Floor Plan Exploration Framework Based on Similarity Distances

Chia-Ying Shih[1] and Chi-Han Peng[1]

National Yang Ming Chiao Tung University[1]

Abstract

Computational methods to compute similarities between floor plans can help architects explore floor plans in large datasets to avoid duplication of designs and to search for existing plans that satisfy their needs. Recently, LayoutGMN [PLF*21] delivered state-of-the-art performance for computing similarity scores between floor plans. However, the high computational costs of LayoutGMN make it unsuitable for the aforementioned applications. In this paper, we significantly reduced the times needed to query results computed by LayoutGMN by projecting the floor plans into a common low-dimensional (e.g., three) data space. The projection is done by optimizing for coordinates of floor plans with Euclidean distances mimicking their similarity scores originally calculated by LayoutGMN. Quantitative and qualitative evaluations show that our results match the distributions of the original LayoutGMN similarity scores. User study shows that our similarity results largely match human expectations.

CCS Concepts

• Computing methodologies → Modeling methodologies;

1. Introduction

To search for similar floor plan designs to copy or to avoid, architectural designers often spend a lot of time exploring floor plans in massive datasets such as RPLAN ([WFT*19]). However, most floor plan similarity calculation methods (including traditional methods such as Intersection-over-Union (IoU) and neural network-based methods such as LayoutGMN [PLF*21]) are not fast enough to support searching in interactive speeds. Therefore, we propose an efficient approach for the goal. In short, we project floor plans into a common low-dimensionality data space such that their Euclidean distances would mimic the similarity scores computed by LayoutGMN, a state-of-the-art floor plan similarity calculation method. In this data space, data exploration tasks such as: 1) searching for similar floor plans, 2) clustering, and 3) pruning of nearly redundant designs, can be efficiently conducted. For example, our method reduced the computation time for task 1) from 296.8 seconds by the original LayoutGMN to just 8.7 seconds. To verify if the similarity measures computed by our method match human expectations, we conducted user studies in questionnaire formats. The results showed that the respondents largely agree with our method's similarity estimations.

Related Work. Determining the similarity between floor plans is a non-trivial task because not only the shapes but also the connectivity of rooms need to be considered. Such "distance" measures between floor plans is a key component for generative methods [NCC*20]. Early approaches only search for specific areas (such as bedroom and living room) [KF05]. Recent methods tend to learn the overall spatial features [SGCM17] and to present floor plans as planar graphs encoding the rooms as vertices [PLF*21].

Recently, LayoutGMN [PLF*21] produced state-of-the-art quantitative and qualitative performances of similarity computation between floor plans by encoding room characteristics (e.g., aspect ratios and locations) and room relationship (e.g., the relative positions and overlapping areas) using an attension-based graph matching network. However, LayoutGMN only computed tripled similarities in which each floor plan don't have a unique feature vector. This means that we cannot simply project the floor plans to a common data space using the feature vectors as coordinates. Another shortcoming of LayoutGMN is that the similarity scores are not upperbounded, further complicating the data projection task.
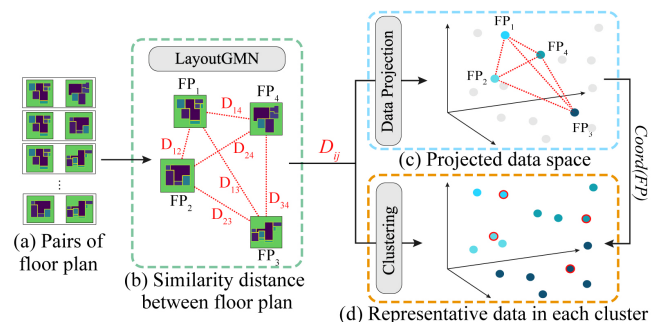
Figure 1: *Overview of our method. (b) Original pair-wise similarity scores between floor plans are computed by LayoutGMN. (c) The floor plans are projected to a common low-dimensional space such that the Euclidean distances mimic the similarity scores, making data exploration tasks such as clustering (d) easier to perform.*

2. Method

Our method is summarized in Figure 1. We denote floor plans as FP_i, $0 \leq i < N$, N is the number of floor plans in the dataset. Recall that LayoutGMN compute similarity between floor plans by computing features vectors (1024 in length) of triples of floor plans and then computing Euclidean distances of the feature vectors as the similarity scores. Note that the feature vector of a floor plan could be different when it is included in different triples. Therefore, they could not be directly used as coordinates in a common 1024-dimensional space. Through experiments, we have found that directly applying traditional dimensionality-reduction techniques such as PCA produced bad results because LayoutGMN tends to produce similarity scores of very large value ranges. Therefore, we instead use cosine similarity to encode distances between two floor plans into the 0 to 1 range as follows:

$$Dist(FP_i, FP_j) = \frac{V_{i,(i,j,k)} \cdot V_{j,(i,j,k)}}{\|V_{i,(i,j,k)}\| * \|V_{j,(i,j,k)}\|} * -1 + 1 \qquad (1)$$

where $V_{i,(i,j,k)} \in R^{1024}$ is the feature vector of floor plan FP_i when calculated among a triple of floor plans FP_i, FP_j, and FP_k by LayoutGMN. Note that LayoutGMN selects the triples such that each triple would include one "anchor" floor plan, one "positive" floor plan that has a higher IoU score to the anchor, and one "negative" one that has a lower IoU score to the anchor, respectively. The similarity score of two floor plans is guaranteed to be the same even if the pair is included in different triples because the pair is still processed by the same GMN module.

Next, we solve for the coordinates of the floor plans in a common data space as a linear least squares optimization problem formulated as follows:

$$\operatorname*{argmin}_{X_i,\ 0 \leq i < N} \sum_{\forall Dist(FP_i,FP_j)\ exists} (\|(X_i - X_j)\| - Dist(FP_i, FP_j))^2 \qquad (2)$$

where X_i is the coordinate of FP_i.

3. Results and Applications

We tested on a computer with Intel i9-10900 2.80GHz CPU and 32GB Rams. We trained LayoutGMN on the whole RPLAN dataset (77664 floor plans) to get 12492731 pairs of similarity scores, and used Google Ceres solver to solve the projected coordinates of the floor plans. We can solve up to 3D data spaces due to memory constraints. Quantitatively speaking, the solving using cosine similarity (which took 56.65 minutes to converge) reduced the residuals from 1.9e+06 (random) to 1.3e+03, hinting a successful convergence. In comparisons, the solving could not converge successfully using original Euclidean distances or log distances. To qualitatively evaluate the projection results, in Figure 2, we show iterations of intermediate solutions of projected coordinates. By coloring the data points with clustering of the similarity scores calculated by LayoutGMN, we find that our projected coordinates largely match the distributions of the original similarity scores.

Applications. Our method enabled searching for similar and dissimilar floor plans of a given floor plan in interactive speeds. With our method, exhaustively comparing a floor plan to all others in the
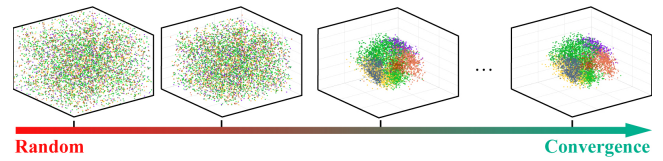
Figure 2: *Projected 3D coordinates of floor plans and their clustering according to the original LayoutGMN similarity scores.*

(a) Input: Real floor plan in RPLAN   (b) Input: user-defined floor plan
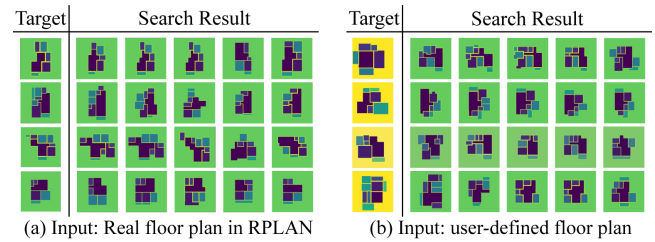
Figure 3: *Searching similar designs by existing (left) or user-defined (right) floor plans.*

dataset took about 8.9 seconds, while using LayoutGMN to compute all the similarity scores took about 296.8 seconds. Users can also insert new floor plans into the dataset. To calculate the coordinate of the new data point, we use LayoutGMN to calculate its similarity scores to all existing floor plans in the dataset, and solve the optimization (Equation 2) again but with all existing data points' coordinates fixed. Figure 3 shows multiple results with existing floor plans in the dataset and novel designs by users as inputs. In general, we found that it is easier to find designs similar to existing entries in the dataset rather than novel designs specified by the user. We think the reason is due to the limited diversity of the dataset. Finally, our method enabled us to detect and prune nearly redundant entries in the original RPLAN dataset. We found 4419 redundant floor plans that are only 50 pixels or less different to others (layout images are of a 256x256 resolution).

User Study. We conducted a questionnaire-based survey of 25 correspondents. Our questionnaire is divided into two parts: ranking and judgment. By calculating the rank correlation, our system obtained a positive correlation result of 63.6% in the ranking part. In the judgment section, there is an 85% pass rate.

4. Conclusion

We propose a method to encode the similarity measurements of LayoutGMN in a common low-dimensional space to facilitate much more efficient exploration. For future work, we would like to explore different ways to encode the similarity scores such as weighted sums of Euclidean distances and cosine similarity.

References

[KF05] KIYOTA K., FUJIYOSHI K.: Simulated annealing search through general structure floor plans using sequence-pair. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 88*, 6 (2005), 28–38. 1

[NCC*20] NAUATA N., CHANG K.-H., CHENG C.-Y., MORI G., FU-
RUKAWA Y.: House-gan: Relational generative adversarial networks for
graph-constrained house layout generation. In *European Conference on
Computer Vision* (2020), Springer, pp. 162–177. 1

[PLF*21] PATIL A. G., LI M., FISHER M., SAVVA M., ZHANG H.:
Layoutgmn: Neural graph matching for structural layout similarity. In
*Proceedings of the IEEE/CVF Conference on Computer Vision and Pat-
tern Recognition* (2021), pp. 11048–11057. 1

[SGCM17] SHARMA D., GUPTA N., CHATTOPADHYAY C., MEHTA S.:
Daniel: A deep architecture for automatic analysis and retrieval of build-
ing floor plans. In *14th IAPR International Conference on Document
Analysis and Recognition (ICDAR)* (2017). 1

[WFT*19] WU W., FU X.-M., TANG R., WANG Y., QI Y.-H., LIU
L.: Data-driven interior plan generation for residential buildings. *ACM
Transactions on Graphics (TOG) 38*, 6 (2019), 1–12. 1