Indoor 2D Apartment Model Obfuscation

Prathmesh Deshmukh
Computer Science
Indiana University Bloomington
pdeshmuk@iu.edu

Gavin Henry Lewis
Computer Science
Indiana University Bloomington
gavlewis@iu.edu

Hardik Asnani
Computer Science
Indiana University Bloomington
hasnani@iu.edu

Abstract

When released to the public, the floor plans of the buildings pose security and privacy risks. Our problem statement is "Generating new floor plans by obfuscating room(s) of the actual floor plan." The problem tackled in this paper is to generate an alternate floor plan where we can hide the room layout of a particular region without changing the structure of the floor plan externally. The previous implementations that we have explored talk about layout generation or 3D reconstruction of the floor plans based on the input floor plan. We have explored implementations by replacing the rooms with the closest match in the dataset and generating a new layout using machine learning. To create the solution to the problem, we have used a generative adversarial network (GAN) which is trained on the original and the obfuscated floor plans. The GAN used is cGAN (pix2pix) which has 54 million parameters. This approach generated a new floor plan for the region that needed to be obfuscated in the original floor plan.

1. Introduction

We plan to work on a project that preserves privacy. For instance, there might be situations wherein the entire floor plan of a place (say an airport) cannot be shown to the public. For this, private places in that area must be obfuscated. We plan to work on an idea that involves obfuscating a 2D apartment layout so that it does not reveal the exact layout. Our problem statement is "Generating new floor plans by obfuscating room(s) of the actual floor plan."

The previous approaches that we have explored discuss 3D reconstruction and graph-based techniques to generate the floor plan layout. The problem statement that we are currently targeting is to create a floor plan for the specific region to be obfuscated, keeping the rest of the floor plan intact. In this project, we have implemented a generative adversarial network (GAN) based approach to generate a new floor plan for the region to be obfuscated.

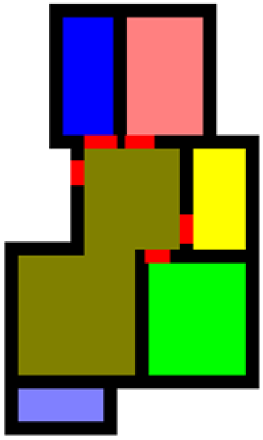
2. Background and Related Work

We have worked on the obfuscation of rooms or a section of the floor plan. This is a new problem statement and the work that we have found is not directly related to the current problem statement in consideration. To understand the pipeline of implementation Prof. Reza has guided us with a clear understanding of how to approach the problem.

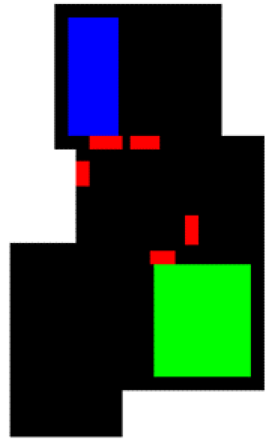
We initially explored the implementation with the Rent3d++ dataset [7] to generate inferences for the floor plan in consideration. But when we obtained the data for the 3D reconstruction of the the floor plan, we realized that the Rent3D++ dataset is a filtered version of the Rent3D dataset where the data is extracted and stored in the non-image format. We also understood the paper Floor-SP [2] that provided us with valuable insight on the mathematical models for the separation and vectorization of the rooms. The energy model that is stated in the paper is highly efficient in determining the room sections so that the room can be extracted efficiently. The algorithm uses room wise coordinate descent so that the room structure is optimized. It first does room segmentation, coordinate descent and then proceeds to loop merging.

The Rent3D dataset [6]is the data we were initially planning to use as it based on the has collected rental ad data of 215 apartments and the number of images in the dataset is 1570. We contacted the authors but we were not able to obtain the data that we needed for the images of floorplans.

The final dataset that we are using in this report is the RPLAN dataset that [9] The dataset has several floorplan data that needs to be compiled to get meaningful insights



(a) Original Floor plan in the dataset



(b) Floor plan with blacked out region for obfuscation

Figure 1. Data Preparation

from the data. The RPLAN dataset has over 80000 real-world floor plans that can be used to train a model that will provide us with efficient outputs. The large dataset is important for us as we cannot train the deep networks or GANS on a smaller dataset.The dataset has 4 channels which depict various regions in the floorplans. The images have the annotation for the boundary and the walls are depicted in black.

We also explored and understood the implementation of the paper Graph2Plan [4] provides us with an understanding of how to generate floor plans from the given layout structure. This paper provides us with valuable insights as to how to proceed with the generation of the floor plans of the obfuscated region. The paper discusses how the graph neural networks can be used to generate the graph layout for the floor plan and thus in turn generate the floor plan as output. The graph2plan paper works on the augmented version of the RPLAN dataset that was discussed in the previous paper. The paper incrementally adds room to the layout model and later proceeds on to generation of the graph layout which will be one of the optimal layout versions of the rooms in the floor plan.

As stated previously this is a new problem and we were not able to find any direct implementation of the problem statement.

2.1. Dataset

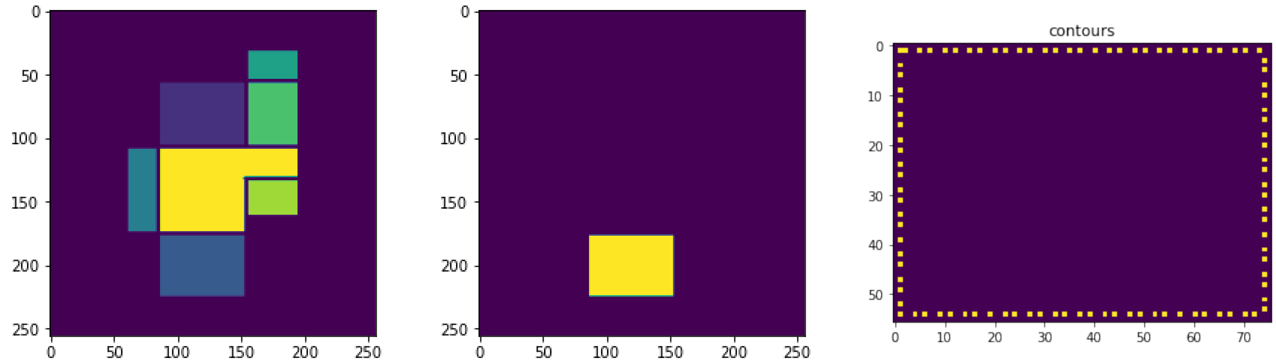
The dataset that we are using is "RPLAN dataset" that is based on the research paper cited [9] . The dataset contains over 80,000 real-world floor plans that have sections of rooms clearly labeled. We have used the room sections in the data to select the rooms that we need for obfuscation.

The RPLAN dataset has a four-channel image (256x256) based obfuscation region. The four channels in the dataset are : the boundary mask, inside mask, wall mask and the room mask. We use the inside mask or the internal room data to obtain the the section that we require for the processing.

2.2. Data Preparation

Room	Pixel Value
Living room	1
Master room	2
Kitchen	3
Bathroom	4
Dining room	5
Child room	6
Study room	7
Second room	8
Guest room	9
Balcony	10
Entrance	11
Storage	12
Wall-in	127
External area	255
Exterior wall	127
Front door	60
Interior wall	127
Interior door	60

The above table shows how we map the various rooms to the pixel values.



(a) Extracted Rooms

(b) Single Room to be obfuscated

(c) Boundary of the room sampled to 100 points

Figure 2. Boundary Extraction and Sampling

Pixel Value	Color Map
1	(0, 128, 128)
2	(0, 255, 0)
3	(255, 0, 0)
4	(0, 255, 255)
5	(255, 255, 0)
6	(255, 0, 255)
7	(128, 128, 128)
8	(128, 128, 255)
9	(128, 255, 128)
10	(255, 128, 128)
11	(128, 128, 0)
12	(128, 0, 128)
127	(0, 0, 0)
255	(255, 255, 255)
60	(0, 0, 255)

The above table is how we map the pixel values to BGR channels in order to generate different colors for the rooms. The doors in the dataset is mapped to the Red color in the BGR channel.

For data preparation we apply a black mask over the image of the floor plan, indicating the region that we need obfuscating. For achieving this task, we randomly select rooms from the dataset and then apply a black mask over the image, making the pixels in that region black. The prepared dataset will contain pairs of the image that our model will learn reconstruction on. The pairs of images are: original image of floor plan as seen in 1a, floor plan with black mask on the rooms to be obfuscated as seen in 1b.

3. Methods

We initially explored two different approaches to obtain a solution of the problem that we have proposed in the paper. The two approaches are:

1. Replacing the room needed to be obfuscated using the chamfer distance of the boundary points of the room. This approach provides us with a similarity score of the room that we needs to be obfuscated and thus we can use appropriate measures to replace it in the new floor plan.

2. Generating a new floor plan of the section of the room that would be needing obfuscation. This approach will provide us with a new generated floor plan that would fit into the region that needs to be obfuscated.

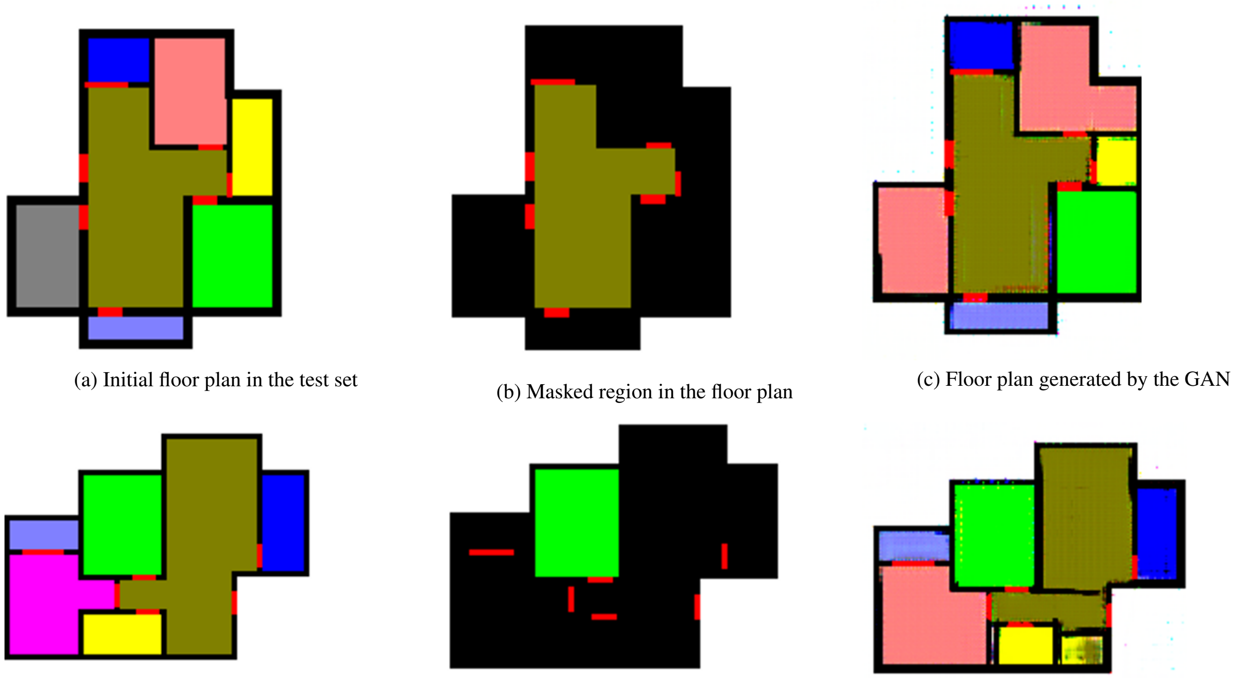
3.1. Nearest Fit Approach

To find the room that we need to replace it with, we first separate out the room we need to obfuscate. The boundary of the room is separated out. The boundary points are extracted into a list form, then this list is stored. The points on the list is then sampled and stored. We are sampling 100 points in the list to generate the point array. Then we plan to compute chamfer distance between the point array of various room output in the given dataset. The chamfer distance would give us the images with the closest match of the points in consideration.

The Chamfer distance (CD) [8] is the evaluation metric that we explored to use as is optimal for calculating the distance that we need for the two pairs of point clouds in consideration. The two point clouds on which this distance is calculated is the initial point cloud and the point cloud of each room in iteration.

From the images shown we can see that the room layout is successfully extracted 2a, then the desired room is separated out 2b and then we find equal distanced 100 points on which we proceed to find the chamfer distance 2c.

This nearest fit approach will ideally give us the list of closest match rooms that we can then extract for the project. We have not explored this approach fully due to time constraint, instead we moved on to the GAN based approach to generate new floor plans.



(a) Initial floor plan in the test set

(b) Masked region in the floor plan

(c) Floor plan generated by the GAN

(d) Initial floor plan in the test set

(e) Masked region in the floor plan

(f) Floor plan generated by the GAN

Figure 3. Understanding the output generated by the GAN

3.2. GAN based approach to generate floor plan

We implemented a GAN-based approach [3] to generate a new floor plan for the region to be obfuscated. The input of the GAN-based method is the data that we obtain from the data preparation section of the paper. This prepared data has the region to be obfuscated blacked out. The implemented GAN is capable of generating a new floor plan that we need for the hidden (blacked out) region.

The pix2pix GAN [5] that we are using helps us to train deep neural networks where image to image translation has to be computed. The model thus is able to understand and gain knowledge for the map that it has to decide for the two image parameters that it gets as input. The knowledge that it gains is conditional to the input it is trained. The discriminator of the GAN provides the appropriate target map plausibility.

For the GAN Model the prepared dataset is dataset is divided into training set (70000 images), validation set (5000 images), and testing set (5000 images).

The generative adversarial network (GAN) that we are using is Conditional GANs(cGANs) which is based on pix2pix that has 54 million parameters. The GAN model was trained using UNET256 generator and PatchGAN discriminator. The learning rate is set to 0.0002 with linear decay in 50 iterations.

The pix2pix implementation that we are using is refered from the source [1]. This provides us with the implementation for the PyTorch adaptation of the pix2pix implementation.

The GAN model was trained on the GPU Tesla P100 for 33 epochs and training set of 70,000 images. The training of the GAN took approximately 11hrs.

The GAN based approach that we are using is the proposed solution to the obfuscation problem that we are trying to takle. Since the pix2pix GAN is able to learn the mapping of the two input images, ie. the original floor plan and the blacked out region in the floor plan. It is able to learn the data that needs to be filled.

4. Results

The trained GAN model generates the new floor plan using the floor plan images that have the region to be obfuscated blacked out. The new floor plan only replaced the blacked out region as required. The external boundaries of the floor plan was never modified.

With the modification of the layout of the walls in the floor plan the GAN was able to generate new rooms as well. The room in the data is labelled using the color the room has. This generation of the floor plan was done on the test split of the dataset.

In the images out the output we can see that the how the initial image available 3a. This initial image is for reference and is not fed to the GAN for inference. The masked image 3b is fed to the GAN for evaluation. The GAN produced the image 3c which as we can see produced a separate layout that is an obfuscation of the initial layout. Thus this approach provides us with results of varying obfuscation levels.

So we can see that the model also is changing the room category which is determined by the color of the room also it is modifying the room layout which is the target of our problem proposed.

5. Discussion

Our research question was, "Generating new floor plans by obfuscating room(s) of the actual floor plan." Through the generated floor plans, we see how GAN based models can generate positive empirical results. It produced high-quality, realistic floor plans. The floor plans are generated quickly once the model is successfully trained with the prepared dataset. On seeing the new floor plans, we are tricked into believing that the new floor plans really exist, but they are all generated.

The results reveal that the approach of generating obfuscated floor plans can be used in the domain of Differential Privacy. This is a new model of cyber security and is a broader research area wherein the information can be publicly shared while withholding crucial information. It should be possible to encode the obfuscated floor plan and decode them back to get the original floor plan when required. The obfuscated floor plan can be encoded and shared with people. Only certain people will know how to decode it, and ultimately, only those people will be able to retrieve the de-obfuscated floor plan. Hence, this can have a good application in Differential Privacy.

Even after successful floor plan generations, there are certain limitations to the GAN based approach. One of the limitations was that we found it hard to evaluate the GAN model because it had no intrinsic evaluation metrics. The best we could do was to inspect the internal layout of the generated floor plans and compare them with those of the real floor plans, but that approach was not reliable. Another limitation was that sometimes, the section to be obfuscated in the generated floor plan was like the section in the original floor plan. It means that if a master bedroom was to be obfuscated from the original floor plan, we observed that instead of being replaced with a new room such as a second bedroom, kitchen, balcony, and so forth, it was still a master bedroom in the generated floor plan.

6. Conclusion

We worked on a problem that involved obfuscation of the 2D apartment layout that ultimately helped in hiding the actual layout. We made use of the Generative Adversarial Network (GAN) based model to achieve the objective of generating new floor plans via obfuscation. RPLAN dataset was used to prepare the final data to train the GAN model and once the training was complete, the model generated new floor plans that looked realistic. However, there was no way to check how good the model was as there was no evaluation metric. Based on the results, we believe that it will be useful in the domain of Differential Privacy where obfuscation and de-obfuscation of the floor plan is required.

The model currently tries to generate the new floor plans that are very much similar to the floor plans that is trained on, as a future work we can create a new organization of the room layouts using the graph based approach as well. [4]

7. Code and Project Repository

The code and the project files can be found in the repository: `https://github.iu.edu/gavlewis/floorplan_obfuscation`

The RPLAN dataset was requested and accessed from the URL: `http://staff.ustc.edu.cn/~fuxm/projects/DeepLayout/index.html`

References

[1] Cyclegan and pix2pix in pytorch. `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`. Accessed: 2022-05-05. 4

[2] Jiacheng Chen, Chen Liu, Jiaye Wu, and Yasutaka Furukawa. Floor-sp: Inverse cad for floorplans by sequential room-wise shortest path. pages 2661–2670, 10 2019. 1

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 4

[4] Ruizhen Hu, Zeyu Huang, Yuhan Tang, Oliver Van Kaick, Hao Zhang, and Hui Huang. Graph2plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2020)*, 39(4):118:1–118:14, 2020. 2, 5

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 4

[6] Chenxi Liu, Alex Schwing, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Rent3d: Floor-plan priors for monocular layout estimation. In *CVPR*, 2015. 1

[7] Madhawa Vidanapathirana, Qirui Wu, Yasutaka Furukawa, Angel X. Chang, and Manolis Savva. Plan2scene: Converting floorplans to 3d scenes. In *CVPR*, 2021. 1

[8] Tong Wu, Liang Pan, Junzhe Zhang, Tai WANG, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. In *In Advances in Neural Information Processing Systems (NeurIPS), 2021*, 2021. 3

[9] Wenming Wu, Xiao-Ming Fu, Rui Tang, Yuhan Wang, Yu-Hao Qi, and Ligang Liu. Data-driven interior plan generation for residential buildings. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 38(6), 2019. 1, 2