O2-1-4

17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Shape-Based Floor Plan Retrieval Using Parse Tree Matching

Philip K. Lee
Electrical Engineering
Stanford University

Björn Stenger
Rakuten Institute of Technology
Rakuten Group, Inc.

Abstract

Existing floor plan retrieval methods match residential floor plans based on room function and adjacency, ignoring the shape of the interior rooms. Inspired by shape grammars, we incorporate interior layout into the similarity metric using a tree structure that represents both the layout hierarchy and room shapes of the floor plan. We create parse trees from floor plans and evaluate their similarity using an appropriately defined tree edit distance. We evaluate the method on a public dataset of 11,250 vector graphic representations of Japanese homes. A user study shows that our method retrieves layouts preferential to those obtained using the exterior outline or room adjacency matrix.

1 Introduction

Floor plan retrieval engines identify recommendations for home buyers from consumer databases or act as a means for architectural inspiration. The task is frequently framed as a graph matching problem due to the innate representation of floor plans as graphs and the effectiveness of graphs as a pattern recognition tool [2]. Graphs representing features of the floor plan topology or room function have been compared using a variety of graph matching techniques [1, 6, 9, 15, 16, 17, 18, 19, 20, 25, 26]. However, many existing techniques do not explicitly account for the shape of the layout, *i.e.* how the rooms are arranged spatially, which is desirable for finding similar residences in databases. Incorporating features such as the room footprint [1] gives consideration to the shape of the floor plan *exterior*, but it does not weight shapes in the *interior* room layout. A challenge in incorporating the interior layout is that no clear metric permits its comparison. Many metrics for generic shape matching have been proposed [24]. However, homes exhibit a hierarchy of functional zones, which are collections of single rooms [9]. This hierarchical nature is not accounted for when applying a shape similarity metric.

Inspired by shape grammars [21, 30], we approach the floor plan retrieval problem by representing floor plans as trees that can be efficiently compared using the tree edit distance [22]. This distance is the minimum-cost sequence of node edit operations to transform one tree into another, and it can be computed in polynomial time using the Zhang-Shasha algorithm [29]. We developed a cost function for tree edits that incorporates the layout hierarchy. Trees are constructed from vector graphic representations of floor plans, which can be obtained using the method in [7, 10].

Figure 1: Floor plan matching. *We define a distance between floor plan images based on the room layout. Instead of working with the input (top) or segmented images (middle), we use a graph representation (bottom), where edges represent walls and nodes represent intersections. Nodes are colored by degree.*

The retrieval method is shape-based and does not consider the functionality of different rooms. Methods that consider different room functionalities could be incorporated as a filter or as an additional weighted cost in the layout similarity. We applied our method to 11,250 Japanese homes (see Figure 1 for examples) selected from the dataset made available by Liu *et al.* [10]. The novelty of this work is that we incorporate shape hierarchy into shape similarity using a tree edit distance matching metric. We thus provide a solution for floor plan retrieval by introducing a shape matching method for hierarchical structures.

2 Prior Work

Existing approaches to floor plan matching compare functionality and room adjacency and do not explicitly account for room shape in the similarity metric. Spectral graph matching was applied to floor plan matching in [6, 17]. Schaffranek compared office layouts using room function and a weighted adjacency matrix dependent on the physi-

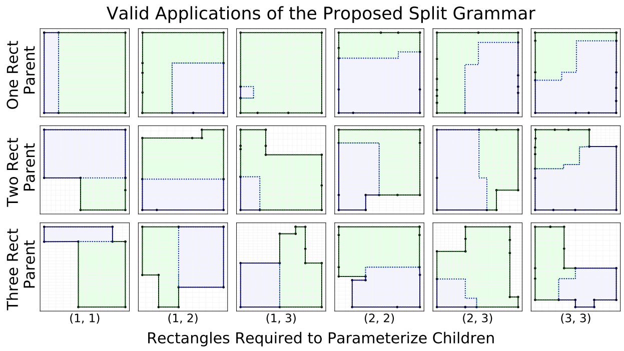

Figure 2: Split grammar for single node. *Each row shows the split for shapes with a different number of parent rectangles. Blue and green colors show areas covered by child nodes.*

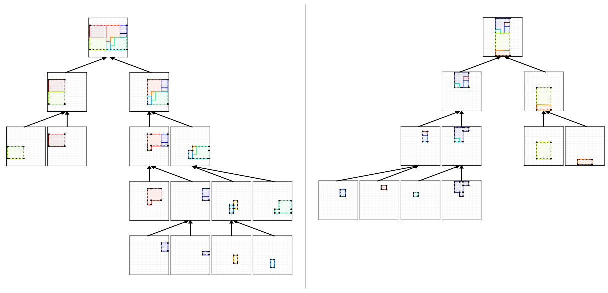

Figure 3: *Floor plan parse trees obtained using greedy area division. Rooms are plotted with different colors. The rooms after the split at the root have similar area and all splits respect room adjacency.*

cal distance between rooms [17]. Hanna compared office layouts but used visibility and boundary features [6]. Langenhan *et al*. created 'semantic fingerprints' by constructing a tree that models the hierarchy of a floor plan [9]. The tree hierarchy represents either rooms, zones, units, or levels. Nodes were labeled with room function and the presence of direct connections, but no shape information is included. Different exact or inexact graph matching techniques were used to query the database [16]. In [26], floor plans are represented as graphs and compared using subgraph embeddings, where nodes contain geometric information such as length and height, shape descriptors, and room types, and are compared using subgraph embeddings. While graphs preserve all room connections and subgraph embeddings encode some higher level shape information, this method discards the shape of each room. Tree matching is computationally efficient and high level shape information is preserved via the parent-child relationship.

2.1 Shape Grammar Representation

Shape grammars [21, 27] follow a set of predefined rules to generate complex shapes and styles including: house layouts [8, 12], Queen Anne-style buildings [5], and facades [3, 11, 14, 27]. Since shape grammars can express many kinds of shapes, we hypothesized that they would form a good representation for floor plan matching based on interior layout. We opted to represent floor plans as a split grammar [27], where each application of the split grammar rule divides the parent shape in two. Split grammars naturally lead to a binary tree representation of floor plans.

3 Floor Plans as Parse Trees

The steps in our floor plan retrieval process are: (a) combine adjacent rooms into parse tree (Section 3.1), (b) assign features (Section 3.2), and (c) compare parse trees (Section 3.3). The parse tree is computed from the planar graph representation of the floor plan which form an undirected graph with corners as nodes and walls as edges.

3.1 Floor Plan Parsing

The goal is to convert a floor plan to a binary parse tree that represent steps of a split grammar. Since floor plans are not well represented by independent splits along coordinate axes, we employ a non-binary split grammar [27]. Binary split grammars as presented by Teboul *et al*. [23] are only capable of dividing shapes into rectangles. A non-binary split grammar permits representation of a more diverse set of room shapes, since shapes after a split can be any pair of basic shapes from the vocabulary. Our parsing problem remains tractable since the floor plan images to be parsed are binary images that are considerably more simple compared to the RGB facade images studied in [23].

We define the split grammar as follows. The basic shapes are in the set $\mathcal{B}$ of shapes formed from the union of three rectangles $\{R_1, R_2, R_3\}$. The rectangles are connected and axis-aligned. Basic shapes have no holes, and are labeled as terminal or non-terminal. At each split, a non-terminal shape is selected and the split rule is applied. A shape is terminal if it contains exactly one room. Examples of valid splits are shown in Figure 2. We found the shape representation to be a reasonable trade-off in terms of accuracy and simplicity, where the majority of shapes in the dataset are accurately captured, while curved shapes lead to an approximation error. The N leaves of the tree represent single rooms. Internal nodes are combinations of adjacent single rooms and have a shape that is the union of its children. Conversely, the two children of an internal node are the result of an applied split. The root node contains all rooms and its shape is the footprint of the floor plan. Nodes are labeled with a feature vector discussed in Section 3.2.

We build the parse tree using a greedy, bottom-up algorithm to obtain near equal area splits close to the root node while respecting room adjacency. The tree parsing process begins with leaf nodes which are initially disjoint. At each iteration, the area of all remaining pairs of disjoint trees (room combinations) that satisfy room adjacency are summed. The pair with the smallest sum-area is combined. This is repeated until all rooms have been combined. Parse trees obtained using this method are shown in Figure 3.

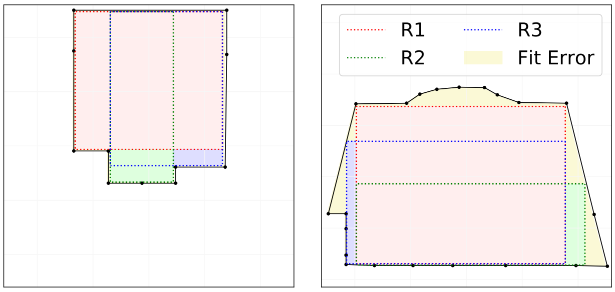

Figure 4: *Examples of room shapes inscribed using rectangles. Dotted outlines represent rectangles used to calculate the affine transforms. A non-rectilinear shape is not completely parameterized by three rectangles.*

Table 1: *Features used in tree edit distance cost.*

Feature	Dim
Width and height of R_1	2
Affine transformations (s_x, s_y, t_x, t_y) that map $R_1 \rightarrow R_2$ and $R_1 \rightarrow R_3$	8
Area of $R_2 \cap \overline{R_1}$ and $R_3 \cap \overline{R_1 \cup R_2}$	2
Area not covered by $R_1 \cup R_2 \cup R_3$	1

3.2 Labeling

Each tree node is labeled with a feature vector obtained from simplifying the room shape. Rooms are simplified to a shape from $\mathcal{B}$ by inscribing up to three overlapping rectangles in the room boundary. R_1 is assigned the largest inscribed rectangle in the room boundary. The largest inscribed rectangle can be solved for a generic polygon by rendering the room footprint and representing it as a Cartesian tree [13]. R_2 and R_3 are inscribed in the remaining area, and extended to maximize overlap with R_1 while remaining within the polygon bounds. Examples of rectangles inscribed using this method are shown in Figure 4. R_1, R_2, and R_3 are used to obtain a translation-invariant feature vector v with 13 values described in Table 1.

3.3 Matching

We use the tree edit distance [22] as a similarity metric between query and candidate parse trees. To compute tree edit distance, the edit operations (relabel, add, delete) are assigned a cost. The trees must also be ordered for tree edit distance to have polynomial complexity [29, 28]. Tree nodes are ordered first by the number of children, then by the room area if the number of children are equal. The cost to relabel node A to B is defined as:

$$C(A, B) = ||area(A)\ v_A - area(B)\ v_B||_1,$$

where, $area(\cdot)$ is the areas covered by the parameterization of the shape at the node, and $|| * ||_1$ is the mean absolute difference. The edit cost is symmetrical and satisfies the triangle inequality [29]. For deletion and insertion, $v_B = \vec{0}$ and $area(B) = 0$. Intuitively, $|| * ||_1$ penalizes differing shapes and the number of inscribed rectangles. The area factors make shapes closer to the root and overall footprint more costly to modify due to their greater area.

4 Experiments

Dataset. 11,250 floorplans were selected from the dataset by Liu *et al*. The cost function and feature vector was designed to provide good matches on 1250 floor plans. The remaining 10,000 were used as a test set for the preference study. The study was conducted using an online form, which participants completed independently. Method names were replaced by letters and the order randomized to avoid bias. Example floor plan retrievals shown in Section 5 are sampled i.i.d. from the set of 1250 floor plans.

Baseline methods. The tree retrieval method was compared to footprint matching [1] and graph matching using the Laplacians of the room adjacency matrix as in [17]. For footprint matching, the distance of the exterior from the centroid of each floor plan was calculated in pixels with rays drawn at increments of $5°$. The matching floor plan minimizes the Euclidean distance between the query and candidate feature vector. For graph matching, the room layout is represented as an undirected weighted graph. Edge weights are the reciprocal of the distance between room centroids if rooms are adjacent and zero otherwise. The l_2-distance of the spectrum of the normalized Laplacian $\mathcal{N}$ was used as the similarity metric [17]. The spectrum is zero padded to the maximum room count to account for different numbers of rooms.

4.1 Layout Preference Study

Twenty-five random floor plans were chosen from the test set. Floor plans that appeared to have missing rooms were removed from the test. Rooms were rendered with a single color, ignoring the room function in order to emphasize the internal shape. For each query, the top four matches obtained using each matching method were presented simultaneously, similar to Figure 5 (with labels removed). Thirty-six participants were asked to select which group was most similar to the query based on layout, room shape and relative sizes of rooms. They were asked to consider the query layout under rotation and reflection. A 'no preference' option was provided for the case that the participant felt no groups were similar to the query.

5 Results

5.1 Comparison of Matching Methods

Figure 5 shows results for query plans from the development set where floor plans with identical footprints exist but have vastly different interior layouts. Spectral matching finds layouts with similar room clustering, but does not respect the footprint of the floor plan. The spectral matching approach is preferential towards floor plans with the same number of rooms. In Figure 5(a), the shape matching preserves the division of the large room and balcony combined with an opposite half with a simple layout and few rooms. Since many plans with the query footprint exist, the footprint match chooses floor plans with a different

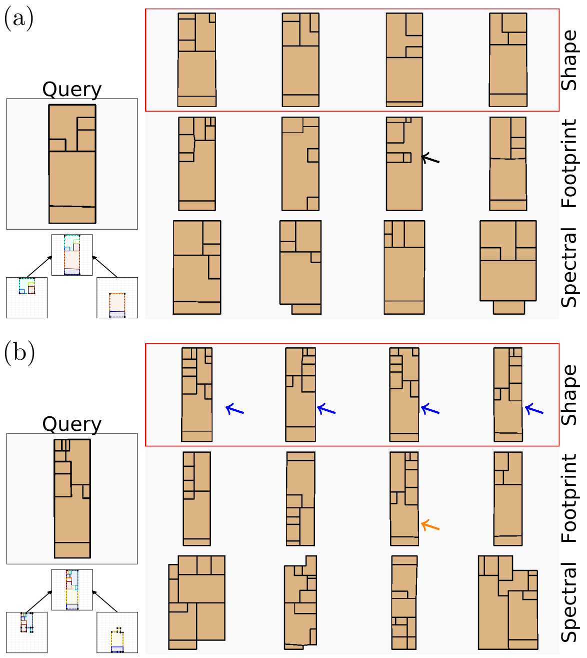

Figure 5: Floor plan matches *with identical footprints.*

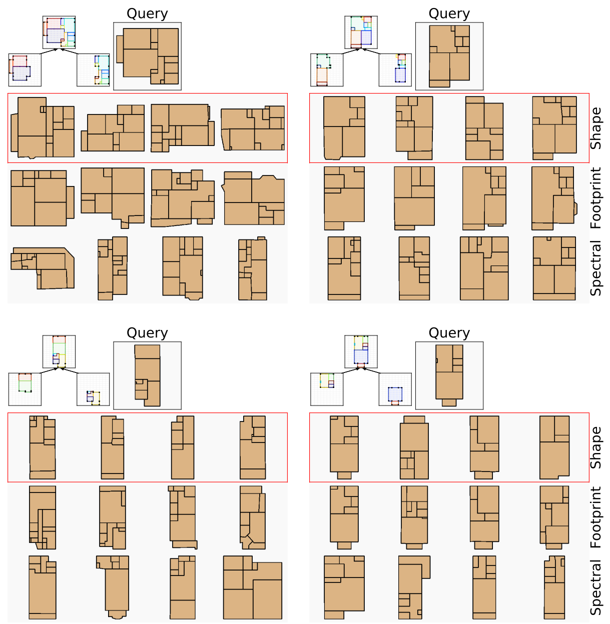

Figure 6: Floor plan matches *without identical footprints.*

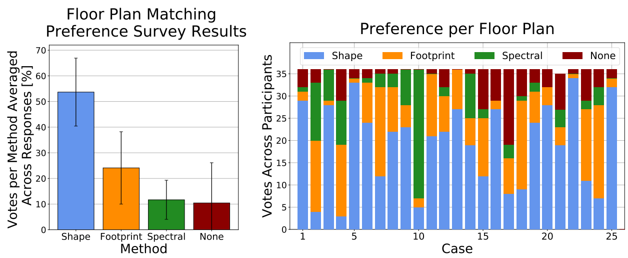

Figure 7: User study results. *The average participant preferred the proposed shape-based method compared to methods based on the foot print or room adjacency matrix. Error bars indicate standard deviation across participants.*

interior layout. In Figure 5(b), there is a distinctive 'T' feature (indicated by blue arrows) that appears in all matches obtained by the proposed shape-based method. In the footprint match, this feature incidentally appears as indicated by the orange arrow. Figure 6 shows matches obtained for floor plans without identical footprints. The shape matching method discovers floor plans with similar footprints that better adhere to the division and complexity of the query.

5.2 Preference Study on Test Set

Parsing floor plans in the test set took approximately 10 hours, but only needs to be calculated once. We computed the tree edit distance against 10,000 candidates, and the average response for each category, and responses for each of the twenty-five cases are shown in Figure 7. The most preferred method of the average participant was the shape-based method ($53\% \pm 11\%$). Inter-user variability was relatively high with shape-based preference ranging from a minimum of 16% to a maximum of 76%. Users with less preference for the shape-based method were more attentive to footprint and clustering rather than individual room shape. Cases where the spectral method was selected the majority of the time had simple footprints and small numbers of rooms.

6 Discussion

We have presented a data structure that permits comparison of floor plans according to their shape. It performs best at comparing floor plans with similar or identical footprints.
Some room shapes could not be represented by our shape simplification method. An approach to addressing the diversity of room shapes would be to apply a different method for computing shape similarity and replace the mean absolute difference in the edit distance. Options include: parameterizing the shape with triangles [4], using a radius feature akin to the footprint calculation, or applying other shape similarity methods [24]. Room function adjacency was ignored in our similarity metric, but could be incorporated as an additional weighted term in the similarity metric [16].

7 Conclusion

We have presented a method for incorporating layout shape in the floor plan retrieval process. This was achieved by parsing floor plans into a tree structure that represents a split grammar and comparing the parse trees with a shape-based tree edit distance. A preference survey indicated that our retrieval method identifies layouts with higher similarity to the query than layouts obtained using previously proposed retrieval methods.

References

[1] Stanislas Chaillou. *AI + Architecture Towards a New Approach*. PhD thesis, Harvard University, 2019.

[2] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty Years of Graph Matching in Pattern Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[3] C. Dore and M. Murphy. Semi-Automatic Modelling of Building Façades With Shape Grammars Using Historic Building Information Modelling. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5/W1:57–64, 2013.

[4] Pedro F. Felzenszwalb. *Representation and Detection of Shapes in Images*. PhD thesis, MIT, 2003.

[5] Ulrich Flemming. More than the sum of parts: the grammar of Queen Anne houses. *Environment and Planning B: Planning and Design*, 14(3):323–350, 1987.

[6] Sean Hanna. Representation and generation of plans using graph spectra. In *Proceedings of the 7th International Space Syntax Symposium*, Stockholm, 2007.

[7] Ahti Kalervo, Juha Ylioinas, Markus Häikiö, Antti Karhu, and Juho Kannala. CubiCasa5K: A Dataset and an Improved Multi-task Model for Floorplan Image Analysis. *SCIA*, 2019.

[8] H Koning and J Eizenberg. The Language of the Prairie: Frank Lloyd Wright's Prairie Houses. *Environment and Planning B: Planning and Design*, 8(3):295–323, 1981.

[9] Christoph Langenhan, Markus Weber, Marcus Liwicki, Frank Petzold, and Andreas Dengel. Sketch-based Methods for Researching Building Layouts through the Semantic Fingerprint of Architecture. *CAAD Futures*, pages 85–102, 2011.

[10] Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-Vector: Revisiting Floorplan Transformation. In *ICCV*, pages 2214–2222, 2017.

[11] Andelo Martinovic and Luc Van Gool. Bayesian grammar learning for inverse procedural modeling. In *CVPR*, pages 201–208, 2013.

[12] Junsik Moon. *Shape grammar for Mies van der Rohe's highrise apartment*. PhD thesis, MIT, 2007.

[13] Carroll Morgan. *Programming from Specifications*. Prentice Hall, second edition, 1994.

[14] Pascal Müller, Gang Zeng, Peter Wonka, and Luc Van Gool. Image-based procedural modeling of facades. *ACM Transactions on Graphics*, 26(3):85, 2007.

[15] Thamonwan Sa-ngawong and Nagul Cooharojananone. A modified graph representation for room layout matching using spectral embedding. In *23rd International Computer Science and Engineering Conference (ICSEC)*, 2019.

[16] Qamer Uddin Sabri, Johannes Bayer, Viktor Ayzenshtadt, Syed Saqib Bukhari, Klaus-Dieter Althoff, and Andreas Dengel. Semantic Pattern-based Retrieval of Architectural Floor Plans with Case-based and Graph-based Searching Techniques and their Evaluation and Visualization. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods*, pages 50–60, Porto, Portugal, 2017.

[17] Richard Schaffranek. Parallel planning An experimental study in spectral graph matching. In *Proceedings of the 10th International Space Syntax Symposium Parallel*, pages 151:1–14, 2009.

[18] Divya Sharma, Chiranjoy Chattopadhyay, and Gaurav Harit. A unified framework for semantic matching of architectural floorplans. In *23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[19] Divya Sharma, Nitin Gupta, Chiranjoy Chattopadhyay, and Sameep Mehta. A novel feature transform framework using deep neural network for multimodal floor plan retrieval.

[20] Divya Sharma, Nitin Gupta, Chiranjoy Chattopadhyay, and Sameep Mehta. Daniel: A deep architecture for automatic analysis and retrieval of building floor plans. In *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017.

[21] George Stiny. Introduction to shape and shape grammars. *Environment and Planning B*, 7:343–351, 1980.

[22] Kuo-Chung Tai. The Tree-to-Tree Correction Problem. *Journal of the ACM*, 26(3):422–433, 1979.

[23] Olivier Teboul, Iasonas Kokkinos, Loïc Simon, Panagiotis Koutsourakis, and Nikos Paragios. Shape grammar parsing via reinforcement learning. In *CVPR*, pages 2273–2280, 2011.

[24] Remco C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Proceedings - International Conference on Shape Modeling and Applications, SMI 2001*, pages 188–197, 2001.

[25] Raoul Wessel, Ina Blümel, and Reinhard Klein. The room connectivity graph: Shape retrieval in the architectural domain. In *Proceedings of WSCG*, 2008.

[26] Raoul Wessel, Sebastian Ochmann, Richard Vock, Ina Blümel, and Reinhard Klein. Efficient retrieval of 3d building models using embeddings of attributed subgraphs. In *Proc. 20th ACM Conf. Information and Knowledge Management (CIKM 2011)*, 2011.

[27] Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Transactions on Graphics*, 22(3):669, 2003.

[28] Kaizhong Zhang. A Constrained Edit Distance between Unordered Labeled Trees. *Algorithmica*, 15(3):205–222, 1996.

[29] Kaizhong Zhang and Dennis Shasha. Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.

[30] Song-Chun Zhu and David Mumford. A Stochastic Grammar of Images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.