3

Topological features for representation and retrieval of similar floor plans

As observed by the detailed literature survey given in Chapter 2, researchers in the area of analysis of architectural floor plans have mainly explored areas like:

- *Symbol spotting*: Symbols are described as a set of logically related graphical parts. These parts are generally visual primitives such as vectors, lines, and arcs or a certain combination of all of these. These primitives in a certain context denote a symbol instance. In the task of symbol spotting thus, all the decor symbols present inside a floor plan are localized and recognized.

- *Floor plan Analysis*: This pertains to structural and semantic analysis of floor plans. Examples of various categories of problems addressed in this type of task are: detecting the various type of walls in a floor plan, segmentation of various rooms in a floor plan, understanding the relationship between the rooms, identifying various functionalities of various parts of a floor plan.

From the literature it can be inferred that symbol spotting and analysis in architectural floor plans have been active research areas in the past. However, a composite framework comprising of analysing a floor plan structurally as well as semantically, and further retrieving floor plans based on certain content or categories is a relatively novel problem. Upon closer observation it was found that a floor plan has various types of meaningful information (content) that can be explored while retrieval. At the same time, inter-class dissimilarity and intra-class similarity among the samples of a floor plan dataset are also to be taken into consideration. Some examples for a meaningful content are, layout information, inner components or decor information, accessibility or approachability of each room from the entry or exit or from each other, aesthetics information or proper placement of the rooms inside the layout, and so on.

In this chapter, the primary focus is on the first two types of content namely, the layout information and the decor information to distinguish between a pair of floor plans. A baseline global framework comprising of match scores computed based on both the topological and decor features is proposed, which paves way for a novel framework for the introduction of retrieval in the area of architectural floor plans. In the subsequent chapters of this thesis, improvement over this baseline is discussed. The Chapter organization is as follows: Sec. 3.1 gives a brief insight into the proposed approach. Subsequent sections detail out the methodology of the proposed approach. In Sec. 3.6 the experimental setup, qualitative and quantitative results of the proposed technique are presented. Section 3.7 provides an analysis of the results obtained and finally Sec. 3.8 concludes the Chapter.

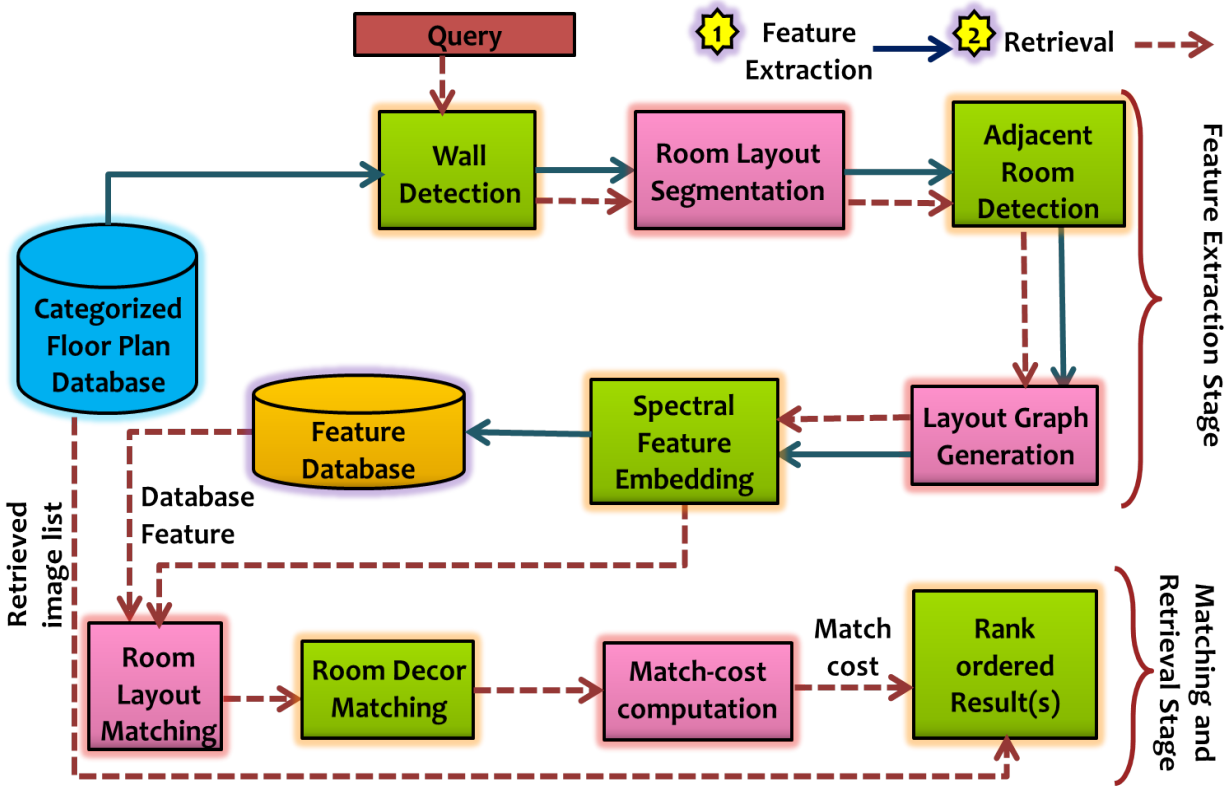


Figure 3.1. : Framework diagram for our structural retrieval based on topology

3.1 BRIEF OVERVIEW

A query by example paradigm has been followed to retrieve similar floor plan images. Figure 3.1 depicts the complete framework of the proposed technique of semantic based architectural floor plan retrieval. The framework has two phases : A) Feature extraction, and B) Matching and Retrieval phase. The solid arrows in the framework diagram (Fig. 3.1) correspond to the feature extraction phase, whereas, the dashed arrows correspond to the retrieval phase. The various components of the framework, in order of operation, are:

1. Floor plan repository, which in this case are the architectural floor plan databases SESYD [Delalandre *et al.*, 2010a] and ROBIN [Sharma *et al.*, 2017].

2. The Wall Detection block, which focuses on delineating a floor plan and detecting room boundaries.

3. Room Layout Segmentation block, which segments the layout into meaningful rooms discarding all the inner decor components present inside the rooms.

4. Adjacent Room Detection (ARD) block wherein the relationship between the segmented rooms is established. Here, the relationship pertains to the topological information of the layout.

5. The output of the ARD block is sent to the Layout Graph generation step where, the topological information of a layout is modelled into a graph for feature representation.

Considering the adjacencies a graph of the floor plan is generated using the topology graph idea proposed in [da Fonseca, 2004].

6. The next block is the Spectral Feature Embedding block where, the topology (neighbourhood) information is embedded into a pattern space as a feature vector. The adjacency matrix obtained from this layout graph is used to extract the spectral features.

7. The next component, the feature database, stores all the feature vectors extracted from the floor plans and serves as a feature repository. This procedure is carried out apriori for all the images present in the floor plan database.

The blocks mentioned above constitute the Feature Extraction Stage. As soon as the query is fired, first the features are extracted from the floor plans using the Stage 1 blocks and then the Matching and Retrieval Stage (Stage 2) comes into picture. The main blocks in Stage 2 are:

1. Room Layout Matching (RLM) block, where, the dissimilarity between the query feature vector and each of the model feature vectors is computed using Euclidean distance metric. The model corresponding to the minimum distance of its feature vector from the query feature vector is chosen as the best match.

2. Room Decor Matching (RDM) block, where the layout from the database to which the query layout matches the best is then chosen and further in the images of that matching layout category, the arrangement of furniture is analyzed. A match cost is computed by matching the room decor of each image of the layout category with the query image and this match cost is then used to rank order the results for obtaining the best matching floor plan image to the query floor plan image.

3. Match-cost Computation block helps in calculating the cumulative match cost obtained from both the Room Layout Matching block and the Room Decor Matching block.

4. Finally, using this cumulative match cost floor plans are fetched from the repository and rank-ordered on the basis of their similarity with the query floor plan.

The main contributions towards the first attempt for designing a baseline framework for floor plan retrieval are as follows:

1. A novel approach to perform floor plan retrieval incorporating both symbol spotting and room level analysis.

2. Performing two level matching for floor plans based on semantics such as layout and decor.

3. Graph spectral embedding of layouts into a pattern space for efficient and fast matching.

All these steps finally complete the retrieval framework for similar floor plans. The details of each step mentioned in the framework diagram are further discussed in the following sections.

3.2 LAYOUT SEGMENTATION

The first step in the proposed framework is layout image segmentation, which consists of structural and semantic analysis to segment the floor plans into respective rooms. Figure 3.2 depicts the various stages of layout segmentation and analysis. Boundary extraction is performed where

Figure 3.2. : Various stages of structural and semantic analysis on an image from SESYD dataset [Delalandre *et al.*, 2010a]: (a) unsegmented layout, results of (b) morphological operation and wall, boundary extraction, (c) Detection of doors and windows (d) Closing gaps in walls and (e) labelling and segmenting the room structure

the boundary extraction algorithm and morphological operations are carried out detecting the walls in the layout. Let the input floor image be denoted by $\mathscr{I}$. This floor plan image is first dilated ($\oplus$) by a disk type structuring element (S_a) of unit radius followed by erosion operation ($\ominus$) by the same element, according to the following equation:

$$\mathscr{I} = \mathscr{I} \oplus S_a \;\; ; \;\; \mathscr{I} = \mathscr{I} \ominus S_a \tag{3.1}$$

These operations lead to segregation of the layout boundary from small unnecessary designs present inside the layout (Fig. 3.2(b)). This detects the walls in the layout to represent the room structures and thus, delineates the rooms within the layout from each other. Next gap closing is performed at the location of doors and window in the floor plan. Floor plans contain doors and windows that connect each room, therefore, are important for semantic analysis. However, the presence of these doors and windows doesn't influence the overall structure of the building. Hence, these two components are removed for structural analysis. The doors and windows inside the layout are present in various orientations and their elimination leads to production of gaps in the walls at their location (refer Fig. 3.2 (c)). The first task therefore, is to detect the doors and windows in the floor plan and close the gaps at their locations so as to construct definite boundaries for the rooms (Fig. 3.2(d)). To detect the doors and windows Harris corner detector [Harris and Stephens, 1988] is employed.

It is observed that the door symbol has an arc like structure (triangular structure) and thus, Harris corner detector detects three corners at the three triangle edges for the door symbol (refer Fig. 3.3(a)). It is to be noted that only the door symbol out of all the decor symbols in the symbol library has this type of a unique triangle-bound 3-corner structure. Through this unique

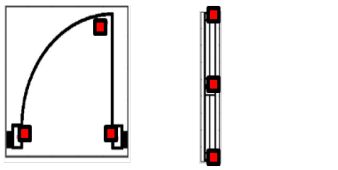


Figure 3.3. : Detecting doors and windows using Harris corner detector

structure the doors stand out as compared to the rest of the symbols in the symbol library. To detect whether the obtained corners form a triangle or not, the three corners identified through Harris corner detector for the door symbol are joined with their adjacent corner point, creating a closed triangle. Further, length of each side (a,b,c) of this triangle is computed and using the triangular inequality : "sum of any two sides should be greater than the third side ($a+b>c$, $a+c>b$, $b+c>a$)", the closed structure is determined to be a triangle. Thus, using this approach doors are detected. The next symbol that needs to be detected in a floor plan for gap closing, is the window. The windows are observed to have collinear corners (refer Fig. 3.3(b)) and thus, get detected through the detection algorithm as the only symbol with collinear corners. Again to identify whether the obtained corners are collinear or not, the co-ordinates of the corner points are noted (x_1, y_1, x_2, y_2 and x_3, y_3). Further, slope of the line between each pair of points is calculated as:

$$slope = \frac{y_1 - y_2}{x_1 - x_2}; \qquad \frac{y_2 - y_3}{x_2 - x_3}; \qquad \frac{y_3 - y_1}{x_3 - x_1} \tag{3.2}$$

If the slope is the same between any pair of corner points, then they are identified to be collinear, hence belonging to the window structure. Thus, uptil now the door and the window locations have been identified.

Upon identifying and eliminating the doors and windows from the floor plan, the next task is to close the gaps in the walls (as shown in Fig. 3.2 (c)). Closing of the gaps is achieved by joining the detected co-ordinates of the doors (in this case the co-ordinates forming 0° angle with each other) and windows setting the values of the pixel location between them as 0, with a thickness of about 40 pixels. Hence, creating a closed wall-structure as in Fig. 3.2 (d). Next, connected component analysis using 8-neighbourhood criterion is performed, to label each room in the layout separately. By this step, the rooms are segmented and are surrounded with walls separating each one of them 3.2(e)). The segmented room layout is sent to the Adjacent Room Detection block for adjacency graph creation.

3.3 ADJACENT ROOM DETECTION

In this step the segmented image is passed through an Adjacent Room Detection (ARD) module to determine the adjacencies between the rooms in the layout. As the segmented rooms are still divided by walls surrounding them, a threshold corresponding to the wall width of about 40 pixels is taken along both vertical and horizontal directions to detect the differently labelled pixels in the image corresponding to the adjacent room. By determining the adjacencies the topology graph

for the layout is created, according to the technique mentioned in [da Fonseca, 2004]. Figure 3.4(c) depicts an example of a topology graph created from the room adjacencies (Fig. 3.4(b)) determined from a given floor plan (Fig. 3.4(a)). The parent node (the one at the centre) represents the layout and the child nodes represent the rooms in the layout. The solid edges between the parent node and the child nodes represent inclusion and dashed edges between the child nodes represents adjacencies. This graph creation further, aids in constructing an adjacency matrix for the rooms detected in the layout which is then used for feature extraction.

However, topology graphs are not directly used for searching similar drawings, since graph matching is a NP-complete problem. Instead of that, the corresponding graph spectrum is used. For each topology graph to be indexed in a database, the descriptors are computed on the basis of its spectrum. In this way, the problem of isomorphism between topology graphs is reduced to computing distances between descriptors. The next section details out the steps to identify the decor components inside the floor plan which are also used for matching purposes.

3.4 FURNITURE DETECTION AND CATEGORIZATION

Algorithm 1 depicts the steps involved in furniture categorization. Firstly, the walls are detected in the floor plan image using a morphological operation as walls are thicker than the other objects inside a floor plan. Then both, the original floor plan image and the image containing only walls are subtracted to obtain only the furniture present inside the floor plan. The decor present inside the whole floor plan is counted on the basis of the application of morphological fill operation applied to the image. Blobs obtained are taken as the decor components inside the floor plan. Further, to categorize the obtained blobs, area-ratio of the individual connected components inside the blob is taken and a unique signature for each furniture component is established by considering the area-ratio of three largest components in each blob. Prior to this, such an area-ratio feature taking the three largest connected components in a furniture symbol is obtained for all the 12 symbols in the symbol library and stored. Signature obtained of all the components inside a floor plan image is finally compared with the signature of the components present in the symbol library and thus, the type of the furniture is assigned to each decor item in the floor plan. Figure 3.5 shows how a furniture in the furniture set is assigned a signature according to Algo. 1.

(a) Architectural Floor plan (b) Results of proposed room layout segmentation technique (c) Layout graph of (a)

Figure 3.4. : Topological graph for the segmented room layout

Algorithm 1 Furniture detection and categorization

Input: Room image without outer walls ($\mathscr{I}_w$), Furniture Template Signatures (F)
Output: Furniture Count (C), Furniture Type Set ($\mathscr{T}$)

C=0, $\mathscr{T}$={}
$\mathscr{B}$=Morphological fill operation($\mathscr{I}_w$)
$C = |\mathscr{B}|$ ▷ $|\cdot|$: *Cardinality*
for $j = 1$ to C **do**
  $C = CC(\mathscr{B}_j)$ ▷ CC()=Connected Component
  for $k = 1$ to $|C|$ **do**
    $\mathscr{A}_k = Area(c_k)$, where $c_k \in C$
  end for
  $A = Sort_{desc}(\mathscr{A})$
  $S(\mathscr{B}_j) = \{(A_1/A_3), (A_2/A_3), 1)\}$ ▷ $S(\cdot)$: Signature
  for $k = 1$ to $|F|$ **do**
    if $S(\mathscr{B}_j) == F_k$ **then**
      $T(\mathscr{B}_j) = T(F_k)$ ▷ $T(\cdot)$: Type of Furniture
    end if
  end for
  $\mathscr{T} = \{\mathscr{T} \cup (T(\mathscr{B}_j)\}$
end for

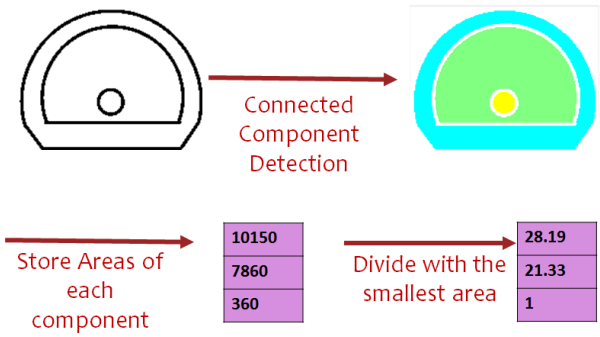
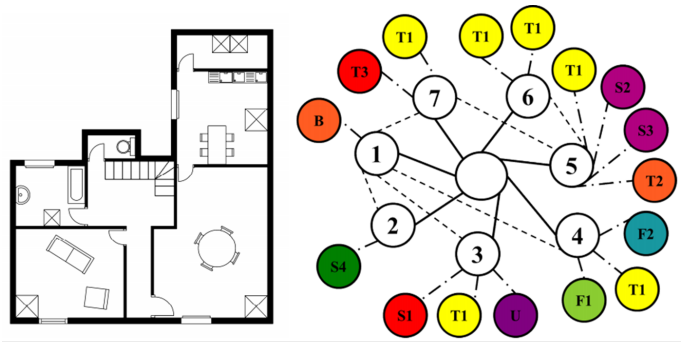


Figure 3.5. : Example showing how a sink symbol in the furniture set is assigned a unique signature.

Figure 3.6(b) depicts an example of the semantic layout graph, consisting of both room adjacencies as well as, room decor representation, corresponding to a layout shown in Fig. 3.6(a) from the SESYD [Delalandre *et al.*, 2010a] dataset. The uncoloured nodes labelled with numbers correspond to the number of rooms present in the layout. The adjacencies between these rooms are depicted by dashed edges. The coloured nodes in the graph correspond to the furniture inside each room. A 360° colour wheel is quantized into uniform bins of value 30° each. According to the orientation of the furniture inside the rooms, a corresponding colour value from the colour wheel is assigned to the furnitures. After obtaining the graph depicting the adjacencies between the rooms and also the decor components, the next task is graph spectral embedding that helps in mapping all graphs to pattern space for retrieval purposes.



(a) Architectural Floor plan (b) Semantic graph of (a)

Figure 3.6. : Room layout and corresponding graph consisting of both room adjacencies as well as room decor representation

3.5 GRAPH SPECTRAL EMBEDDING

Graph Spectral Embedding [Luo *et al.*, 2003] approach is adopted to represent the room layout graphs created in the previous stage in a pattern-space. Such a spectral embedding helps to map similar floor plans closer in the pattern space and thus while retrieval all similar plans are clustered together for efficient matching. Using this technique one can map graphs containing a number of nodes to a feature vector of uniform size for all the floor plans in the database. To overcome the problem of how to map the structure of a graph onto a vector of fixed length, the graph-spectral decomposition methods are explored. Leading Eigenvectors of the graph adjacency matrix are used to define Eigenmodes of the adjacency matrix. Thereafter, the spectral decomposition of the adjacency matrix is performed. The use of graph spectrum as an indexing method is beneficial as (1) it captures local topology, (2) it is invariant to subgraph re-order, and (3) it is stable, since small changes in the graph produce little changes in its spectrum. The graph obtained in the previous step is further represented as an adjacency matrix for spectral decomposition as shown in the next subsection.

3.5.1 Spectral Feature representation

Given N images in the database, their graphs are represented as $G_1, G_2 ..., G_N$. The k^{th} graph is denoted as $G_k = (V_k, E_k)$, where V_k is the set of vertices and $E_k \in V_k \times V_k$ is the edge-set. For each graph G_k, an adjacency matrix A_k of size $|V_k| \times |V_k|$ is computed, where $|\cdot|$ denotes the size of the set.

From the calculated adjacency matrices $A_k, k = 1 \ldots N$, where, N is the total number of graphs for the images in the database, the Eigenvalues λ_k are computed by solving the equation

$$|A_k - \lambda_k I| = 0 \tag{3.3}$$

and the associated Eigenvectors $\vec{\phi}_k$ by solving the system of equations,

$$A_k \phi_k^i = \lambda_k^i \phi_k^i \tag{3.4}$$

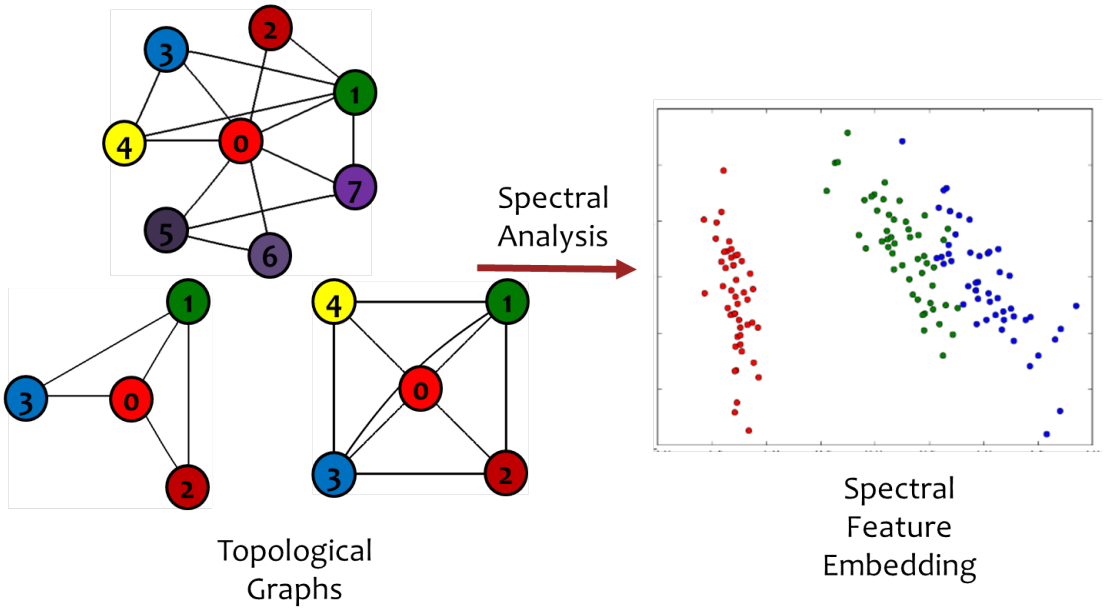


Figure 3.7. : Embedding features from the topological graphs into spectral space. Adjacency matrices with varying sizes are reduced to fixed length vectors in the pattern space. Kindly note, in the right hand side, similar vectors are clustered closely in the feature space.

, where, i is the Eigenmode index.

Spectral feature vector representing the spectrum of the graph G_k is constructed from the top n Eigenvalues of A_k taken in decreasing order. For the k^{th} graph, this vector is represented as :

$$\vec{F}_k = (\lambda_k^1, \lambda_k^2 .... \lambda_k^n)^T. \tag{3.5}$$

Figure 3.8 (a) represents a floor plan image and its corresponding segmentation into rooms. Figure 3.8 (b) represents the topological graph for this segmented layout depicting adjacencies and relationships between the rooms. Node colours correspond to the rooms in the floor plan. Figure 3.8 (c) is an adjacency matrix representation of this topological graph. This adjacency matrix is spectrally analysed and broken down into it's leading Eigen values arranged in a decreasing order as shown in Fig. 3.8 (d). This further forms the feature vector for each layout.

3.5.2 Spectral feature embedding

Once a feature vector is obtained as mentioned in Sec. 3.5.1 the next task is to analyse this feature vector and reduce it into a uniform size vector representing the layout information from each floor plan (refer Fig. 3.7, where due to feature extraction all graphs have been projected on a space of uniform dimension). Uniformity in the feature size is needed as floor plans in the dataset have different number of rooms leading to different sizes of adjacency matrices. Hence, to compare all the adjacency matrices for similarity there was a need to reduce them to a same size feature vector. To achieve this, principal components analysis is performed, following the parametric Eigenspace idea of [Murase and Nayar, 1994]. Rationale behind using this approach is to organise graphs into a pattern-space in which similar structures are close to one-another, and dissimilar structures are far apart. The graphs extracted from each image are vectorised in the way outlined in the Sec. 3.5.1. Principal components analysis (PCA) of the feature vector $\vec{F}_k$ is performed next. For the N

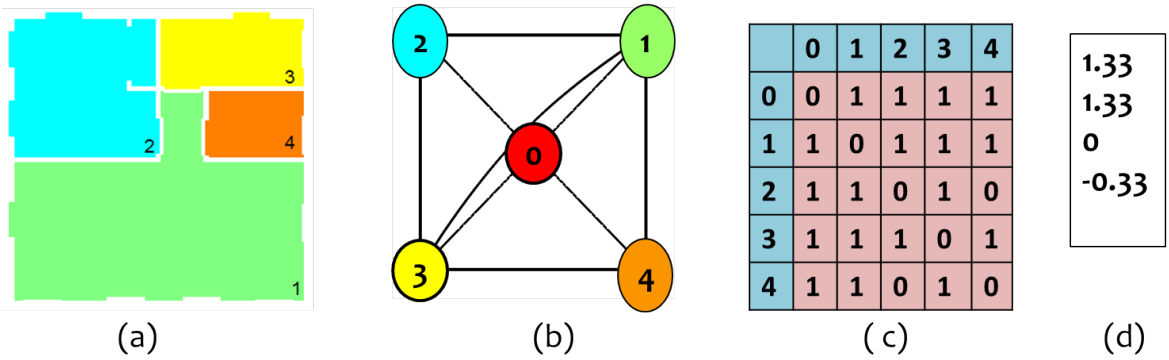


Figure 3.8. : Topological feature extraction from a room layout in a floor plan

different graph representations of the layouts in the database their spectra are arranged in a matrix as:

$$R = [\vec{F}_1, \vec{F}_2, \ldots, \vec{F}_N]. \tag{3.6}$$

The Covariance matrix, Cv for R is computed as,

$$Cv = RR^T. \tag{3.7}$$

A spectral decomposition of Cv results in Eigenvalues ρ upon solving the Eigenvalue equation

$$|Cv - \rho I| = 0 \tag{3.8}$$

and the corresponding Eigenvectors $\vec{\psi}$ are found by solving the Eigenvector equation.

$$Cv\vec{\psi}_i = \rho_i \vec{\psi}_i. \tag{3.9}$$

The Principal components directions are obtained by using the first three leading Eigenvectors of C. The three leading Eigenvectors give a wholesome estimation of the adjacencies in the layout [Luo *et al.*, 2003]. Three orthogonal vectors span the co-ordinate system of the Eigenspace as $\Phi = (\vec{\psi}_1, \vec{\psi}_2, \vec{\psi}_3)$. This aids in projecting the individual graphs represented by the vectors $\vec{F}_k; k = 1, 2, \ldots, N$ on the pattern space as

$$\vec{x}_k = \Phi^T \vec{F}_k. \tag{3.10}$$

Hence, each graph obtained from the layouts is represented as a three-component vector $\vec{x}_k$ in the Eigenspace as $\vec{x}_k = (x'_k, x''_k, x'''_k)^T$. After the computation of the features, in the next section, the matching technique is described.

3.5.3 Feature Matching and Retrieval

A two phase matching and retrieval, Room Layout Matching (RLM) and Room Decor Matching (RDM) is proposed. Feature matching and retrieval of similar layouts (RLM) is performed by determining the proximities between all the layouts. The distance (d) 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:

$$d = \sqrt{(x'_q - x'_m)^2 + (x''_q - x''_m)^2 + (x'''_q - x'''_m)^2} \tag{3.11}$$

.

The smaller the distance more similar the graphs are. Thus, this similarity measure helps us to order the architectural layouts represented as graphs. Rank ordering of the images helps in generating the most similar images to the query input and can efficiently retrieve exactly matched or approximately similar layouts from the database to suit the application.

After the first level matching of the layouts the next level match to find the semantic difference between the layouts (RDM) is done. An algorithm to compare the number and type of the objects inside the room (see Alg.2) is proposed, where $\mathscr{T}$ represents type of furniture, $\mathscr{C}^u$ represents number of unique furniture in the whole floor plan. After determining the closest layout, the unique furnitures present in each room are identified. To compute the semantic difference, between a pair of layouts, the number and type of the objects present inside the rooms of the query layout is compared with the model layout. Further a matching cost ($cost_{qm}$) is assigned which increments if there is an object level matching between the layouts and decrements otherwise. The final $cost_{qm}$ obtained from Alg.2 is used to rank order retrieved results based on semantic difference. A flowchart for Room Decor Matching is also provided in Fig. 3.9 to depict the various steps involved in the RDM framework.

Algorithm 2 Calculate semantic difference between layouts

Input: Query Image ($\mathscr{I}_q$), Model Image ($\mathscr{I}_m$)
Output: Matching Cost ($cost_{qm}$)

$\mathscr{T}_q = \{\mathscr{T}_q^i\}; 1 \leq i \leq |\mathscr{C}_q^u|$ ▷ $\mathscr{C}^u$: No. of unique items
$\mathscr{T}_m = \{\mathscr{T}_m^j\}; 1 \leq j \leq |\mathscr{C}_m^u|$ ▷ $\mathscr{T}$: type of furniture
$cost_{qm} = 0$
if $(|\mathscr{C}_q| \neq |\mathscr{C}_m|)$ **then** ▷ $\mathscr{C}$: No. of items
    $cost_{qm} = -1$
else
    for all i, j **do**
        if $\mathscr{T}_q^i = \mathscr{T}_m^j$ **then**
            $cost_{qm} \leftarrow cost_{qm} + 1$
        else
            $cost_{qm} \leftarrow cost_{qm} - 1$
        end if
    end for
end if
    return $cost_{qm}$

3.6 EXPERIMENTS AND RESULTS

The experiments are performed on the SESYD dataset [Delalandre *et al.*, 2010a] and the ROBIN dataset [Sharma *et al.*, 2017]. The SESYD dataset has 10 different types of layouts, each with 100 images varying in their arrangement of furnitures inside the rooms. The image sizes in the database vary from 6775×2858 to 2056×1837. On the other hand, the ROBIN dataset has about 510 floor plans each divided into 3 broad categories based on the number of rooms present inside them and varying in the global structure. All floor plans are binarized to ensure that only structural information of the floor plans is used for the analysis.

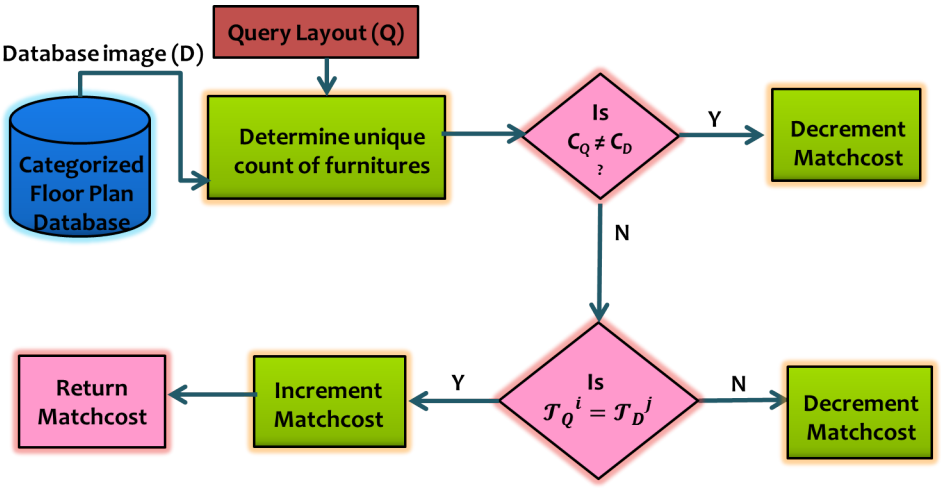


Figure 3.9. : Framework for Room Decor Matching

Figure 3.10. : Top 5 retrieved results corresponding to a query from the SESYD dataset.

3.6.1 Qualitative Results

The framework starts off with segmenting the architectural layouts into rooms followed by, detecting the various furnitures present in each room. A semantic representation of the layout graph (see Sec. 3.3) is obtained, by connecting the furniture nodes with the room node that they belong to using dash-dot edges.

The proposed RLM and RDM techniques, helped in rank ordering of the matched layouts based on the semantic difference i.e. differences in the arrangement and number of furniture inside the rooms. Figure 3.10, shows qualitative retrieval result corresponding to a query from the SESYD dataset. The red bounding boxes depict a change of decor between the query layout and the retrieved layouts, thus leading to dissimilar layouts being retrieved later than the similar ones

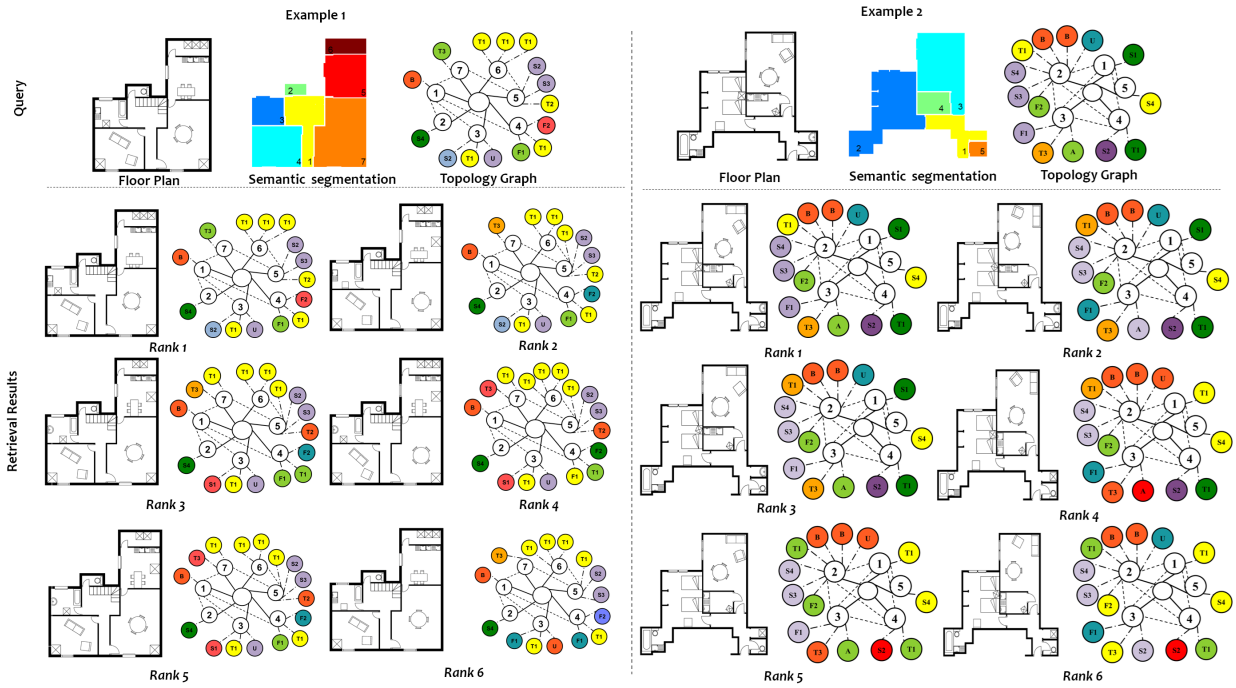


Figure 3.11. : Retrieval results for two different query architectural floor plans taken from the SESYD dataset. The top row shows the query image, the corresponding room layout and the semantic graph generated from the floor plan image. For the given query images only the top six retrieval results are shown.

during matching. As the result is demonstrated on SESYD dataset and it is observed that within a category in SESYD dataset only the decor arrangement differs, and the layout remains the same. Hence, the first rank ordered result is the query layout itself. The subsequent results differ in the placement, position, type and number of the furniture present inside the floor plan. For example, the Rank 5 result differs from the query floor plan in terms of a table's orientation as well as lacks a small table in comparison to the query. Hence, is retrieved later. To elaborate further, the rank ordered retrieval results along with their corresponding semantic graphs for two such queries, are shown in Fig. 3.11. The first row in Fig. 3.11 represents the query image, its room level segmentation, and the corresponding semantic graph. The following rows correspond to the top six rank ordered results along with their semantic graphs. It can be observed that the layout category in the rank ordered results is the same as that of the query and thus, ensures high retrieval accuracy. The first rank ordered result obtained from the layout database is the query image itself due to minimum distance from the query. The subsequent results differ in the type, number and orientation of furnitures inside each room.

After explanation of how the retrieval results differ in their layout/ decor arrangement, further, more qualitative results on both SESYD dataset and ROBIN dataset are shown in Fig. 3.12. In Fig. 3.12, the subfigures 1. (a) and (b) correspond to the query given from SESYD dataset and the subsequent rank-ordered results. While, 2. (a) and (b) correspond to the query given from ROBIN dataset. On closer observation it can be seen that retrieval framework performs well while layout matching and all layouts from the same category are retrieved as can be seen form the same global shapes of the layouts in the query as well as the retrieved results.

Figure 3.12. : Retrieval results for two different query architectural floor plans taken from the SESYD dataset and the ROBIN dataset. 1. (a) and (b) correspond to the query given from SESYD dataset and the subsequent rank-ordered results. 2. (a) and (b) correspond to the query given from ROBIN dataset

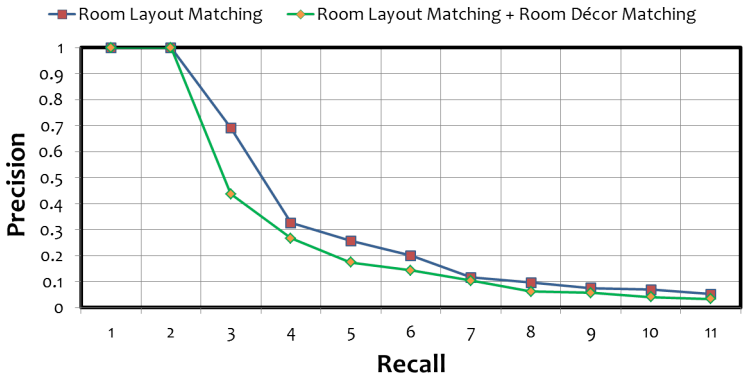


Figure 3.13. : Quantitative result comparing the RLM and the RDM approach on the ROBIN dataset

3.6.2 Quantitative Results

Quantitative analysis of the proposed algorithm was also done to observe the results. Precision (P) and Recall (R) are calculated using the equations mentioned below:

$$Precision(P) = \frac{|\{\text{relevant floor plan images}\} \cap \{\text{retrieved floor plan images}\}|}{|\{\text{retrieved floor plan images}\}|} \tag{3.12}$$

$$Recall(R) = \frac{|\{\text{relevant floor plan images}\} \cap \{\text{retrieved floor plan images}\}|}{|\{\text{relevant floor plan images}\}|} \tag{3.13}$$

Figure 3.13 shows the PR plot obtained by taking query from the ROBIN dataset and retrieving based on layout matching and a combination of layout and decor matching separately. In Fig. 3.13, the blue line marked with red squares denotes the PR plot obtained through layout matching (RLM) and the green line with yellow diamond markers depicts PR plot obtained with combination of both RLM and RDM. This study helped in observing how well the algorithm behaves in case of both matching strategies taken exclusively. The initial Recall value was obtained as 1 due to the first rank ordered retrieved result to be the same as the query. The subsequent results had some differences in the layout and decor. It was observed that taking the global layout, i.e. the placement of rooms inside a floor plan was a broader way of classifying the layouts, hence, the area under the PR plot with layout matching was obtained to be greater. This also owes to the fact that as more detailed decor characterization and matching is looked into, it leads to a slightly poor categorization of the layouts because of the specificity while retrieval.

Similarly, quantitative analysis of retrieval performance on the SESYD dataset was also performed. As discussed before, SESYD dataset has large differences between the layouts across its 10 categories but inside a category, all the layout shapes are the same, only differing in their decor arrangement. The values of the distance metric depicting the similarity between layouts from SESYD dataset are shown in Figure 3.14. In Fig. 3.14, the x-axis corresponds to the ten different categories of layouts present in the database, whereas, the y-axis corresponds to the distance value between two layout categories in terms of the distance between the feature vectors representing them. The nodes depict the distance of each category from the 10 categories in the database. For example, as shown in Fig. 3.14, yellow line with triangles corresponds to the distances with respect to layout category 3 in the SESYD database. Note that the distance is 0 with respect to category 3 and highest with respect to category 9. This is due to the very large difference in the layout characteristics between category 3 and 9. It is evident from the plot that our proposed RLM technique (see Sec. 3.5.3) yields high intra-category similarity and low inter-category similarity between layouts, as expected. To illustrate further, one can see in Fig. 3.15 for layout 8 and 4 that the first rank retrieved result is layout 8 itself due to exact matching between the query and the Rank 1 result. Further, Rank 2 result has similar number of rooms and adjacency information as the query. Other retrieved results follow suit. Due to high inter-class dissimilarity between floor plans in the SESYD dataset, a perfect Precision value was obtained for all the recall values while retrieval.

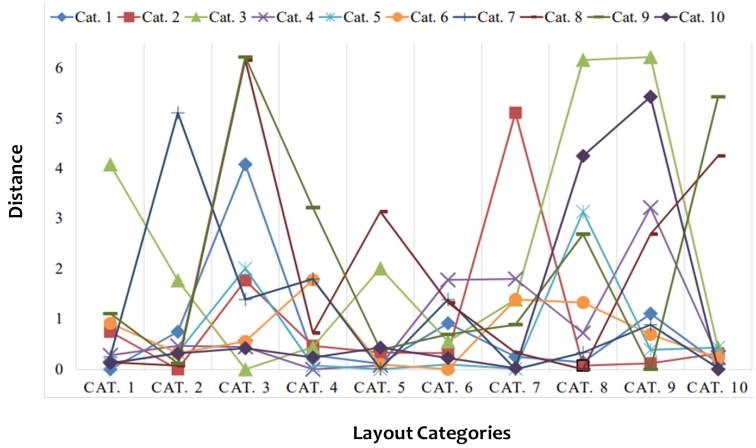


Figure 3.14. : Plot of intra-class and inter-class distances between layouts using our feature matching technique on the SESYD datset.

3.7 DISCUSSION

The framework discussed in this Chapter efficiently performs the various processing steps starting from, wall detection to room segmentation, graph creation of the layouts, feature computation, graph matching between the layouts, object level matching between the layouts and finally rank ordering the results. Another important aspect to be analyzed here is the search time. To compute the search time of the proposed approach time taken in seconds is computed at every step for the SESYD dataset. This time is averaged over 1000 samples belonging to a single class in the SESYD dataset. Figure 3.16 shows the class-wise computation time-ratio for the layouts in the dataset. Colour coding signifies various components of the proposed framework and helps establishing correspondence to understand which step performs computationally efficient in the proposed framework.

It can be observed that the wall detection step took an average time of 4.481 secs. for all the layout categories. Segmentation into rooms was a little computationally expensive step owing to a number of intermediate processing steps. The average time taken by the segmentation step was 18.047 secs. Graph generation and spectral feature embedding of the graphs took an average time of 5.513 secs. and 5.292 secs. respectively. Figure 3.16 shows this analysis, where, values shown in the bar chart denote the ratio of time taken in seconds for each step to the total time taken for execution.

As discussed in Sec. 3.3, graph matching is a computationally expensive process. To avoid this spectral graph embedding is proposed in this Chapter, which creates uniform three component vectors for representing all the graphs, which helps in clustering similar graphs closer in the pattern space. Therefore, this technique considerably reduced the computation time while matching the layouts. The average time taken at this step is 7.569 secs.

Analysis revealed that the furniture level matching inside the layouts was the most computationally expensive step in the proposed approach. This step took the maximum average time of 154.036 secs for every layout because of 1) iteratively searching for the arrangement of furniture inside each layout, and 2) large dimension (6775×2858) of the layout images. An efficient indexing method or hashing technique for the second level match could yield better results.

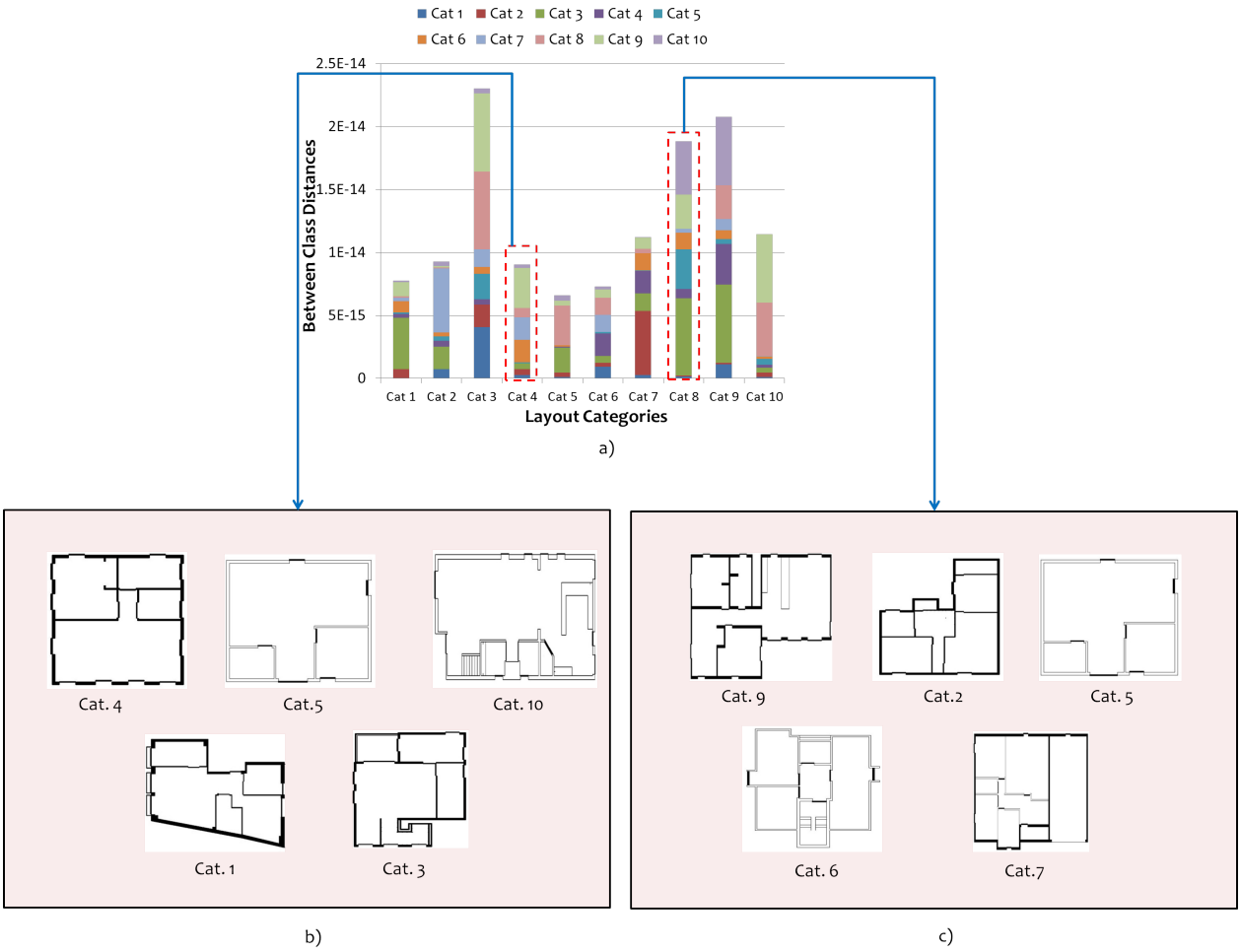


Figure 3.15. : Layout matching from floor plans taken from SESYD dataset. Two such results from Category 8 and 4 are highlighted with a red bounding box.

Similar analysis was also performed for the ROBIN dataset. As the classes in ROBIN dataset are 51 in number, therefore, it is not feasible to show class-wise search time graph for the ROBIN dataset. Alternatively, in Fig. 3.17 the search time taken in seconds averaged over all the 510 samples in the ROBIN dataset, at every step of the framework is represented as a bar-graph. It was observed that the most computationally step was the decor matching step again taking an average time of about 7.34 secs., which results from the fact that, iteratively searching for the arrangement of furniture inside each layout is computationally expensive. On the other hand, the most computationally efficient step is the Automatic Room Detection step.

As this was a novel baseline framework proposed in this Chapter, therefore, due to a lack of availability of the implementation of such a composite framework for retrieval and matching of floor plans in the literature, there is an inability to compare the results with other approaches. Although, in Fig. 3.18 the comparison of segmentation step to the one proposed in [de las Heras *et al.*, 2014] is shown. Where, segmentation results on two images from CVC-FP [de las Heras *et al.*, 2015] dataset are shown, with, the left hand side being (a) the input floor plan and (b) and (c) correspond to the technique proposed in [de las Heras *et al.*, 2014] and the proposed approach respectively. It can be observed that the proposed segmentation technique performs at par with the technique proposed in [de las Heras *et al.*, 2014].

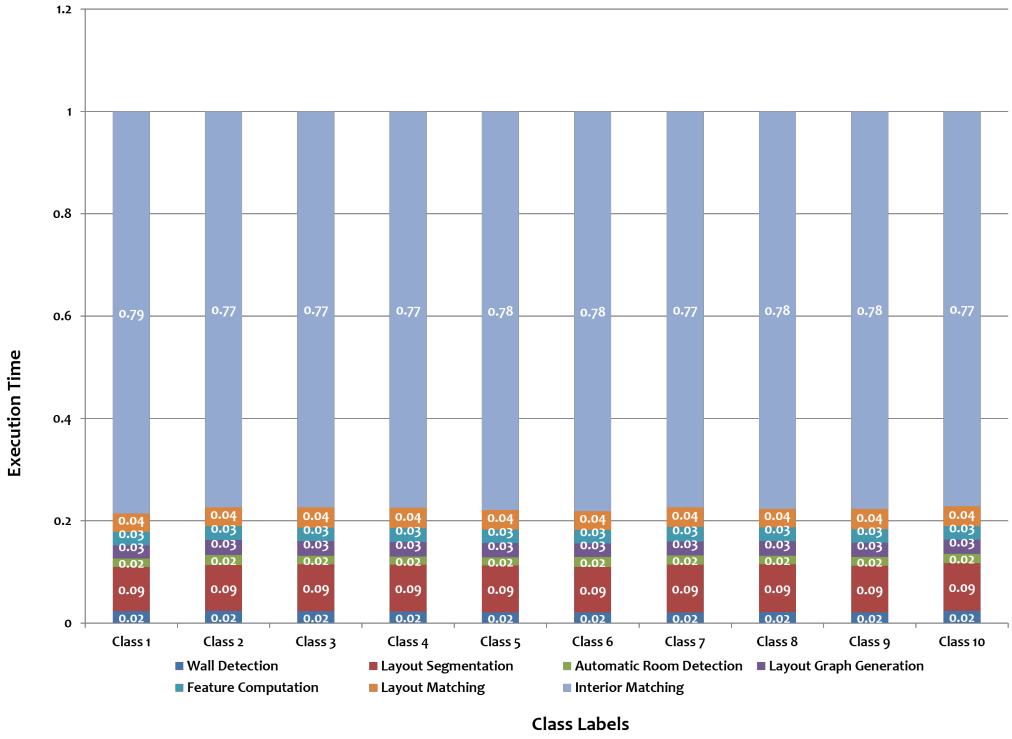


Figure 3.16. : Execution time for the intermediate stages of processing of the framework for different categories of floor plans on the SESYD dataset. Values shown in the bar chart denote the ratio of time taken in seconds for each step to the total time taken for execution.

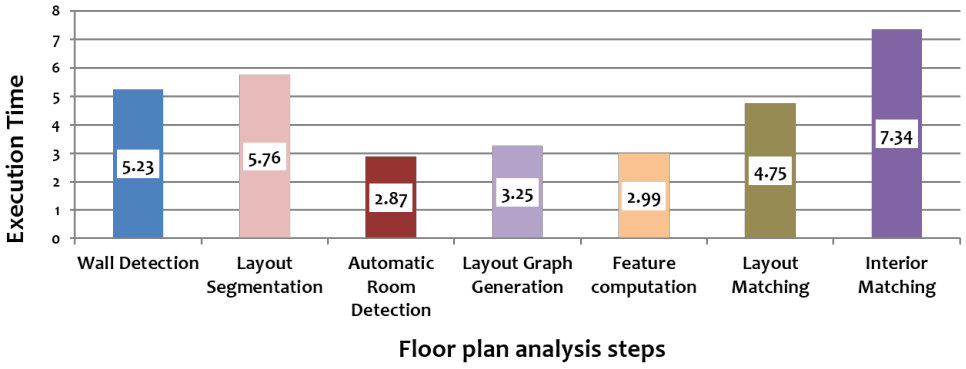


Figure 3.17. : Execution time for the intermediate stages of processing of the framework averaged over all 51 categories of floor plans in the ROBIN dataset. Values shown in the bar chart denote the time taken in seconds.

The approach proposed in [Weber *et al.*, 2010] taking sketch as a query to retrieve floor plans is partly related to the work proposed in this Chapter. However, the key difference is that the authors in [Weber *et al.*, 2010] represent query layouts as sketches using only primitive shapes

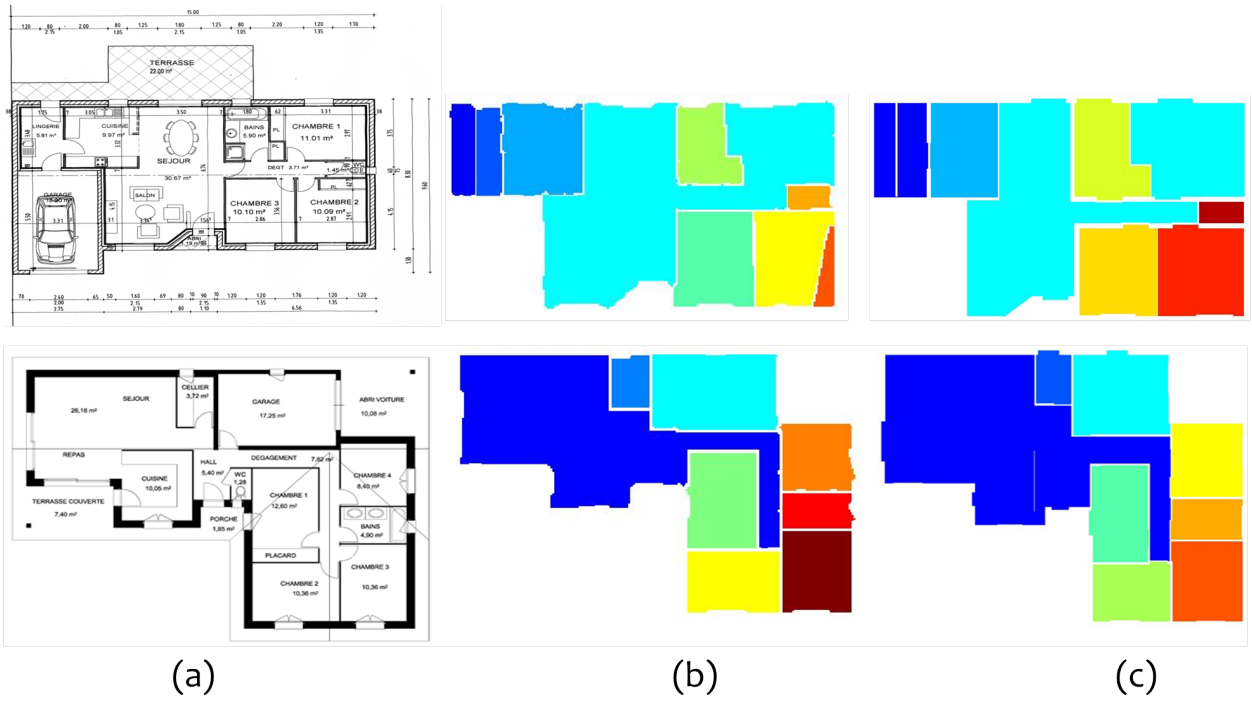


Figure 3.18. : Segmentation results on two images from CVC-FP [de las Heras *et al.*, 2015] dataset (a) Input floor plan, (b) Technique proposed in [de las Heras *et al.*, 2014], (c) Our approach

such as rectangles. In contrast, in this Chapter the room decor represented by complex geometric primitives is considered. Moreover, in [Weber *et al.*, 2010] the adjacencies between the rooms represented in the layouts, in sketch form were very simplistic in nature which is quite complex in our case. Furthermore, they have ignored the representation and matching of the room decor between the layouts.

3.8 SUMMARY

In this Chapter, a framework for searching floor plans by structural analysis, as well as, semantic similarity is proposed. Inclusion of the room decor arrangement during retrieval is a novel idea that can find application in scenarios where the user has specific choices in terms of the fully furnished floor plans. Furthermore, a spectral embedding approach to represent graphs obtained from the layout is proposed in the baseline retrieval framework proposed in this Chapter. This technique helped in reducing the computation time taken during the graph matching stage to a considerable level.

In this Chapter, handcrafted topological features were proposed to be extracted from the floor plans. Topology and room decor features although help in distinguishing two floor plans and retrieving similar ones, but inclusively are not a good enough measure to analyse a floor plan completely. It is very difficult to curate handcrafted features powerful enough to extract all the details from a floor plan. On the other hand, deep neural networks are able to analyse a floor plan in a minute manner extracting all the relevant features at once. In the literature, deep neural network based features have shown promising results for the retrieval of natural images. However their application to floor plan images has not been researched. Therefore, in the next chapter, deep

neural networks are explored to extract structural and semantic information from a floor plan for the retrieval task.