GRAPH REPRESENTATION FOR ROOM LAYOUT MATCHING USING SPECTRAL EMBEDDING

Miss Thamonwan Sa-ngawong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Applied Mathematics and Computational Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2019

การแสดงกราฟสำหรับการจับคู่แผนผังห้องโดยใช้การฝังตัวเชิงสเปกตรัม

น.ส.ธมลวรรณ สง่าวงศ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Thesis Title	GRAPH REPRESENTATION FOR ROOM LAYOUT MATCHING USING SPECTRAL EMBEDDING
By	Miss Thamonwan Sa-ngawong
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Associate Professor NAGUL COOHAROJANANONE, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

.... Dean of the Faculty of Science

(Professor POLKIT SANGVANICH, Ph.D.)

THESIS COMMITTEE

.... Chairman

(Associate Professor PETARPA BOONSERM, Ph.D.)

.... Thesis Advisor

(Associate Professor NAGUL COOHAROJANANONE, Ph.D.)

.... Examiner

(Thap Panitanarak, Ph.D.)

.... External Examiner

(Suriya Natsupakpong, Ph.D.)

ธมลวรรณ สง่าวงศ์ : การแสดงกราฟสำหรับการจับคู่แผนผังห้องโดยใช้การฝังตัวเชิงสเปกตรัม. (GRAPH REPRESENTATION FOR ROOM LAYOUT MATCHING USING SPECTRAL EMBEDDING) อ.ที่ปรึกษาหลัก : รศ. ดร.นกุล คูหะโรจนานนท์

การจับคู่กราฟมีประสิทธิภาพในการค้นหาแบบแปลนที่มีลักษณะคล้ายกันเมื่อข้อมูลแผนผังทางสถาปัตยกรรมมีขนาดเพิ่มมากขึ้น เนื่องจากการคำนวณเพื่อค้นหาการจับคู่แบบแปลนโดยใช้การฝังตัวเชิงสเปกตรัมนั้นใช้เวลาในเพียงแค่ไม่กี่วินาที ดังนั้นวิธีนี้จึงเป็นวิธีที่ได้รับความนิยม อย่างไรก็ตามการสมสัณฐานของแบบแปลนแต่ละแบบทำให้ความถูกต้องแม่นยำในกระบวนการจับคู่ลดลงและกลายเป็นจุดอ่อนของวิธีนี้ ดังนั้นเราจึงเสนอการแสดงกราฟสำหรับการจับคู่แบบแปลนห้องโดยใช้การฝังตัวเชิงสเปกตรัม โดยปกติแล้วการแสดงกราฟของแบบแปลนจะกำหนดโหนดแทนห้องและเส้นเชื่อมแทนการเชื่อมต่อระหว่างห้อง นอกจากนี้การฝังตัวเชิงสเปกตรัมมีวัตถุประสงค์เพื่อค้นหาเวกเตอร์แสดงคุณสมบัติของแต่ละแบบแปลนโดยไม่สนใจความหมายของห้องแต่วิธีที่นำเสนอนี้จัดการกับทั้งความหมายของห้องซึ่งคือการเชื่อมต่อระหว่างพื้นที่ภายนอกและภายในห้องและโครงสร้างของแต่ละแบบแปลนด้วย และเรายังแสดงให้เห็นว่าการเพิ่มโหนดใหม่ขึ้นมานั้นสามารถจัดการกับการสมสัณฐานของกราฟบนแนวคิดทางคณิตศาสตร์ที่เรียกว่าการทดสอบค่าเฉพาะได้ วิธีการที่นำเสนอประกอบด้วย 3 ขั้นตอน ได้แก่ การสกัดแบบแปลน การขยายกราฟทอพอโลยี และการจับคู่แบบแปลน ประสิทธิภาพจากการทดสอบวิธีการของเราแสดงให้เห็นว่าวิธีการที่เรานำเสนอสามารถปรับปรุงความแม่นยำในการจับคู่เพิ่มขึ้นจากวิธีการทั่วไปประมาณร้อยละ 27.81

สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	ลายมือชื่อนิสิต ....
ปีการศึกษา	2562	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6071948323 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORD: floor plan, graph representation, spectral embedding

Thamonwan Sa-ngawong : GRAPH REPRESENTATION FOR ROOM LAYOUT MATCHING USING SPECTRAL EMBEDDING. Advisor: Assoc. Prof. NAGUL COOHAROJANANONE, Ph.D.

Graph matching is efficient to search similar layout when the architectural floor plan data size is increasing. Because the computational time of floor plan matching using spectral embedding is only in seconds, so it is one of the popular methods. However, the isomorphism of each floor plan leads to low accuracy in the matching process and it becomes the weakness of this method. Therefore, we propose a graph representation for room layout matching using spectral embedding. Normally, graph representations of the floor plan define nodes as rooms and edges as connections between rooms. Besides, the graph spectral embedding is to find the feature vector of each floor plan by ignoring the semantic of rooms. Our proposed method also considers both room semantic which is the connection between the area outside and inside the room, and the structure of each layout. Furthermore, we show that by adding an extra node, our method can handle the isomorphism of a graph based on a mathematical idea called eigenvalue testing. There are three main processes in the proposed method: floor plan extracting, appended topology graph and floor plan matching. The performance from our experiment shows that our proposed method improve the matched accuracy from the conventional method by about 27.81 percent.

Field of Study: Applied Mathematics and Computational Science		Student's Signature ....
Academic Year: 2019		Advisor's Signature ....

ACKNOWLEDGEMENTS

I would like to express my very great appreciation to my generous advisor Associate Professor Nagul Cooharojananone for his constructive suggestions for my thesis. Whenever I felt discouraged, he always provided me inspiration and encouragement through obscurity. I gained a lot of experience not only in academics but also in the real-life from his supporting. Furthermore, I would like to thank him for the opportunities and the time he has given me during my master's degree.

Besides, I would like to thank all instructors and staff in Applied Mathematics and Computational Science (AMCS) program who always support me behind my accomplishment.

I would also like to thank all my friends who never let me face the problem alone and made my life happier. Especially my best friend, Nawapan Laocharoensuk for all the help she gave me with her great idea.

Finally, I felt very delighted having full support from my cozy family. I wish to thank from my sincere heart.

Thamonwan Sa-ngawong

TABLE OF CONTENTS

LIST OF FIGURES

Page

CHAPTER I

Introduction

Most architects use existing layouts to refer how architectural floor plans were designed in the past before they plan to build some constructions of houses or condominiums. Furthermore, they use a similar design of floor plans while solving a new architectural problem, in order to give more details on how previous, similar architectural situations were solved. Additionally, most customers search for their specific property from an existing layout to make a decision before buying or renting a house or condominium.

The relationship of rooms is an important choice that customers will consider while they want to buy or rent their condominium. For example, some don't want the bathroom to be inside their bedroom and some want the living room to be connected to the balcony. Furthermore, living in a condominium is very popular nowadays, so the market competition of property sale and rental is very high. Looking up through the layouts to search for a similar floor plan can be a difficult and inconvenient process because of large amount of floor plan data. Moreover, with the modern progress of the digital world, several floor plans are archived in a digital form. Therefore, the floor plan searching techniques must be adopted efficiently. We were inspired to improve the matching technique of layouts to support the growing industry and technology of the world.

Graph in pattern recognition is an approach creating abstractions of the raw data to classify their pattern. The abstractions are used to find the similarity between the objects. Therefore, a good representation of the abstractions should include necessary data to classify objects for an appropriate model. Vector is one of the most popular abstractions using to represent the object because it can be transformed into a matrix space and simply using the Euclidian distance for the similarity. However, vector is not a satisfied descriptor if we want to consider the relationship among the components of the object. In this situation, graph representation is better to compare the similarity of the relationship inside objects. Generally, graph encompasses more details than vector, but its similarity cannot be found directly using the Euclidian distance.

Graph matching methods have proven to be helpful for searching for a query architectural floor plan. Graph plays a major role in floor plan matching (Weber, Liwicki, & Dengel, 2011), but it also has a challenging problem since it is a computationally expensive process when the database is very large. However, its solution can be reducible from polynomial-time called NP-complete problem, indexing with well-founded total order for faster subgraph isomorphism detection (Bunke, 2000) is one of the methods that reduce the computational time, and also significantly reduces the storage amount and indexing time for graphs. The optimal quadratic-time isomorphism of ordered graphs was proposed in (Jiang & Bunke, 1999). A spectral method proposed in (Qiu & Hancock, 2006), (Luo, Wilson, & Hancock, 2003) can be utilized for graph matching by transforming the graphs into a feature vector to be matched. Furthermore, graph spectral embedding tries to connect vector and graph representation instead of using only one representation to greater handle with more flexibility for real applications. This method also reduced the searching time for matching similar floor plans. Although, graph spectral embedding is an interesting method, it has some weakness while using with floor plan matching as shown in Figure 1 which represents that floor plans (a) and (b) have different structures such that the bedroom is the center that connects to another room in (a), and the living room is the center that connects to another room in (b). However, after we use graph spectral embedding, we found that both (a) and (b) have the same component vector, thus, leading to the incorrect result of floor plan matching.

Even though some brute force methods are proposed to search for a similar floor plan in the database, the complexity of computational time when increasing floor plan data still be the obstacle in the matching process. Therefore, we

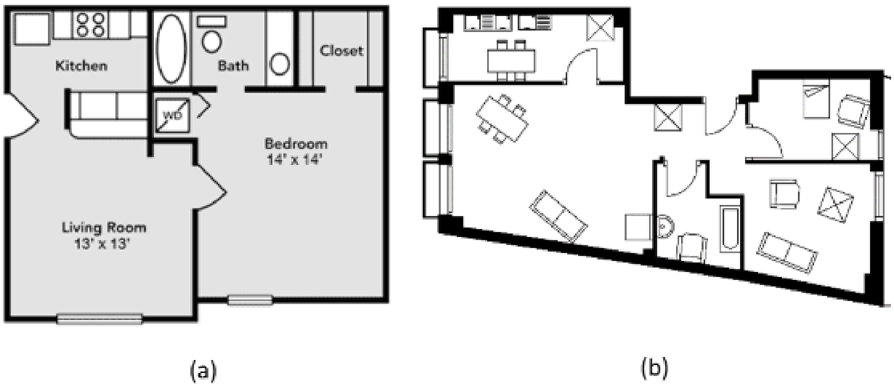


Figure 1: Example of a different structure floor plan: (a) the bedroom is the center that connects to another room, (b) the living room is the center that connects to another room.

have proposed a novel graph representation for room layout matching using spectral embedding to improve the accuracy of the matching process. We enlarge the size of an adjacency matrix from $n \times n$ to $(n+1)\times(n+1)$ by adding a new vertex for a single floor plan with n rooms. The area connecting between outside and inside the layout is the additional vertex. We obtained new feature vectors from the isomorphism of the graph. These refine the similar floor plan by considering both the structure and label of rooms.

Background knowledge and details of the standard methods for floor plan matching are introduced in chapter II. chapter III has discussed the idea of the appended topology graph and the procedure of our proposed method. The result and our experiment are represented in chapter IV. Finally, we conclude the proposed method in chapter V.

CHAPTER II

Background Knowledge and Literature Reviews

In this chapter, we will introduce the techniques that we use in our proposed method. We use graph and topology graph to represent the floor plan. Then, graph isomorphism is used to analyze the structure of each graph. Principal component analysis is used in floor plan matching process. Finally, we divide graph matching into two parts: exact and inexact graph matching.

2.1 Background Knowledge

2.1.1 Graph

Most real-world problems were created into an abstract to make it simple to solve. The graph is also helpful to represent the structure of an architectural floor plan by showing the data and relationship of the floor plan. A graph is a pair of sets (V, E), where V is the set of vertices and E is the set of edges, formed by pairs of vertices. E is a multiset, in other words, its elements can occur more than once so that every element has a multiplicity. Often, we label the vertices with letters (for example: $a, b, c, \ldots$ or $v_1, v_2, v_3, \ldots$) or numbers $1, 2, 3, \ldots$. An example of graph representation shows in Figure 2.

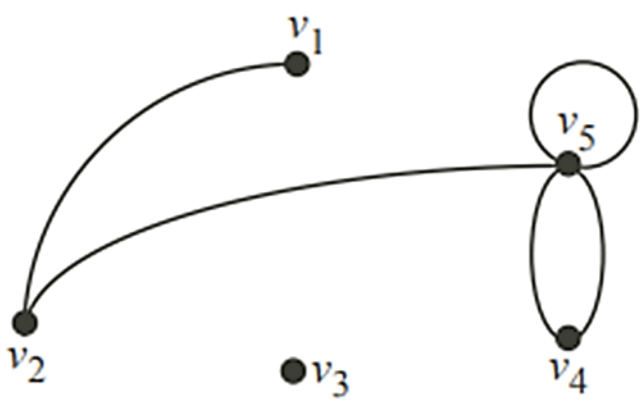


Figure 2: An example of graph representation with five vertices.

From Figure 2, we have $V=\{v_1,v_2,v_3,v_4,v_5\}$ for the vertices and there are edges from v_1 to v_2, v_2 to v_5, v_5 to v_5, v_5 to v_4 and v_5 to v_4, so we have

$E=\{(v_1,v_2),(v_2,v_5),(v_5,v_5),(v_5,v_4),(v_5,v_4)\}$ for the edges. And there is no edge that connect to v_3.

2.1.2 Graph isomorphism

Isomorphism of graphs is the concept to distinguish between two graphs without considering the specific names of the vertices. For example, A and B in Figure 3 are different in their vertex sets, but they are the same graph.

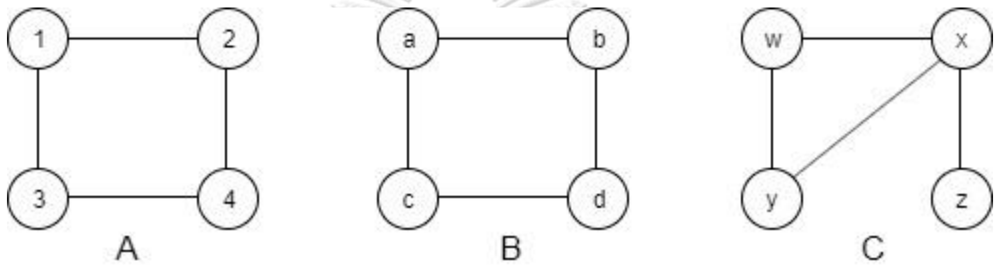


Figure 3: A and B are isomorphic and a non-isomorphic graph C; each have four vertices and four edges.

From Figure 3, all graphs are connected with four vertices and four edges. However, we can distinguish C from others because its structure is different. In addition, D and E in Figure 4 are seem different designs, but they are the same structure because both of them have five vertices and edges, all vertices have degree equal to two, and they have the same adjacency matrix.

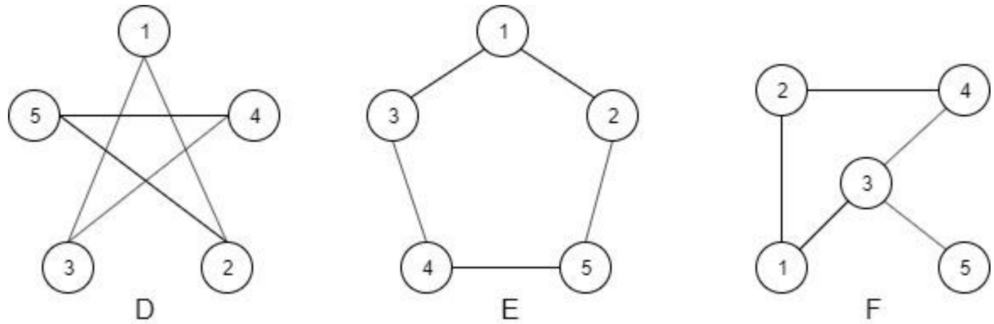


Figure 4: D and E are isomorphic and a non-isomorphic graph F; each have five vertices and five edges.

Definition 1. Two graphs are isomorphic if their vertices can be rearranged and relabeled, without breaking any edges, to make the graphs identical (Ziff, Finch, & Adamchik, 1997).

In Figure 3, we can relabel the vertices of graph A with those of graph B in such a way; 1 is relabeled as a, 2 as b, 3 as c, and 4 as d. Then all edge sets in the relabeled graph is already identical. Regarding the two graphs in Figure 3, we can write $A \cong B$ to denote this isomorphism. On the other hand, we cannot rearrange and relabel the vertices in graph C with A and B, so graph C is not isomorphic to either of A or B.

Normally, if two graphs are isomorphic their properties should be as follows:

• Have the same number of nodes and edges

• Have the same degree lists

• Have exactly the same matrix representation.

For example, all graphs in Figure 4 have the same number of their vertices and edges which equal to five. The degree lists of D and E are (2,2,2,2,2), but the degree lists of F is (1,2,2,2,3). Also, their adjacency matrices are

$$D = \begin{bmatrix} 0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1 \\ 1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0 \end{bmatrix}, E = \begin{bmatrix} 0\,1\,1\,0\,0 \\ 1\,0\,0\,0\,1 \\ 1\,0\,0\,1\,0 \\ 0\,0\,1\,0\,1 \\ 0\,1\,0\,1\,0 \end{bmatrix}, \text{ and } F = \begin{bmatrix} 0\,1\,1\,0\,0 \\ 1\,0\,0\,1\,0 \\ 1\,0\,0\,1\,1 \\ 0\,1\,1\,0\,0 \\ 0\,0\,1\,0\,0 \end{bmatrix}.$$

Thus, we can conclude that graph D is isomorphic to graph E.

2.1.3 Topology graph

A topological graph is a representation of a graph in the plane, where the vertices of the graph are represented by distinct points and the edges joining the corresponding pairs of points. The points representing the vertices of a graph and the arcs representing its edges are called the vertices and the edges of the topological graph. For example, we have $v = \{1,2,3,4\}$ and $E = \{(1,2),(1,3),(1,4)\}$ for the set of vertices and edges of the topology graph in Figure 5.

The topology graph that uses in this work consists of parent node represent layout, child node represents a room, a solid edge represents inclusion, a dashed edge represents adjacencies. An example of a topology graph represented a floor plan is shown in Figure 5.

Figure 5: (a) is an original floor plan, (b) is the room layout segmentation of (a), and (c) is the topology graph of (a).

2.1.4 Principal component analysis (PCA)

Principal Components Analysis (PCA) (Shlens, 2014) is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. It is a way of identifying patterns in data to highlight their similarities and differences.

PCA is a procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. It is mostly used as a tool in exploratory data analysis and often used to visualize genetic distance and relatedness between populations. PCA can be done by eigenvalue decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after a normalization step of the initial data.

The main advantage of PCA is that the information will not lose too much while we reduce the number of dimensions for finding the pattern of the object.

Since the data is high dimensions, its computational is also high. It's also indicating us to difficulty imagine about the relationship of the data. So, PCA can reduce the dimension according to these two following main points

- An unnecessary dimension is removed

- The most important dimension will be kept

For example, our data is 2-dimension as follow

$$Data = \begin{bmatrix} 1.7321 & 1.7321 & 1.6180 & 2.3894 \\ 1.7321 & 1.7321 & 1.6180 & 1.9653 \\ 0 & 0 & 0.6180 & 1.3668 \end{bmatrix}.$$

The process of PCA lead us to compute the covariance matrix, so we have

$$Cov(Data) = \begin{bmatrix} 14.3275 & 13.3142 & 4.2658 \\ 13.3142 & 12.4807 & 3.6861 \\ 4.2658 & 3.6861 & 2.2501 \end{bmatrix}.$$

Next, we calculate the eigenvectors and eigenvalues for this matrix which tell us useful information about our data, then we have

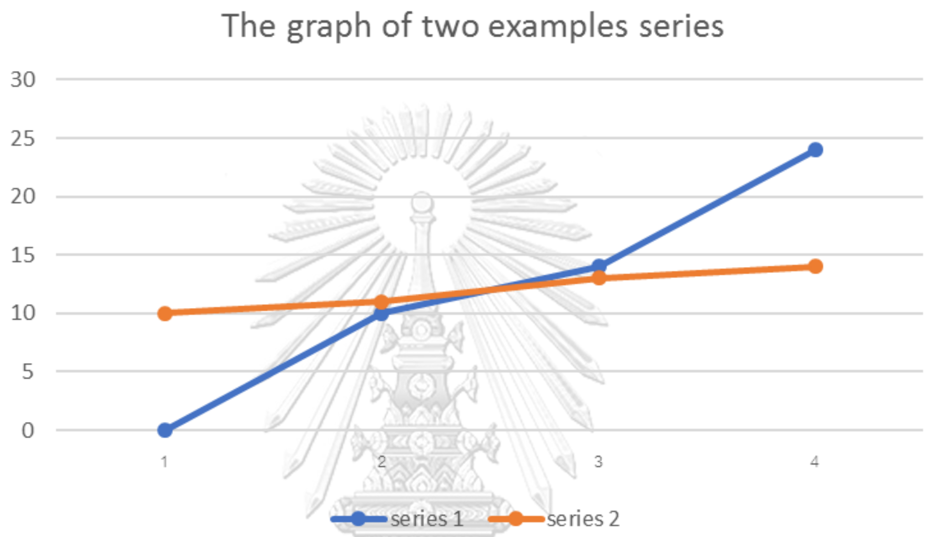
$$eigenvalue = \begin{pmatrix} 0.0139 \\ 1.0607 \\ 27.9836 \end{pmatrix},\ eigenvector = \begin{pmatrix} 0.6986 & 0.0135 & 0.7154 \\ -0.6868 & 0.2929 & 0.6652 \\ -0.2006 & -0.9560 & 0.2139 \end{pmatrix}.$$

Finally, we form our data in a feature vector in one dimension to represent each data.

PCA aims to analyze only especially important data. It uses the statistic process and mathematical tool called Matrix for explaining the data easier to understand. This technique creates a new model to reduce the complexity of the data set as if looking at the old data set with a new viewpoint by not changing the raw data. PCA consists of integrated mathematical techniques such as standard deviation, variance, covariance, covariance matrix, eigenvectors, and eigenvalues.

2.1.4.1. Standard deviation

Standard deviation (S.D.) is the measurement using to explain how well the distribution of data is.



	1	2	3	4
series 1	0	10	14	24
series 2	10	11	13	14

Figure 6: The graph of two examples series.

From Figure 6, the average ($\overline{x}$) of each series are both equal to 12.

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

However, this number cannot distinguish the meaning of these two series. As we see the blue line is more distribution than the orange line. Therefore, there are some statistic technique called standard deviation used to find the distance between each point and the center point as the following equation

$$S.D. = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}\ .$$

Then, from Figure 6, we have $S.D.$ of series 1 = 9.93311, and $S.D.$ of series 2 = 1.825742. So, the standard deviation tells us that series 1 is more spread out from the average than series 2.

2.1.4.2. Variance

Variance as denoting by $\mathrm{var}(X)$ is the variability measurement that measures how the data span. It is the average squared deviation from the mean score. We can compute a variance as the following formula

$$\mathrm{var}(X) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}\ .$$

2.1.4.3. Covariance

Since standard deviation and variance are the technique to analyze the data with one dimension. If we want to analyze the data with more than one dimension, we will have some measurement to distinguish the distribution between two sets of data called covariance. We can compute a covariance as the following formula

$$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n-1}\ .$$

Positive covariance refers X and Y are positively related that means if X increases then Y also increases. The meaning of negative covariance is an exactly opposite relationship. Zero covariance means X and Y are not related. From Figure 6, we have $\mathrm{cov}(series1, series2) = 17.33333,$ so it refers series 1 and series 2 are positively related.

2.1.4.4. Covariance matrix

A covariance matrix is the matrix of all coordinate covariance for each dimension. For instance, if our data consists of three dimensions $X,Y,$ and $Z,$ then the covariance matrix is

$$\begin{pmatrix} \operatorname{cov}(X,X) & \operatorname{cov}(X,Y) & \operatorname{cov}(X,Z) \\ \operatorname{cov}(Y,X) & \operatorname{cov}(Y,Y) & \operatorname{cov}(Y,Z) \\ \operatorname{cov}(Z,X) & \operatorname{cov}(Z,Y) & \operatorname{cov}(Z,Z) \end{pmatrix}.$$

We see that the main diagonal is the variance. Next, we will explain the last two techniques called eigenvectors and eigenvalues used in PCA which are the heart of the data science field. Therefore, the covariance matrix of series 1 and series 2 in Figure 6 is

$$\begin{pmatrix} 98.66667 & 17.33333 \\ 17.33333 & 3.333333 \end{pmatrix}.$$

2.1.4.5. Eigenvectors and eigenvalues

An eigenvector is a vector that remains its direction when we apply a linear transformation to it. The determination of the eigenvectors and eigenvalues of a system is very useful in the principal component analysis (PCA). Each eigenvector is paired with a corresponding eigenvalue. An eigendecomposition is the decomposition of a square matrix A into eigenvectors and eigenvalues. If we calculate the eigenvectors and eigenvalues using Matlab, we will have $[T,D]=eig(A);$ where A is the input matrix, T is the matrix of eigenvectors, and D is the matrix of eigenvalues.

$$T = \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_{\lambda_1} & x_{\lambda_2} & \cdots & x_{\lambda_n} \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

$$\text{and } D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_1 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}.$$

Next, we would like to describe through the main process of principle component analysis.

Step 1: Calculate the covariance matrix C of the data.

Step 2: Calculate eigenvectors and corresponding eigenvalues.

Step 3: Sort the eigenvectors according to their eigenvalues in decreasing order.

Step 4: Choose first k eigenvectors and that will be the new k dimensions.

Step 5: Transform the original n dimensional data into k dimensions.

A new set of dimensions is found by PCA and all dimension are orthogonal. The ranked according to the variance of data along them means more important principal.

2.2 Literature reviews

We have divided graph matching methods for floor plan matching into two categories: exact and inexact graph matching method (Riesen, Jiang, & Bunke, 2010), (Carletti, 2016).

2.2.1 Exact graph matching

Exact graph matching is the method to identify the structure and labels between two graphs. Each graph structure is represented as an adjacency matrix. However, there is no unique order for the nodes of a graph. A single graph with n nodes has $n!$ possibilities to order all nodes, so there are $n!$ different adjacency matrices. Thus, there are many patterns of their adjacency matrices we need to compare. The identity of two graphs g_1 and g_2 is commonly established by defining a function, termed graph isomorphism, that maps g_1 to g_2. Although, there is no polynomial runtime algorithm (Hartmanis, 1982) for the problem of graph isomorphism, many scientists developed polynomial algorithms to solve some specific types of graphs, such as tree (Bunke & Shearer, 1998), ordered graphs (Jiang & Bunke, 1999), planar graphs (Hopcroft & Wong, 1974), bounded-valence graphs (Luks, 1982), and graph with unique node labels (Dickinson, Bunke, Dadej, & Kraetzl, 2004). An example of floor plan matching using exact graph matching method is shown as the following;

Step 1: Create the topology graph for an original floor plan.

From Figure 7, we have $V = \{K, LR, BR, BDR\}$, and
$E = \{(LR, K), (LR, BDR), (LR, BR)\}$

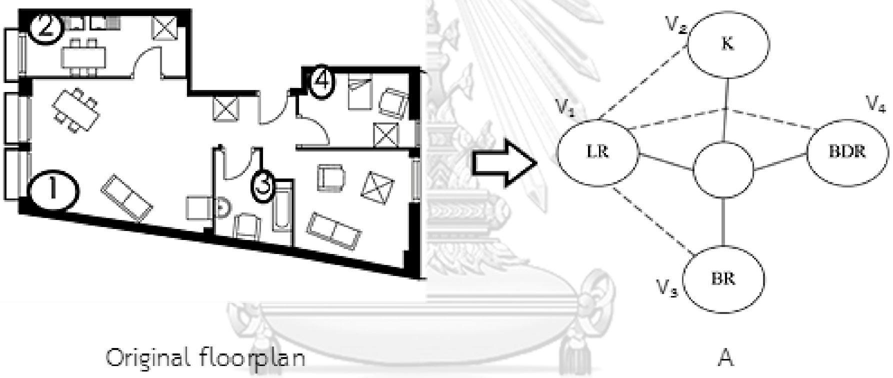
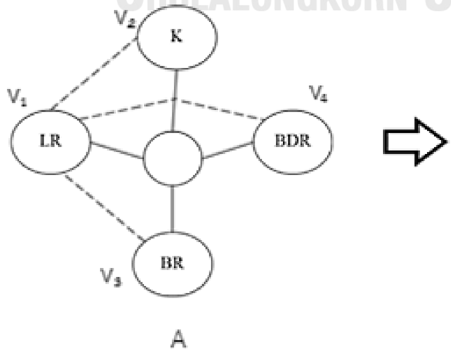


Figure 7: A is the topology graph representing an original floor plan.



	V1	V2	V3	V4
V1	LR	1	1	1
V2	1	K	0	0
V3	1	0	BR	0
V4	1	0	0	BDR

An adjacency matrix

Figure 8: A topology graph (left) and its adjacency matrix (right).

Step 2: Transform the topology graph into an adjacency matrix where the element in row i and column j is equal to 1 if room i connects to room j, and 0 otherwise.

Step 3: Determine all permutations of an adjacency matrix.

Since, there are four rooms in the original floor plan, so we have to generate 24 patterns of an adjacency matrix as shown in Figures 9 and 10.

	V1	V2	V3	V4
V1	LR	1	1	1
V2	1	K	0	0
V3	1	0	BR	0
V4	1	0	0	BDR

(a)

	V1	V3	V2	V4
V1	LR	1	1	1
V3	1	BR	0	0
V2	1	0	K	0
V4	1	0	0	BDR

(b)

	V1	V4	V2	V3
V1	LR	1	1	1
V4	1	BDR	0	0
V2	1	0	K	0
V3	1	0	0	BR

(c)

	V1	V2	V4	V3
V1	LR	1	1	1
V2	1	K	0	0
V4	1	0	BDR	0
V3	1	0	0	BR

(d)

	V2	V1	V3	V4
V2	K	1	0	0
V1	1	LR	1	1
V3	0	1	BR	0
V4	0	1	0	BDR

(e)

	V2	V3	V1	V4
V2	K	0	1	0
V3	0	BR	1	0
V1	1	1	LR	1
V4	0	0	1	BDR

(f)

	V2	V3	V4	V1
V2	K	0	0	1
V3	0	BR	0	1
V4	0	0	BDR	1
V1	1	1	1	LR

(g)

	V4	V1	V2	V3
V4	BDR	1	0	0
V1	1	LR	1	1
V2	0	1	K	0
V3	0	1	0	BR

(h)

	V2	V4	V1	V3
V2	K	0	1	0
V4	0	BDR	1	0
V1	1	1	LR	1
V3	0	0	1	BR

(i)

	V2	V4	V3	V1
V2	K	0	0	1
V4	0	BDR	0	1
V3	0	0	BR	1
V1	1	1	1	LR

(j)

	V2	V1	V4	V3
V2	K	1	0	0
V1	1	LR	1	1
V4	0	1	BDR	0
V3	0	1	0	BR

(k)

	V3	V1	V2	V4
V3	BR	1	0	0
V1	1	LR	1	1
V2	0	1	K	0
V4	0	1	0	BDR

(l)

Figure 9: Matrices (a)-(l) show permutations of an adjacency matrix of an original floor plan.

	V3	V2	V1	V4
V3	BR	0	1	0
V2	0	K	1	0
V1	1	1	LR	1
V4	0	0	1	BDR

(m)

	V3	V4	V1	V2
V3	BR	0	1	0
V4	0	BDR	1	0
V1	1	1	LR	1
V2	0	0	1	K

(n)

	V3	V1	V4	V2
V3	BR	1	0	0
V1	1	LR	1	1
V4	0	1	BDR	0
V2	0	1	0	K

(o)

	V3	V2	V4	V1
V3	BR	0	0	1
V2	0	K	0	1
V4	0	0	BDR	1
V1	1	1	1	LR

(p)

	V3	V4	V2	V1
V3	BR	0	0	1
V4	0	BDR	0	1
V2	0	0	K	1
V1	1	1	1	LR

(q)

	V4	V2	V1	V3
V4	BDR	0	1	0
V2	0	K	1	0
V1	1	1	LR	1
V3	0	0	1	BR

(r)

	V4	V2	V3	V1
V4	BDR	0	0	1
V2	0	K	0	1
V3	0	0	BR	1
V1	1	1	1	LR

(s)

	V4	V3	V1	V2
V4	BDR	0	1	0
V3	0	BR	1	0
V1	1	1	LR	1
V2	0	0	1	K

(t)

	V4	V1	V3	V2
V4	BDR	1	0	0
V1	1	LR	1	1
V3	0	1	BR	0
V2	0	1	0	K

(u)

	V4	V3	V2	V1
V4	BDR	0	0	1
V3	0	BR	0	1
V2	0	0	K	1
V1	1	1	1	LR

(v)

	V1	V3	V4	V2
V1	LR	1	1	1
V3	1	BR	0	0
V4	1	0	BDR	0
V2	1	0	0	K

(w)

	V1	V4	V3	V2
V1	LR	1	1	1
V4	1	BDR	0	0
V3	1	0	BR	0
V2	1	0	0	K

(x)

Figure 10: Matrices (m)-(x) show permutations of an adjacency matrix of an original floor plan.

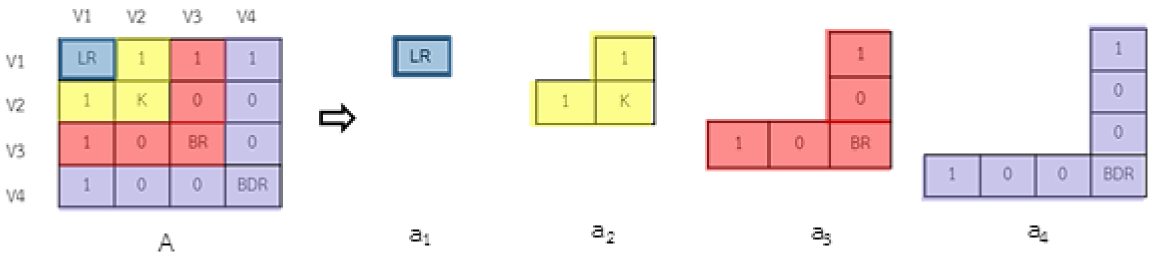


Figure 11: a_1-a_4 are the row-column vector of an adjacency matrix A.

Step 4: Transform all adjacency matrices into row-column vectors.

From Figure 11, we have four row-column vectors of an adjacency matrix A as the following:

$$a_1 = (LR)$$
$$a_2 = (1, K, 1)$$
$$a_3 = (1, 0, BR, 0, 1)$$
$$a_4 = (1, 0, 0, BDR, 0, 0, 1)$$

Step 5: Create the decision tree for an adjacency matrix.

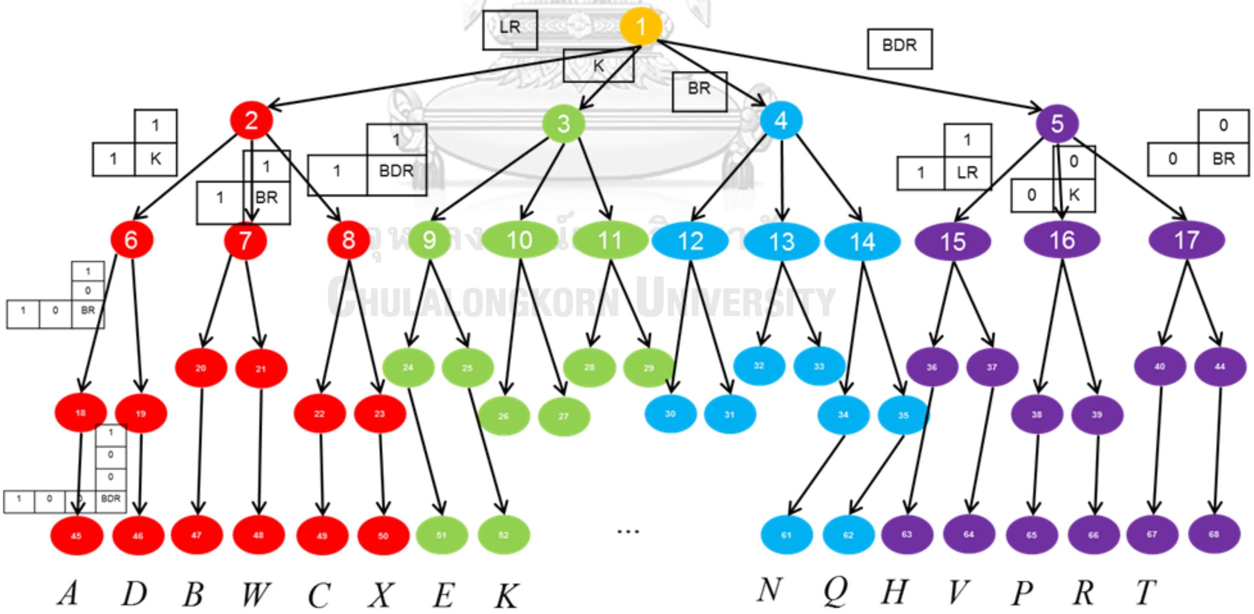


Figure 12: A decision tree for graph A.

We need to generate the decision tree using $n!$ patterns of all adjacency matrices to compare with only one floor plan in the database, so this approach has a major drawback from finding all adjacency matrices as we see in Figure 12.

The major advantage of exact graph matching method is their strict definition and substantial mathematical foundation. However, it is required that the corresponding node and edge labels in the two graphs have to be identical.

2.2.2 Inexact graph matching

Because of the fact that it is inflexible to use the exact graph matching method in a real-world application when the attributes are different in shape or distortion. Many scientists propose an inexact graph matching to handle with inflexible errors from an exact method. Finding a descriptor of each graph to detect similarities between two graphs is the general concept of inexact graph matching method. The descriptors that used to compare the structure of graphs also depend on their representation. Thus, the additional important features of a pattern recognition system were affected by the representation. Vector is the maximum descriptor used among all the possible representations. The floor plans are described as vectors by extracting a finite set of numerical features.

Graph spectral embedding is an interesting inexact graph matching technique since its approach depends on the decomposition of the matrices corresponding to the structure of floor plans. The matrices will obtain the same eigendecomposition if two graphs are isomorphic. This technique directly finds a descriptor expanded for vectorial object descriptions, so it is more convenient than other methods. Generally, the converse from the equality of eigendecompositions to graph isomorphism is not true. Hence, it is not guaranteed that the same feature vector, will be matched with the same floor plan. In addition, the drawback of this approach is that they are rather sensitive towards structural errors such as missing or some nodes. These main problems motivate us to improve this technique to be more accurate in the matching process.

In summary, both exact and inexact graph matching is differently useful for floor plan matching. However, the node and edge labels used to find the similarity in exact graph matching method have to be identical. Thus, the inexact graph matching method better handles with a more general floor plan than exact graph matching. Our proposed method, consider the region outside the floor plan in the extracting process.

We improve accuracy in the matching process under the converse from the equality of eigendecompositions to graph isomorphism by fixing the room through graph spectral embedding (Carletti, 2016; Sharma, Chattopadhyay, & Harit, 2016).

CHAPTER III
Proposed Method

In this work, we proposed a graph representation for room layout matching using spectral embedding. The basic idea of this method is to find a descriptor for each floor plan using the eigendecomposition, then use it to compare the similarity between the query floor plan and floor plans in our database. Although, Graph spectral embedding is effective for floor plan matching (Chung & Graham, 1997), (Fischler & Elschlager, 1973), there are some drawbacks. If two graphs of floor plans are isomorphic, it is not guaranteed that the structure of room relationship is similar. Therefore, our proposed method tries to handle with this problem by adding the area outside the room as the external node into the original graph in the appended topology graph process.

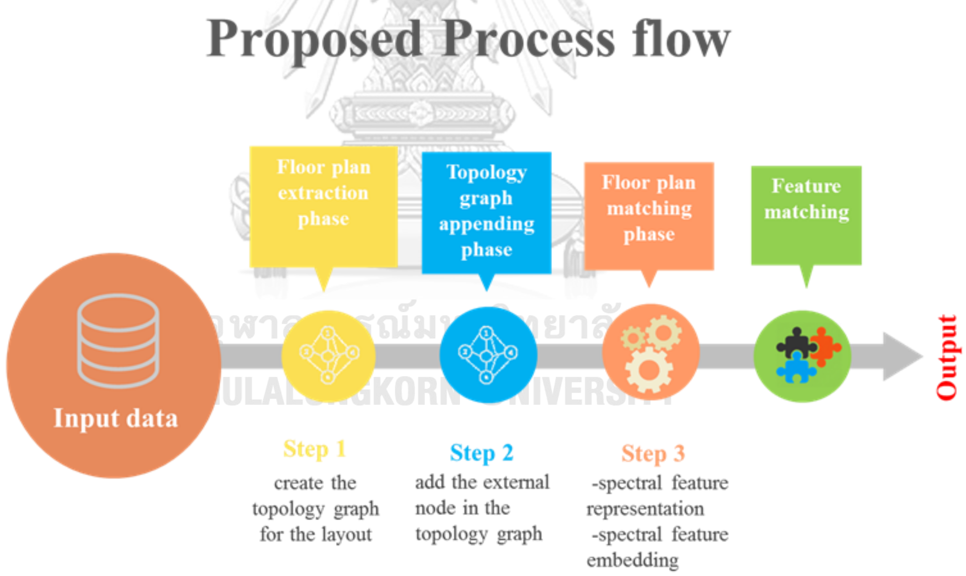


Figure 13: The process flow of our proposed method.

We modify a structure of graphs by putting more details which are the position outside connected to inside the layout. The method can handle with two isomorphic graphs that their relation of rooms is different from each other. In addition, we obtain new feature vectors from eigendecomposition and the result is more efficient than a conventional method. The proposed method is divided into three parts: 1) floor plan extracting, 2) appended topology graph, and 3) floor plan matching. We create a topology graph for the original floor plan in the first step, then add the external node during the second step. Finally, we use graph spectral embedding to match the query floor plan with floor plans in our database.

3.1 Floor plan extracting

We create the topology graph from the original floor plan image as shown in Figure 14. The attributes of the topology graph are that a parent node represents layout, a child node represents a room, a solid edge represents inclusion, and a dashed edge represents adjacency.

3.2 Topology graph appending

The external node which is the connection between outside and inside the room will be added into the original topology graph. Graph spectral embedding is very sensitive when its structure has been changed, so adding more vertices would change the eigendecomposition result. The paradigm illustrates in Figure 15. Thus, the proposed method significantly matches between the feature vector and its corresponding floor plan more accurately.

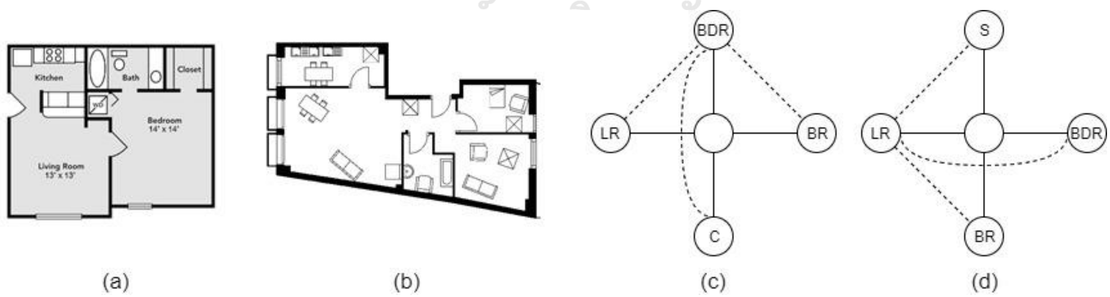


Figure 14: (a) and (b) are the original floor plan, (c) is the topology graph represent (a), and (d) is the topology graph represent (b).

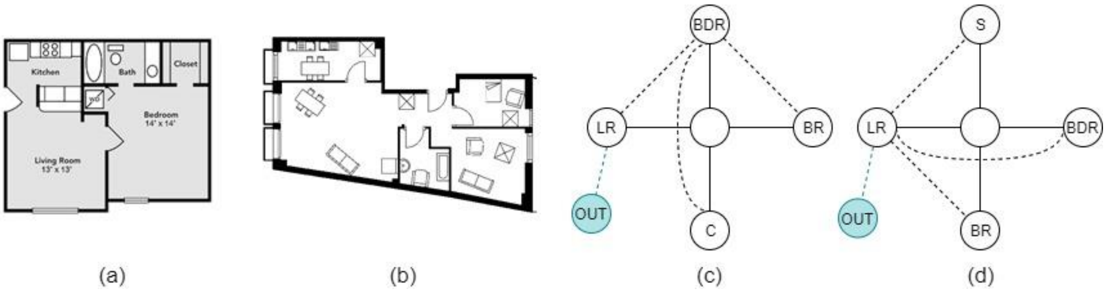


Figure 15: (a) and (b) are the original floor plan, (c) is the appended topology graph represent (a), and (d) is the appended topology graph represent (b).

3.3 Floor plan matching

There are three main steps for floor plan matching: spectral feature representation, spectral feature embedding, and feature matching.

3.3.1 Spectral feature representation

Spectral feature representation is the method to represent each graph in terms of its eigenvalue and eigenvector as follow:

Step 1: Given N images of the floor plan in our dataset.

Step 2: Let $G_k = (V_k, E_k)$ be the k^{th} graph,

where V_k is the set of vertices.

E_k is the set of edges.

Afterwards, graphs $G_1, G_2, ..., G_N$ are represented for all floor plans.

Step 3: Construct an adjacency matrix A_k for each graph G_k.

This is a $|V_k| \times |V_k|$ symmetric matrix whose element with row index i and column index j is

$$A_k = \begin{cases} 1 & ;\ (i, j) \in E \\ 0 & otherwise \end{cases}$$

From the adjacency matrices $A_k, k = 1, 2, ..., N.$

- Calculate the eigenvalues λ_k by solving the equation

$$| A_k - \lambda_k I |= 0.$$

- The associated eigenvectors ϕ_k by solving the system of equations

$$A_k \phi_k = \lambda_k \phi_k .$$

Spectral feature vector representing the spectrum of the graph ϕ_k is constructed from the top n eigenvalues of A_k taken in decreasing order.

For the k^{th} graph, this vector is:

$$\vec{F}_k = (\lambda_k^1, \lambda_k^2, \lambda_k^3)^T . \quad (1)$$

3.3.2 Spectral feature embedding

We use the concept of principal components analysis (PCA) follow by the parametric eigenspace idea. The reason of using this approach is to classify graphs into a pattern-space in which similar structures are close to one another, and dissimilar structures are far apart. The extracted graph from each image is vectorized. Then we compute the feature vector $\vec{F}_k$ using Principal components analysis. Afterwards, we arrange their spectrums in a matrix as $R = [\vec{F}_1, \vec{F}_2, ..., \vec{F}_N]$ for the different graph representations of the layouts in the database, and compute the covariance matrix as, $C = RR^T$. A spectral decomposition of C results in the eigenvalues ρ and the corresponding eigenvectors $\vec{\varphi}$. The Principal components directions are obtained by using the first three leading eigenvectors of C. Three orthogonal vectors span the co-ordinate system of the eigenspace as $\phi = (\vec{\varphi}_1, \vec{\varphi}_2, \vec{\varphi}_3)$.

This aids in projecting the individual graphs represented by the vectors $\vec{F}_k ; k = 1, ..., N$ on the pattern space as

$$\vec{x}_k = \phi^T \vec{F}_k . \quad (2)$$

Hence, each graph obtained from a layout is represented as a three-component vector $\vec{x}_k$ in the eigenspace as

$$\vec{x}_k = (x_k^{'}, x_k^{''}, x_k^{'''})^T.$$

3.3.3 Feature matching

Feature matching and retrieval of similar layouts is performed by determining the nearness between all the layouts. The distance between query graph's feature vector $\vec{x}_q = (x_q^{'}, x_q^{''}, x_q^{'''})^T$ and that of the model graph's feature vector $\vec{x}_m = (x_m^{'}, x_m^{''}, x_m^{'''})^T$ is calculated as

$$d = \sqrt{(x_q^{'} - x_m^{'})^2 + (x_q^{''} - x_m^{''})^2 + (x_q^{'''} - x_m^{'''})^2}. \tag{3}$$

The similarity of two graphs is measured by using Euclidian distance. Then, rank the order of distances in ascending order. The smallest distance is the most similar floor plan between the query floor plan and the floor plan in our dataset.

3.4 Eigenvalues and non-isomorphism

While it is very difficult to prove that two graphs are isomorphic, it is relatively simpler to prove that two graphs are non-isomorphic (Spielman, 2018). Since two graphs are isomorphic when the correctly relabeled graph have the same matrix representation. They must have the same eigenvalue. Since the eigenvalues of Figures 16 (a) and (b) which calculated by Matlab are clearly different, we can conclude that they are non-isomorphic.

3.5 Non-isomorphic graphs

We can use eigenvalues to show that two graphs are non-isomorphic.

Claim: Two graphs that have different eigenvalues cannot possibly be isomorphic.

Proof: Two isomorphic graphs can be rearranged and relabeled such that they both have the same matrix representation. Thus, they have the same eigenvalues.

Therefore, if two graphs do not have the same eigenvalues, then they cannot possibly be isomorphic (McKay, 1981).

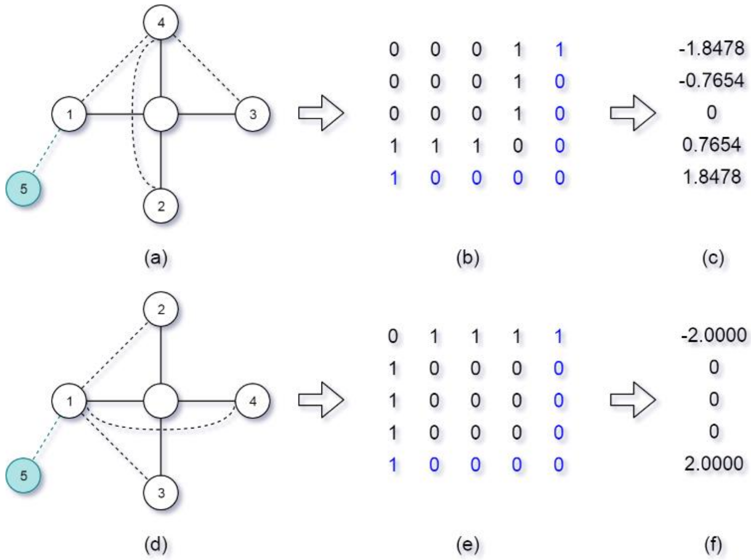


Figure 16: Testing of the isomorphism of graph using eigenvalue: (a) and (d) are the graphs of floor plans, (b) and (e) are adjacency matrices of (a) and (d), respectively. (c) and (f) are eigenvalues of (a) and (d), respectively.

CHEPTER IV

Results and Discussion

We evaluated our method using 800 of floor plan images of condominiums holding almost all characteristics of all floor plan patterns. Our dataset includes eight floor plan types as shown in Table 1.

Table 1: The quantity of each type of floor plan image in our dataset.

number of rooms	2	3	4	5	6	7	8	9
quantity	51	89	159	147	121	65	43	25

There are 11 types of rooms: Living Room, Bedroom, Bathroom, Kitchen, Study Room, Closet Room, Patio, Retreat Room, Exercise Room, Utility Room, and Balcony. There are 38% of floor plans that the center of layout is a living room, 38% that the center are a living room and a bedroom, 16% that the center is a bedroom, and 9% for others. One query floor plan was selected for all different types from floor plans with two to nine rooms. We compare the results from the matching process between a conventional and our proposed methods as shown in Table 2. The first row of each type of floor plans represents the number of correct matched result from our dataset which has the same label and structure with the query floor plan, the second row represents the number of incorrect matched floor plan which has some different in their label and structure with the query floor plan, and the last row represents the percentage of accuracy for matching process. The result shows that the accuracy of our proposed method always greater than or equal to a conventional method for all patterns of query floor plans. The reason is from the isomorphism of floor plan structure. We are not only considering the structure but also handle with both of the structure and the label of rooms for each floor plan. Our method can ignore some possible floor plans that have the same structure but different room labels. Thus, the results of our proposed method would better than the original method.

Table 2: The result of matching process compares between a conventional and proposed method.

Number of rooms		Conventional method	Proposed method
2	Correct	50	50
	Incorrect	0	0
	Recall	100%	100%
3	Correct	118	118
	Incorrect	64	10
	Recall	65%	92%
4	Correct	8	8
	Incorrect	105	1
	Recall	7%	89%
5	Correct	3	3
	Incorrect	78	72
	Recall	4%	4%
6	Correct	25	25
	Incorrect	41	37
	Recall	38%	40%
7	Correct	1	1
	Incorrect	35	2
	Recall	3%	33%
8	Correct	2	1
	Incorrect	8	0
	Recall	20%	100%
9	Correct	2	2
	Incorrect	1	1
	Recall	67%	67%

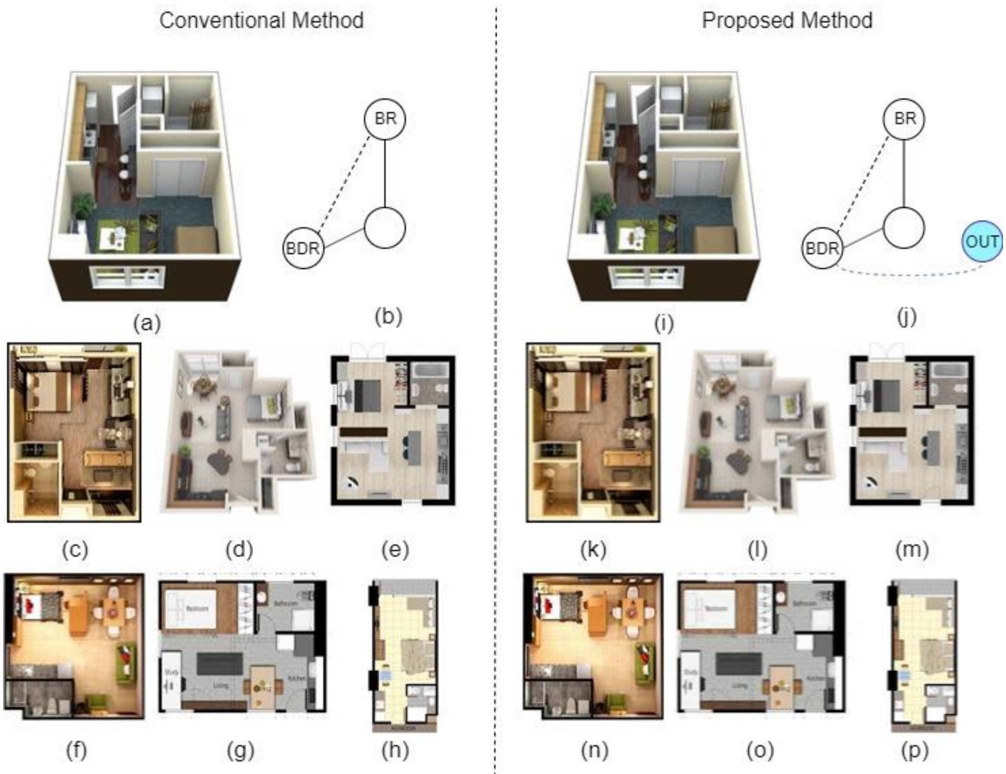


Figure 17: The result of the conventional method (left) and the proposed method (right) using a query floor plan with two rooms.

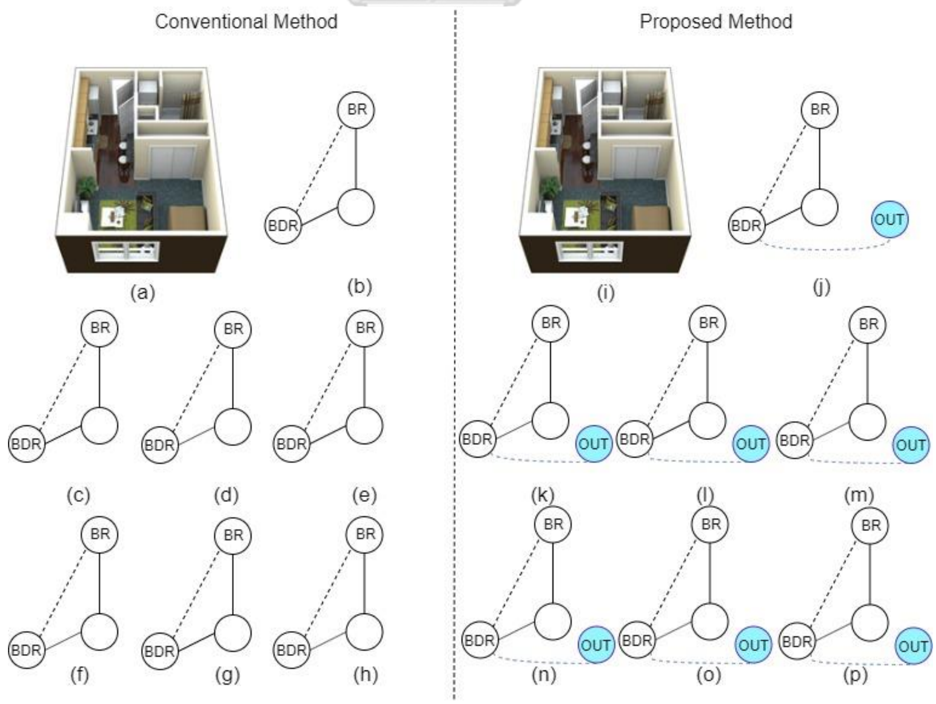


Figure 18: The result of the conventional method (left) and the proposed method (right) using a query floor plan with two rooms in the form of topology graph.

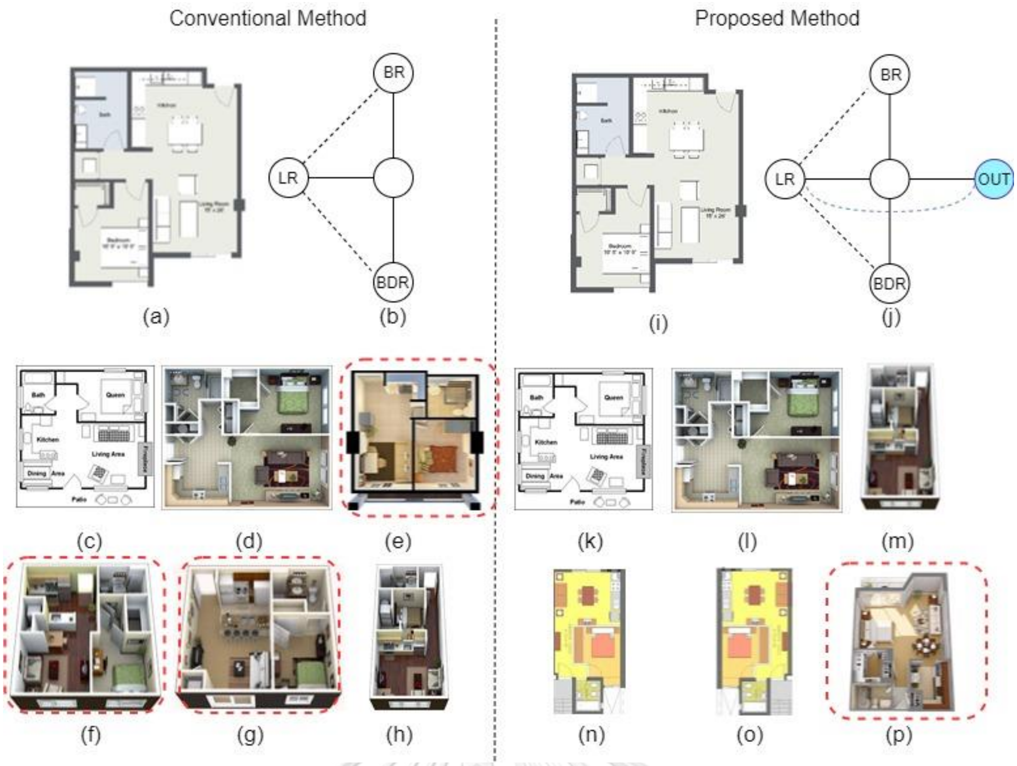


Figure 19: The result of the conventional method (left) and the proposed method (right) using a query floor plan with three rooms.

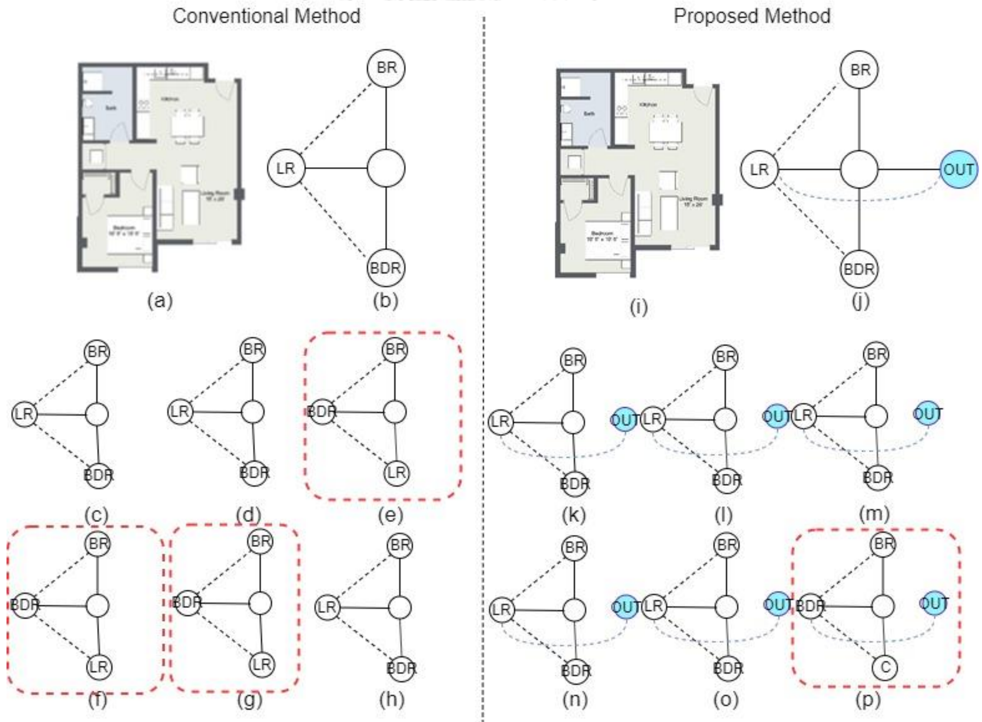


Figure 20: The result of the conventional method (left) and the proposed method (right) using a query floor plan with three rooms in the form of topology graph.

From Figure 17, the top row is the query floor plan with its graph representation. Floor plans (c)-(h) are the result of the conventional method and floor plans (k)-(p) are the result of the proposed method. The structure of the topology graph comparing between both methods is shown in Figure 18. The following results show only the top six order according to the query floor plan.

From Figure 19, the top row is the query floor plan with its graph representation. Floor plans (c)-(h) are the result of the conventional method and Floor plans (k)-(p) are the result of the proposed method. The circle represents an incorrect result.

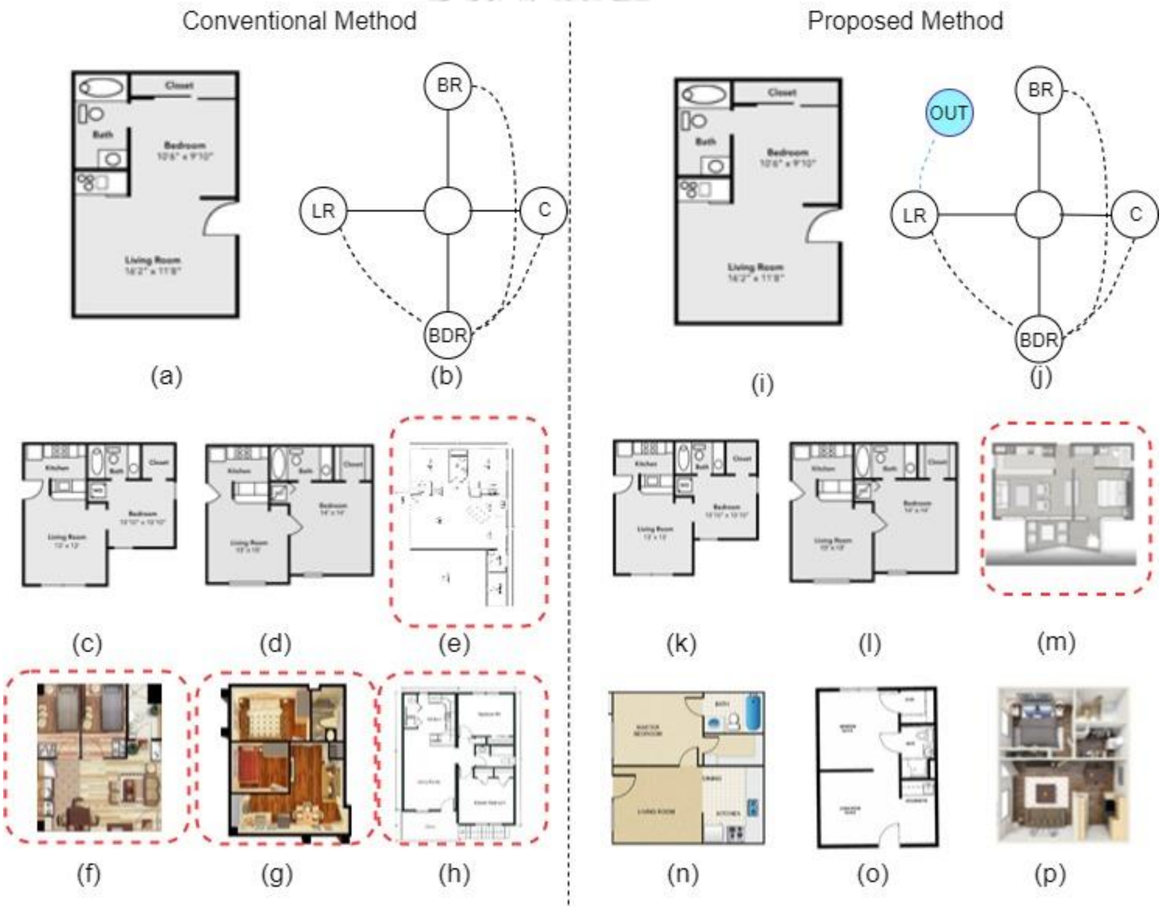


Figure 21: The result of the conventional method (left) and the proposed method (right) using a query floor plan with four rooms.

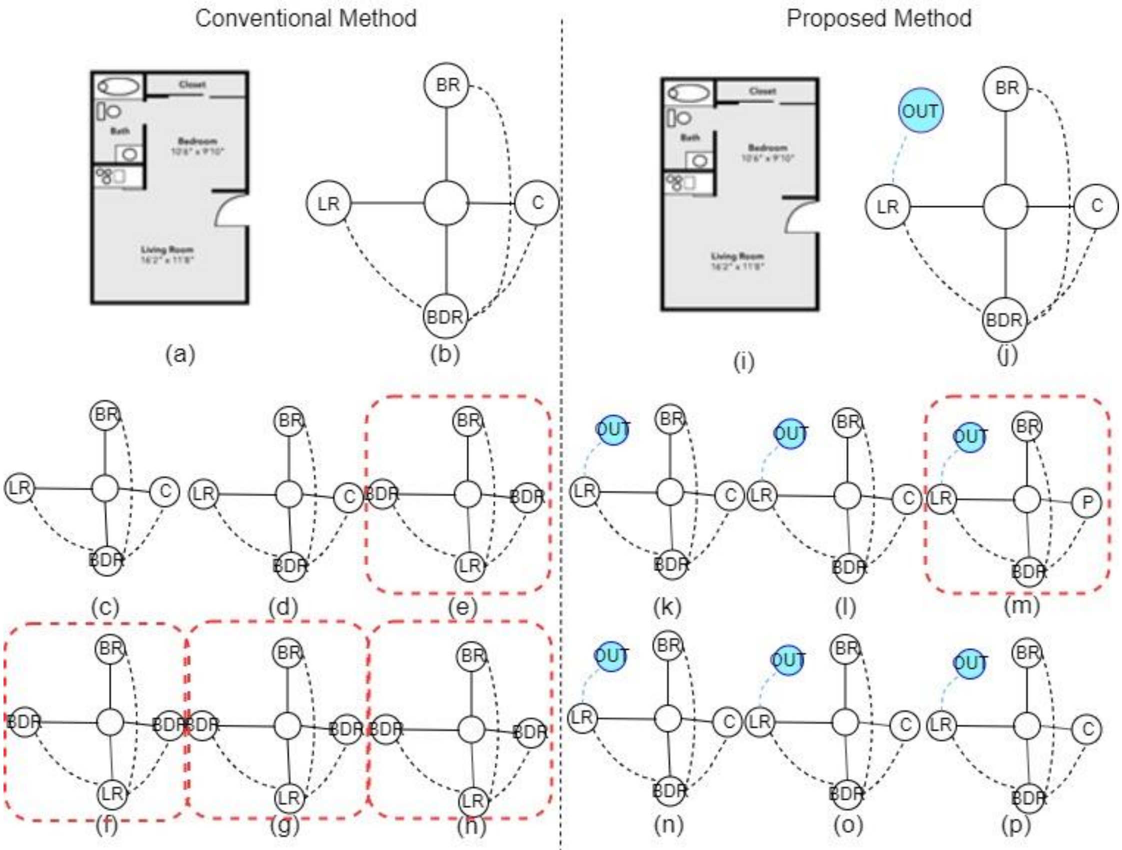


Figure 22: The result of the conventional method (left) and the proposed method (right) using a query floor plan with four rooms in the form of topology graph.

The first row in Figure 21 is the query floor plan with its graph representation. Floor plans (c)-(h) are the result of the conventional method and floor plans (k)-(p) are the result of the proposed method. The circle represents an incorrect result. In addition, the following topology graph is shown in Figure 22.

From Figure 23, the top row is the query floor plan with its graph representation. Floor plans (c)-(h) are the result of the conventional method and floor plans (k)-(p) are the result of the proposed method. The structure of the topology graph comparing both methods is shown in Figure 24. The following results show that when there are five rooms in the floor plan, the accuracy from the floor plan matching of our proposed method is decreasing due to unexpected layouts as shown in the red circles.

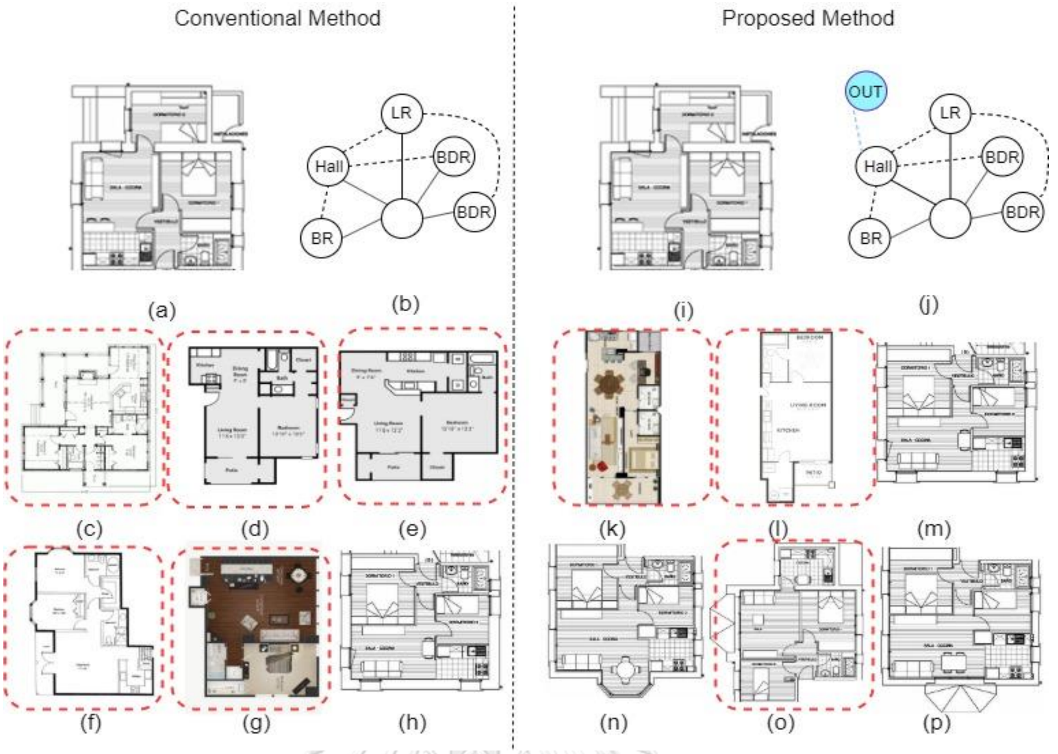


Figure 23: The result of the conventional method (left) and the proposed method (right) using a query floor plan with five rooms.

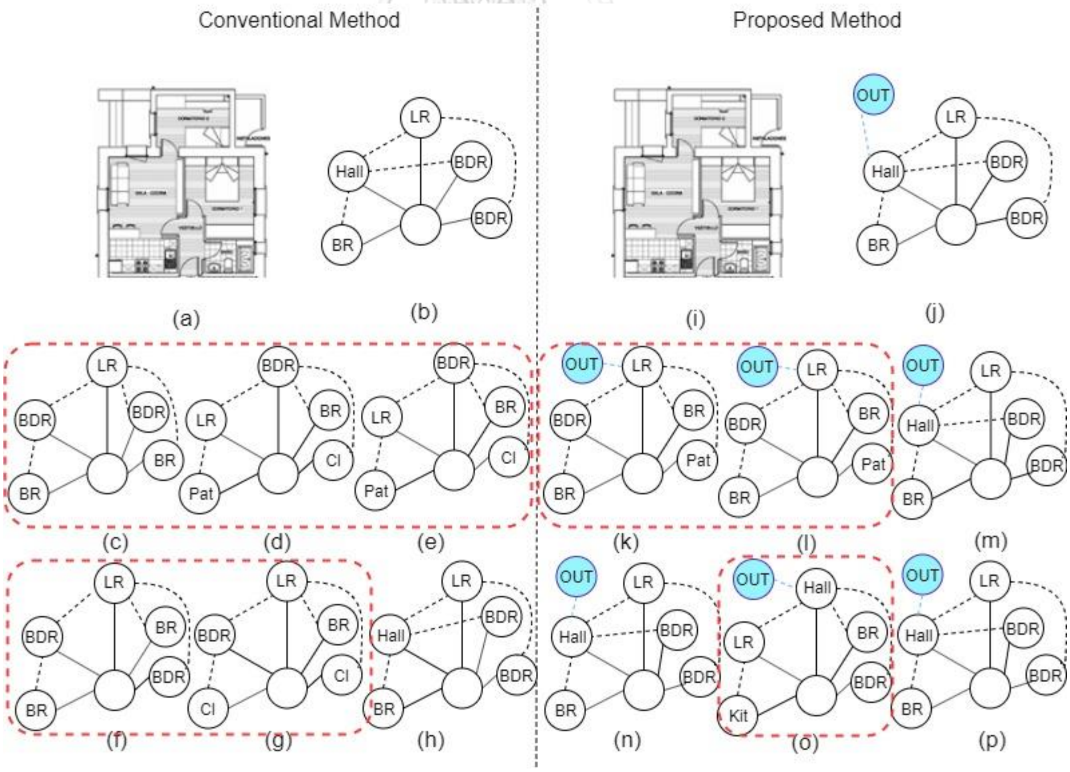


Figure 24: The result of the conventional method (left) and the proposed method (right) using a query floor plan with five rooms in the form of topology graph.

Figure 25: The result of the conventional method (left) and the proposed method (right) using a query floor plan with six rooms.

From Figure 25, the top row is the query floor plan consists of six rooms with its graph representation. Floor plans (c)-(h) are the result of the conventional method and floor plans (k)-(p) are the result of the proposed method. The structure of the topology graph comparing both methods is shown in Figure 26. The following results show that when there are six rooms in the floor plan, the accuracy from the floor plan matching of our proposed method seems to be as same as the conventional method due to unexpected layouts as shown in the red circles.

The results from Figures 27-32 show that our proposed method is not well suited for the floor plans consisted of rooms greater than four. Since most layouts have the same structure of the room connection, but they have many room types arranged for each floor plan.

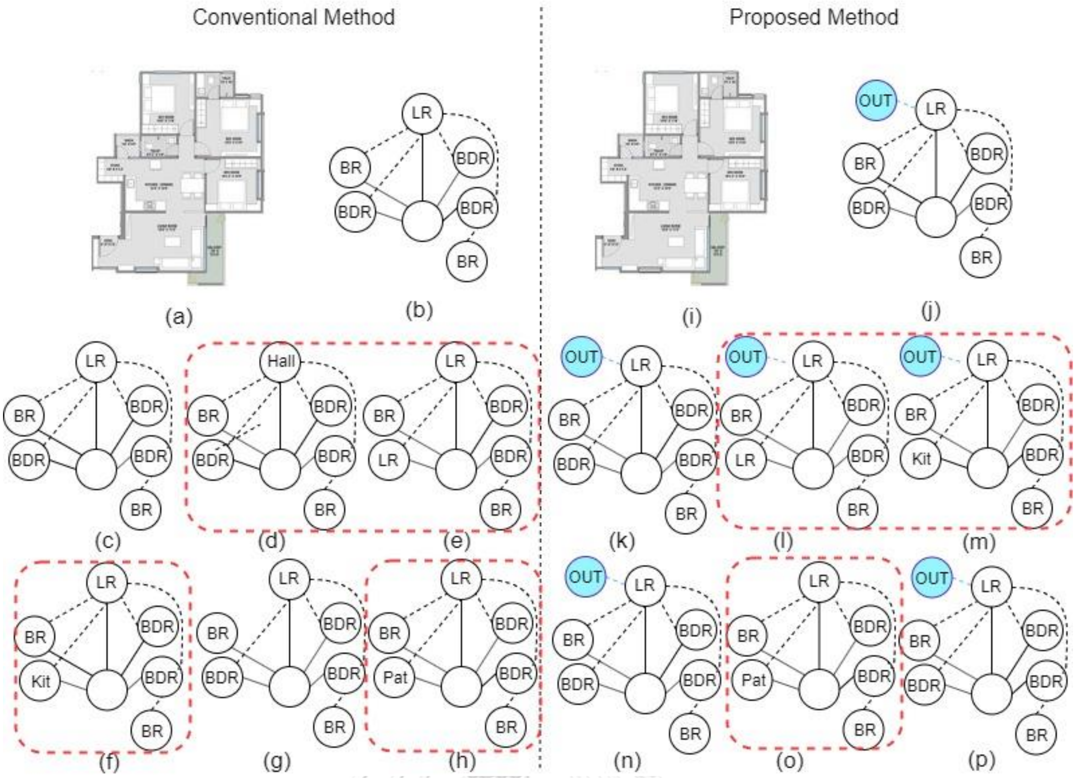


Figure 26: The result of the conventional method (left) and the proposed method (right) using a query floor plan with six rooms in the form of topology graph.

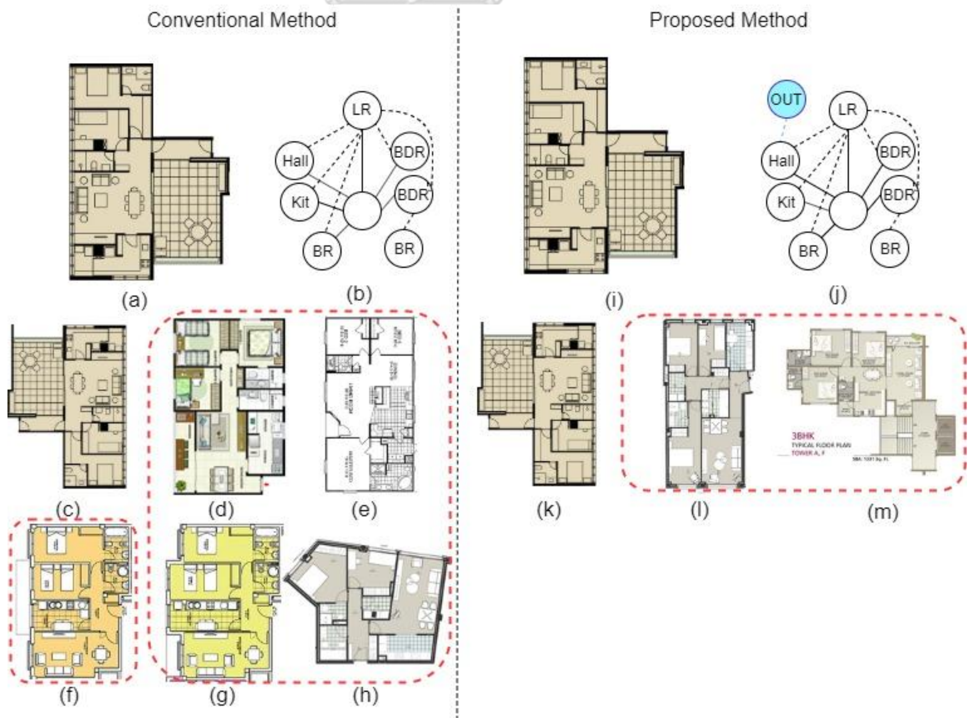


Figure 27: The result of the conventional method (left) and the proposed method (right) using a query floor plan with seven rooms.

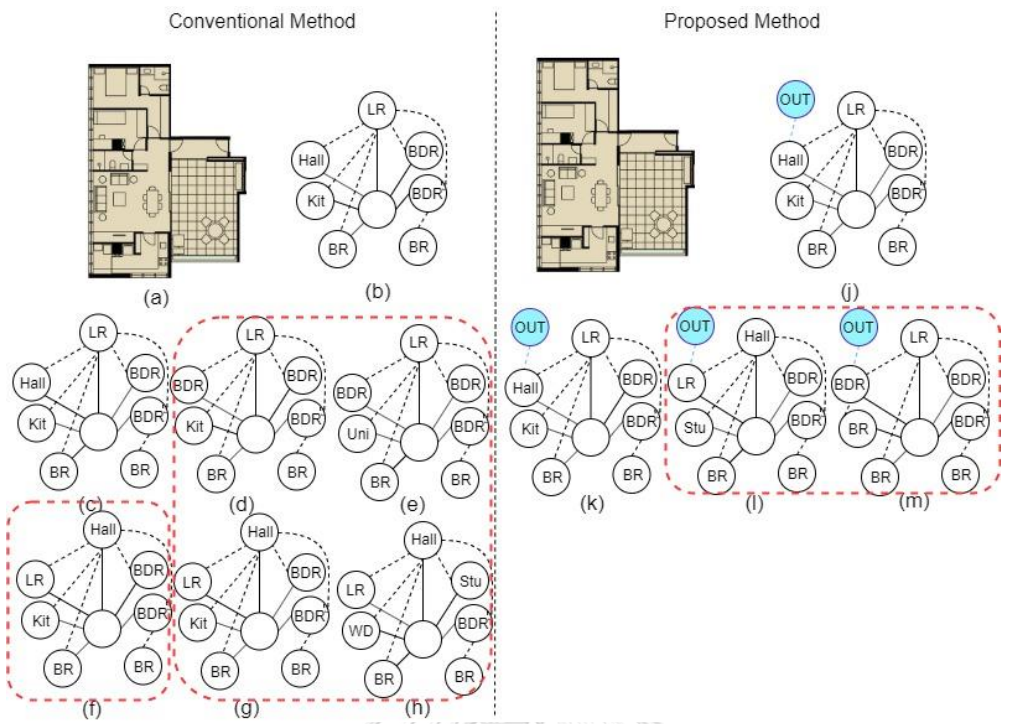


Figure 28: The result of the conventional method and the proposed method using a query floor plan with seven rooms in the form of topology graph.

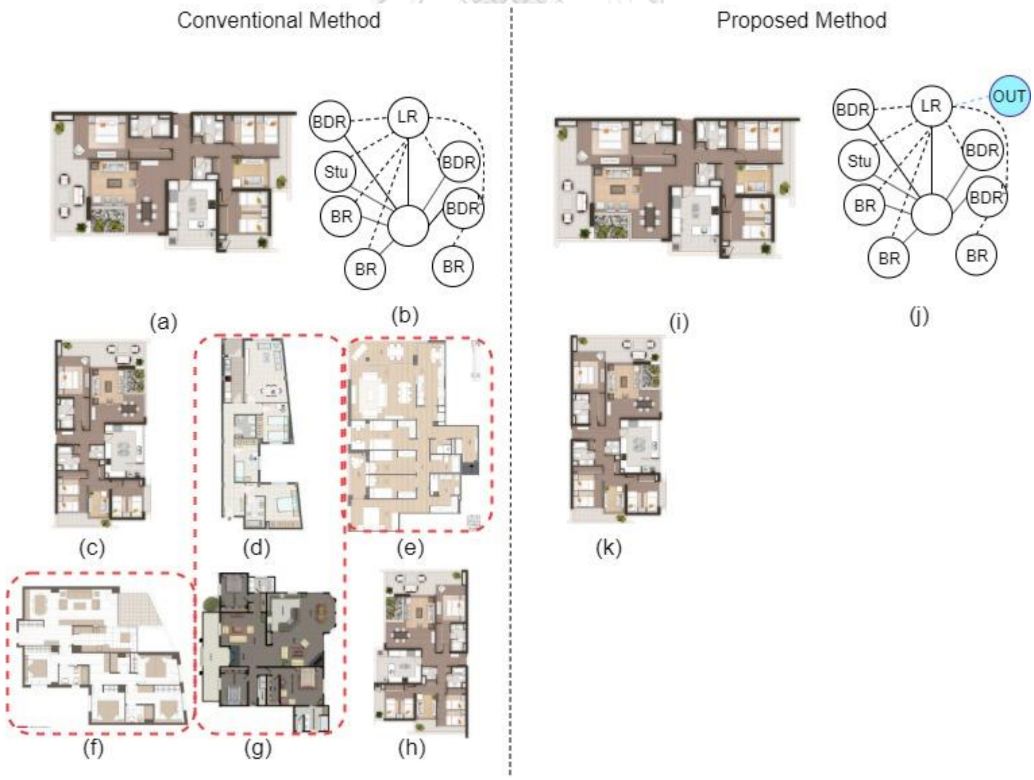


Figure 29: The result of the conventional method (left) and the proposed method (right) using a query floor plan with eight rooms.

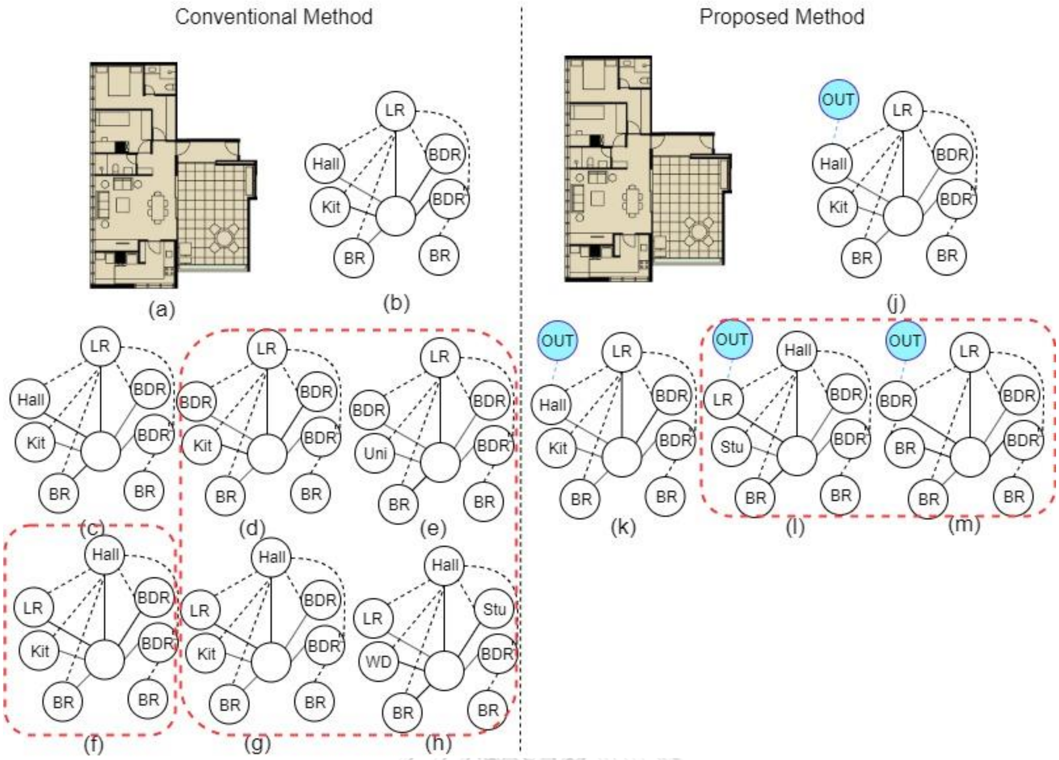


Figure 30: The result of the conventional method and the proposed method using a query floor plan with eight rooms in the form of topology graph.

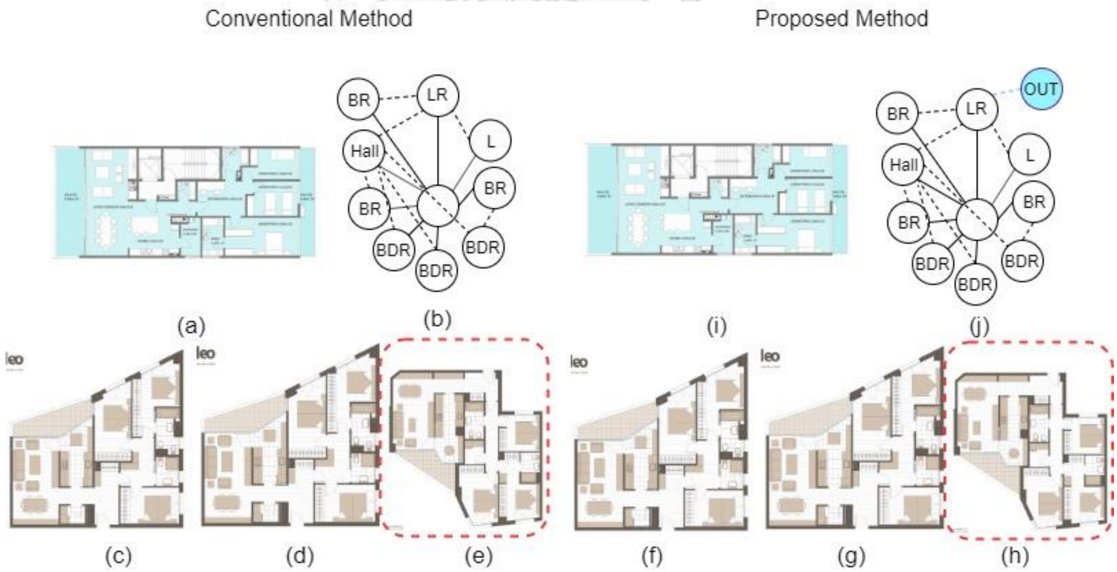


Figure 31: The result of the conventional method (left) and the proposed method (right) using a query floor plan with nine rooms.

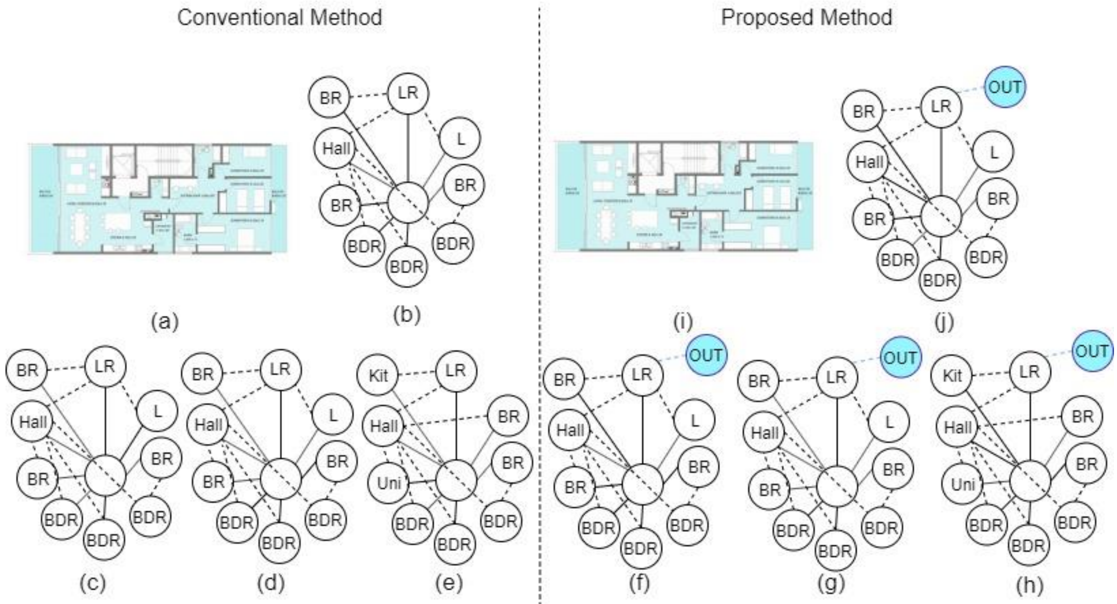


Figure 32: The result of the conventional method and the proposed method using a query floor plan with nine rooms in the form of topology graph.

According to the results from the experiments, we can see that adding a new node in the original topology graph makes the relationship among graph more meaningful. Firstly, the attributes in an original graph are not sufficient to represent each of floor plans. Extending more details in the graph representation process can omit some undesirable retrieval results but it cannot avoid the floor plans we should retrieve. The proposed method can handle with the isomorphism of graph by fixing some nodes, then we obtain the new feature vector because the eigendecomposition is very sensitive if some details in the matrix has changed.

In Figures 33 (a) and (b), we see that adding a new node will extend one more dimension of its adjacency matrix. Then the eigendecomposition may lead us to the new eigenvalues and eigenvectors. In addition, from Figure 34 we see that the graph with two vertices has two different points to add an extra node. Besides, their adjacency matrices after adding the extra node lead us to the same eigendecomposition that means extending the size of an adjacency matrix is not affect the eigenvalues and eigenvectors. So, this is the reason why the efficiency of the floor plan with two rooms in our proposed method equal to the conventional method.

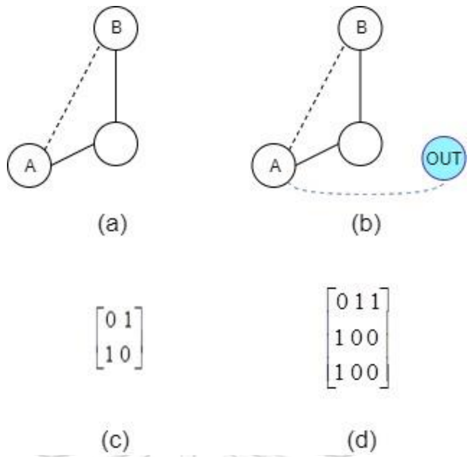


Figure 33: (a) is an original graph, (b) is an added extra node of graph (a), (c) and (d) are their adjacency matrices.

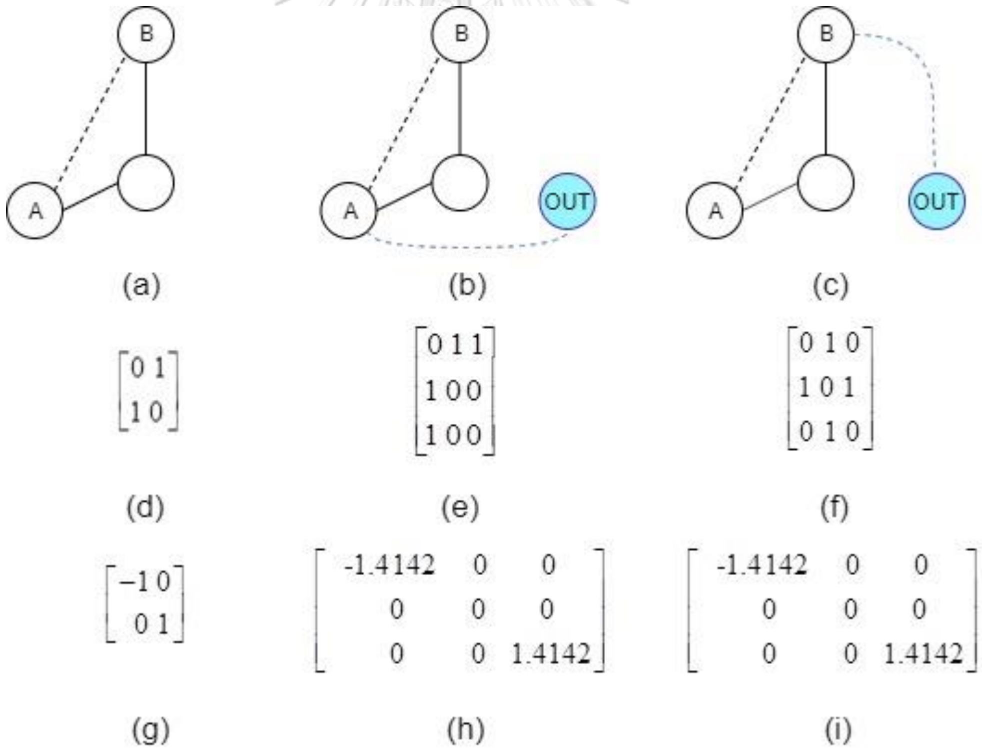


Figure 34: (a) is an original graph, (b) and (c) are an added extra node in different vertex of graph (a), (d)-(f) are their adjacency matrices, and the values in diagonal of matrices (g)-(i) are their eigenvalues.

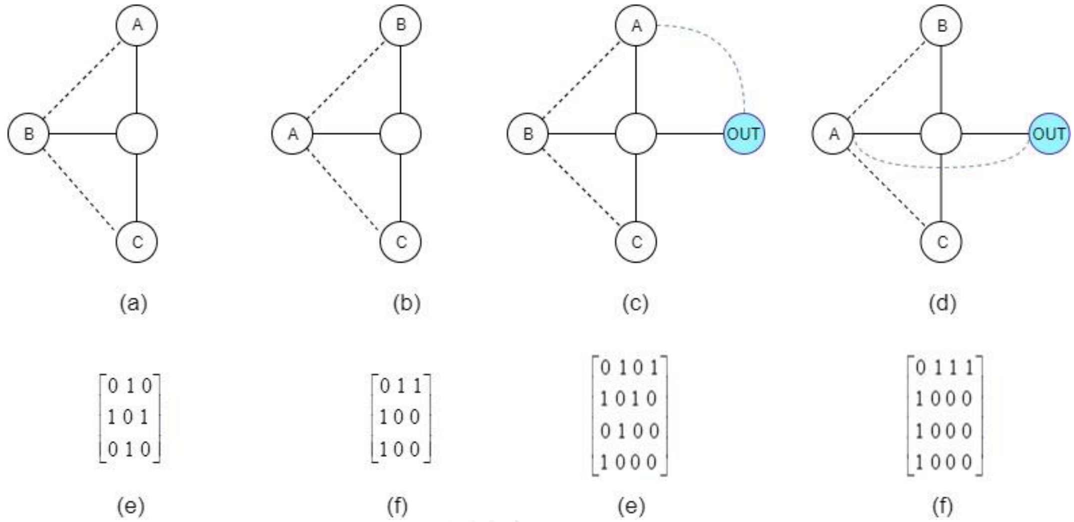


Figure 35: The first row is the topology graphs and the second row is their adjacency matrices.

Furthermore, four graphs in Figures 35 (a)-(d), we see that original graphs in (a) and (b) have the same structure called isomorphism, but they are different in their room type labels. Because node B in (a) is the center connecting between nodes A and C, but node A in (b) is the center connecting between node B and C. However, the eigendecomposition of (a) and (b) are not different. After we add an extra node at node A in (c) and (d), we construct a new non-isomorphic graph. So, the eigendecomposition leads us to the new different eigenvalues and eigenvectors to generate a new feature vector representing each floor plan. Therefore, we can distinguish two graphs having the same structure but different in their node arrangements under conditions that two graphs consist of three rooms with the same vertices set and have the same added extra node. Moreover, the greater number of rooms, the more room type. So, if the layouts consist of more rooms but their structure still isomorphic, then the proposed method cannot distinguish them using the eigendecomposiotion well enough. Hence, efficiency of the proposed method decreases when the number of rooms increases.

CHAPTER V
Conclusions

In summary, we proposed a graph representation for floor plan matching using spectral embedding. A spectral embedding uses less time during the matching process than other conventional methods. Its decomposed an adjacency matrix of a graph to generate a descriptor to represent each floor plan as three component vectors called a feature vector. After we enlarged the size of an adjacency matrix, the feature vector of the proposed method not only determined the structure of each floor plan but also considers the center of rooms. In addition, our proposed method helped in clustering the pattern of a similar floor plan by ranking the order of the distance between a query floor plan and floor plans in a dataset. This method support searching the similar floor plan in more specific details. The customer can search for their query floor plan more correctly.

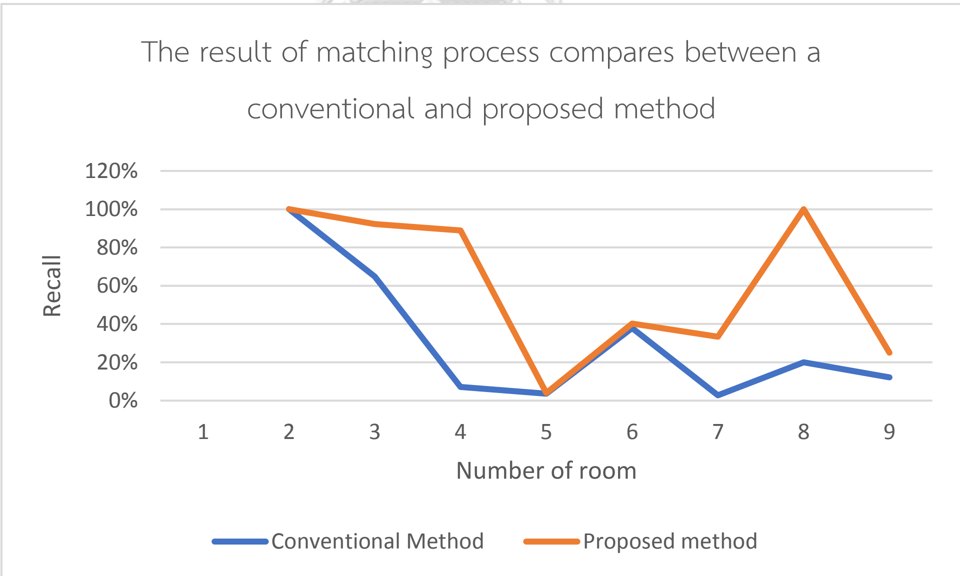


Figure 36: The graph of floor plan matching.

The proposed method is useful for room layout matching. The execution time using to match the query floor plan with floor plans in the dataset is taken only in seconds. The accuracy during the matching process is improved. However, the proposed method cannot handle some variables such as the variety of rooms in each floor plan as shown in Figure 37. The reason of our weakness is that the eigendecomposition cannot extract all important data from the graph such as the label of rooms. So, from Figure 36 indicate us that the increasing number of rooms significantly reduces the accuracy in the matching process. Thus, the query floor plan with the number of rooms not greater than four is suitable for the proposed method, and this became the limitation for our proposed method.

5.1 Future work

In this work, our proposed method concentrated only on dealing with the isomorphism of graph causes it to show undesirable results. Thus, the proposed

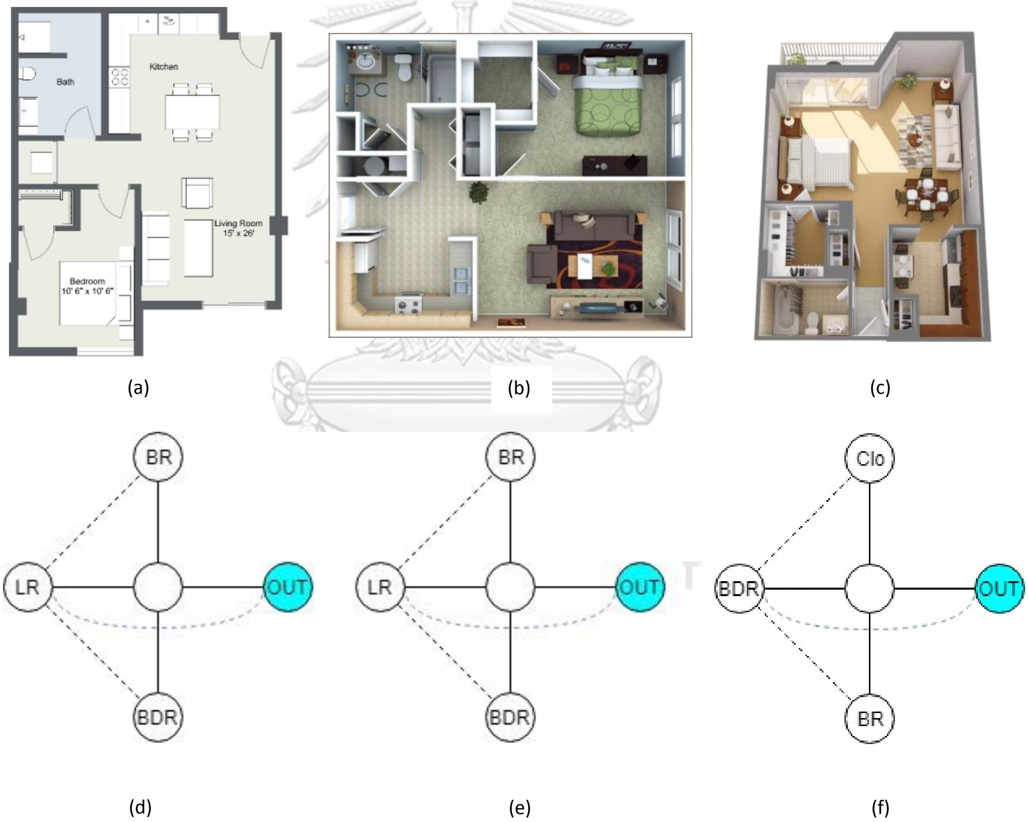


Figure 37: (a) is the query floor plan and (d) is the topology graph of (a). (b) is a correct retrieved result and (e) is the topology graph of (b). (c) is an incorrect retrieved result of our proposed method and (f) is the topology graph of (c).

method is not sufficient for the floor plans that differs from the standard pattern such as the floor plan that includes special rooms. Moreover, when the query floor plan consists of rooms greater than four, it infers us to consider more extra details corresponding to the information and the number of rooms in the floor plan. For further work, if we can find the descriptor considering both the semantic and structure of the room, it can improve the efficiency for floor plan matching base on the idea of graph representation.

REFERENCES

Bunke, H. (2000). *Graph matching: Theoretical foundations, algorithms, and applications.* Paper presented at the Proc. Vision Interface.

Bunke, H., & Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern recognition letters, 19*(3-4), 255-259.

Carletti, V. (2016). *Exact and Inexact Methods for Graph Similarity in Structural Pattern Recognition PhD thesis of Vincenzo Carletti.*

Chung, F. R., & Graham, F. C. (1997). *Spectral graph theory*: American Mathematical Soc.

Dickinson, P. J., Bunke, H., Dadej, A., & Kraetzl, M. (2004). Matching graphs with unique node labels. *Pattern Analysis and Applications, 7*(3), 243-254.

Fischler, M. A., & Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on computers*(1), 67-92.

Hartmanis, J. (1982). Computers and intractability: a guide to the theory of NP-completeness (michael r. garey and david s. johnson). *Siam Review, 24*(1), 90.

Hopcroft, J. E., & Wong, J.-K. (1974). *Linear time algorithm for isomorphism of planar graphs (preliminary report).* Paper presented at the Proceedings of the sixth annual ACM symposium on Theory of computing.

Jiang, X., & Bunke, H. (1999). Optimal quadratic-time isomorphism of ordered graphs. *Pattern Recognition, 32*(7), 1273-1283.

Luks, E. M. (1982). Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of computer and system sciences, 25*(1), 42-65.

Luo, B., Wilson, R. C., & Hancock, E. R. (2003). Spectral embedding of graphs. *Pattern Recognition, 36*(10), 2213-2230.

McKay, B. D. (1981). *Practical graph isomorphism*: Department of Computer Science, Vanderbilt University Tennessee, USA.

Qiu, H., & Hancock, E. R. (2006). Graph matching and clustering using spectral partitions. *Pattern Recognition, 39*(1), 22-34.

Riesen, K., Jiang, X., & Bunke, H. (2010). Exact and inexact graph matching: Methodology and applications. In *Managing and Mining Graph Data* (pp. 217-247): Springer.

Sharma, D., Chattopadhyay, C., & Harit, G. (2016). *A unified framework for semantic matching of architectural floorplans.* Paper presented at the 2016 23rd International Conference on Pattern Recognition (ICPR).

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Spielman, D. A. (2018). Testing for Graph Isomorphism Using Eigenvalues.

Weber, M., Liwicki, M., & Dengel, A. (2011). *Indexing with well-founded total order for faster subgraph isomorphism detection.* Paper presented at the International Workshop on Graph-Based Representations in Pattern Recognition.

Ziff, R. M., Finch, S. R., & Adamchik, V. S. (1997). Universality of finite-size corrections to the number of critical percolation clusters. *Physical review letters, 79*(18), 3447.

APPENDIX

Table 3: The result of matching process compares between a conventional and proposed method

Number of rooms		Conventional Method (Sharma, 2016)	Proposed method
2 (51)	Retrieved	50	50
	Correct	50	50
	Recall	100%	100%
3 (189)	Retrieved	182	128
	Correct	118	118
	Recall	65%	92%
4 (159)	Retrieved	113	9
	Correct	8	8
	Recall	7%	89%
5 (147)	Retrieved	81	75
	Correct	3	3
	Recall	4%	4%
6 (121)	Retrieved	66	62
	Correct	25	25
	Recall	38%	40%
7 (65)	Retrieved	36	3
	Correct	1	1
	Recall	3%	33%
8 (43)	Retrieved	10	1
	Correct	2	1
	Recall	20%	100%
9 (25)	Retrieved	3	3
	Correct	2	2
	Recall	67%	67%

From Table 3, The recall from our experiment for floor plans with two to nine rooms of the conventional method is 37.87%, and the recall of our proposed method is 65.67%. Therefore, our proposed method improves the matched accuracy from the conventional method by 27.81%.

VITA

NAME	Thamonwan Sa-ngawong
DATE OF BIRTH	23 May 1994
PLACE OF BIRTH	Chachoengsao, Thailand
INSTITUTIONS ATTENDED	Chulalongkorn University
HOME ADDRESS	50/7 Moo 7, Paknam, Bangkla Distict, Chachoengsao, 24110