecture for segmentation, while our work used Darknet 53 as a backbone, which was proven to be an efficient architecture for classification and object detection tasks. In addition, we took advantage of modern training techniques, such as ADAM optimizers, dice loss, and data augmentation. With many quality results obtained for wall and door segmentation images, the recently developed centerline algorithm was then applied. This showed a reasonable centerline with minimal branches and smoother edges. Thanks to the real shape detected by the centerline, flexible wall description became possible. Previous vector generation methods seldom expressed the complex structure of wall edges successfully, only dealing with rectangular shapes or orthogonal lines. However, our approach included diagonal edges as well. In addition, we went further by generating not just vector data, but by visualizing examples of real spatial data documents defined by OGC standards, such as CityGML and IndoorGML, to show the utility of our result. This work had never been tried before, and here a fully designed method was shown for using 2D floorplan images to generate vector data that can be used as spatial data. In addition, because floorplan images are easy to acquire, we expect that generating a large amount of indoor spatial information automatically, in an accessible and affordable manner, will become possible.

The limitations of our work are as follows. In this study, walls and doors are considered the main objects consisting of spatial data. However, there are other objects such as windows and stairs that could also fulfill this role. They should also be included in further studies to produce semantically rich data. In addition, every coordinate is a relative coordinate that cannot be visualized in widely

used viewers, such as a FME inspector or FZKviewer developed by Karlsruhe Institute of Technology. The coordinate system connecting the real world and generated data should be considered and included in the documents. From a technical viewpoint, the nodes on diagonal edges should be removed to avoid expressing single edges with more than two nodes.

Author Contributions: Conceptualization, Hanme Jang and JongHyeon Yang; methodology, Hanme Jang and JongHyeon Yang; software, Hanme Jang and JongHyeon Yang; validation, JongHyeon Yang; formal analysis, Hanme Jang; investigation, Hanme Jang and JongHyeon Yang; resources, Kiyun Yu; data curation, Hanme Jang and JongHyeon Yang; writing—original draft preparation, Hanme Jang; writing—review and editing, Hanme Jang and JongHyeon Yang; visualization, Hanme Jang and JongHyeon Yang; supervision, Kiyun Yu; funding acquisition, Kiyun Yu. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Spatial Information Research Program (NSIP), grant number 19NSIP-B135746-03 and the APC was funded by Ministry of Land, Infrastructure, and Transport of the Korean government.

Acknowledgments: The completion of this work could not have been possible without the assistance of many people in GIS/LBS lab of Department of Civil and Environmental Engineering at Seoul National University. Their contributions are sincerely appreciated and gratefully acknowledged. To all relatives, friends and others who shared their ideas and supported financially and physically, thank you.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Xu, D.; Jin, P.; Zhang, X.; Du, J.; Yue, L. Extracting indoor spatial objects from CAD models: A database approach. In Proceedings of the International Conference on Database Systems for Advanced Applications, Hanoi, Vietnam, 20–23 April 2015; Liu, A., Ishikawa, Y., Qian, T., Nutanong, S., Cheema, M., Eds.; Springer: Cham, Switzerland, 2015; pp. 273–279.
- 2. Jamali, A.; Abdul Rahman, A.; Boguslawski, P. A Hybrid 3D Indoor Space Model. In Proceedings of the 3rd International GeoAdvances Workshop, Istanbul, Turkey, 16–17 October 2016; The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences: Hannover, Germany, 2016.
- Shirowzhan, S.; Sepasgozar, S.M. Spatial Analysis Using Temporal Point Clouds in Advanced GIS: Methods for Ground Elevation Extraction in Slant Areas and Building Classifications. *ISPRS Int. J. Geo-Inf.* 2019, 8, 120. [CrossRef]
- 4. Zeng, L.; Kang, Z. *Automatic Recognition of Indoor Navigation Elements from Kinect Point Clouds*; International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences: Hannover, Germany, 2017.
- 5. Isikdag, U.; Zlatanova, S.; Underwood, J. A BIM-Oriented Model for supporting indoor navigation requirements. *Comput. Environ. Urban Syst.* **2013**, *41*, 112–123. [CrossRef]
- 6. Gimenez, L.; Hippolyte, J.-L.; Robert, S.; Suard, F.; Zreik, K. Review: Reconstruction of 3D building information models from 2D scanned plans. *J Build. Eng.* **2015**, *2*, 24–35. [CrossRef]
- Macé, S.; Locteau, H.; Valveny, E.; Tabbone, S. A system to detect rooms in architectural floor plan images. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, Boston, MA, USA, 9–11 June 2010; ACM: New York, NY, USA, 2010; pp. 167–174.
- 8. Rendek, J.; Masini, G.; Dosch, P.; Tombre, K. The search for genericity in graphics recognition applications: Design issues of the qgar software system. In *International Workshop on Document Analysis Systems*; Springer: Berlin, Heidelberg, September 2004; pp. 366–377.
- 9. Gimenez, L.; Robert, S.; Suard, F.; Zreik, K. Automatic reconstruction of 3D building models from scanned 2D floor plans. *Automat. Constr.* **2016**, *63*, 48–56. [CrossRef]
- Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Improved automatic analysis of architectural floor plans. In Proceedings of the 2011 International Conference on Document Analysis and Recognition (ICDAR), Beijing, China, 18–21 September 2011; pp. 864–869.
- Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Automatic room detection and room labeling from architectural floor plans. In Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems, Gold Cost, QLD, Australia, 27–29 March 2012; pp. 339–343.
- 12. De las Heras, L.P.; Ahmed, S.; Liwicki, M.; Valveny, E.; Sánchez, G. Statistical segmentation and structural recognition for floor plan interpretation. *Int. J. Doc. Anal. Recog.* **2014**, *17*, 221–237. [CrossRef]

- 13. Dodge, S.; Xu, J.; Stenger, B. Parsing floor plan images. In Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications, Nagoya, Japan, 8–12 May 2017; pp. 358–361.
- 14. Liu, C.; Wu, J.; Kohli, P.; Furukawa, Y. Raster-to-vector: Revisiting floorplan transformation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2195–2203.
- 15. Kang, H.K.; Li, K.J. A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 116. [CrossRef]
- 16. Boeters, R.; Arroyo Ohori, K.; Biljecki, F.; Zlatanova, S. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 2248–2268. [CrossRef]
- 17. Goetz, M.; Zipf, A. Towards defining a framework for the automatic derivation of 3D CityGML models from volunteered geographic information. *Int. J. 3-D Inf. Model.* **2012**, *1*, 1–16. [CrossRef]
- 18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
- 20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
- 21. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 801–818.
- 22. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
- 23. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
- 24. Rajchl, M.; Lee, M.C.; Oktay, O.; Kamnitsas, K.; Passerat-Palmbach, J.; Bai, W.; Damodaram, M.; Rutherford, M.A.; Hajnal, J.V.; Kainz, B.; et al. Deepcut: Object segmentation from bounding box annotations using convolutional neural networks. *IEEE Trans. Med. Imaging* **2016**, *36*, 674–683. [CrossRef] [PubMed]
- Lee, J.; Kim, E.; Lee, S.; Lee, J.; Yoon, S. FickleNet: Weakly and Semi-Supervised Semantic Image Segmentation Using Stochastic Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5267–5276.
- 26. Gramblicka, M.; Vasky, J. Comparison of Thinning Algorithms for Vectorization of Engineering Drawings. *J. Theor. Appl. Inf. Technol.* **2016**, *94*, 265.
- 27. Huang, L.; Wan, G.; Liu, C. An improved parallel thinning algorithm. In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, UK, 6 August 2003; Volume 3, p. 780.
- Boudaoud, L.B.; Solaiman, B.; Tari, A. A modified ZS thinning algorithm by a hybrid approach. *Vis. Comput.* 2018, 34, 689–706. [CrossRef]
- 29. Zhang, H.; Yang, Y.; Shen, H. Line junction detection without prior-delineation of curvilinear structure in biomedical images. *IEEE Access* **2017**, *6*, 2016–2027. [CrossRef]
- 30. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
- 31. Jae Joon, J.; Jae Hong, O.; Yong il, K. Development of Precise Vectorizing Tools for Digitization. *J. Korea Spat. Inf. Soc.* **2000**, *8*, 69–83.
- 32. Or, S.H.; Wong, K.H.; Yu, Y.K.; Chang, M.M.Y.; Kong, H. Highly automatic approach to architectural floorplan image understanding & model generation. *Pattern Recog.* **2005**.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).