2
Background and Related Work

2.1 INTRODUCTION

In this chapter, the existing literature regarding the research areas discussed in this thesis is presented, followed by a brief review of the existing datasets in the area of floor plan analysis. Document Image Analysis (DIA) tasks involving Graphics Recognition specifically have evolved mainly due to the growing trend of not only storing digital documents these days, but also due to extraction, classification, and indexing of them according to the information conveyed by the documents. This gained popularity after the emergence of scanners that increased the managing and handling of digitized documents. Application of Graphics Recognition to domains like engineering drawings, electrical diagrams, flowcharts, maps, mathematical and chemical formulae, and architectural plans has been done from the very start of the evolution of this field. After one recognizes the graphics in a particular document the next task is to derive something meaningful out of it. As this thesis pertains with engineering drawings, specifically floor plans therefore, in the next section the existing work related to symbol spotting, floor plan analysis and retrieval in floor plans, the commonly used datasets in floor plan analysis and generic feature representation and deep learning retrieval methods is discussed.

2.2 STATE OF THE ART IN FLOOR PLAN ANALYSIS

Floor plan analysis is one of the most trendy applications in graphical document understanding due to the actual demand of real world applications. Thus, several systems dealing with heterogeneous architectural inputs such as sketch drawings, machine generated documents, and CAD files have been presented in the recent years. Nevertheless, floor plan interpretation problem has not yet been solved for two main reasons: the inherent difficulty of the problem i.e. the visual language may differ from plan to plan, and the lack of reusability and improvement of the existing approaches. In the following paragraphs, the most recent techniques for floor plan analysis are described in detail. In Tab. 2.1, some key approaches are listed with their salient features.

2.2.1 3D reconstruction of floor plans

The work proposed in [Ah-Soon and Tombre, 1997], [Dosch and Masini, 1999], and [Dosch *et al.*, 2000] deals with floor plan 3D reconstruction. In these works, the input is scanned floor plans. First, a preprocessing step is followed to separate both text and graphics information. Then the thick and thin lines present in the graphical layer are segregated and vectorized. Thick lines depict walls whereas, the thin ones depict the rest of the symbols, including windows and doors. In this process, two variations of walls are considered: first being the walls represented by thick parallel lines and the second case being walls represented by a single thick line. Arcs in the symbols represent doors, windows are detected using small closed loops, and rooms are generally

Table 2.1. : Reviews of approaches for floor plan analysis

Paper Title	Technique	References
A system to understand hand-drawn floor plans using subgraph isomorphism and Hough transform	Attributed graph structure, Straight line Hough transform for speeding matching	Lladós *et al.* [1997]
A Prototype System for Interpreting Hand-Sketched Floor Plans	Automatically converts hand-sketched floor-plans into the CAD formation, Template matching of closed regions	Aoki *et al.* [1996]
A complete system for the analysis of architectural drawings	Converts the drawing into a description in terms of basic architectural entities, Reconstructing in 3D	Dosch *et al.* [2000]
Robust and Accurate Vectorization of Line Drawings	Vectorizing the graphical parts of line drawings, Compute feasibility domains for accurate vectorization	Hilaire and Tombre [2006]
The Room Connectivity Graph: Shape Retrieval in the Architectural Domain	Characterize 3D architectural models on underlying arrangement of their rooms, Retrieval of 3D models.	Wessel *et al.* [2008]
A System to Detect Rooms in Architectural Floor Plan Images	Hough transform, Decomposing rooms to near convex regions	Macé *et al.* [2010]

a composition of bigger connected components. Finally, a single level in a floor plan or several floors in the same building are identified and their 3D reconstruction is performed [Ah-Soon and Tombre, 1997] by finding special symbols as beams, staircases, etc. [Dosch and Masini, 1999]. Both in [Dosch and Masini, 1999] and [Dosch *et al.*, 2000], emphasis is made on the necessity of human feedback while working with complex plans. Or *et al.* [2005] focus on 3D model generation from a 2D plan. Rendek *et al.* [2004] preprocess the image by firstly segregating the text and graphics and then converting the graphical layer to a vector form. They further, manually delete other noisy lines and symbols to finally detect the plan structure. Once the remaining structure consists of only lines belonging to walls, doors, and windows, a set of polygons using each polyline of the vectorized image is generated. At the end, blocks are represented by polygons; thick lines represent outer walls, rectangles represent windows inside the walls, and arcs represent doors. This system is able to generate the 3D model corresponding to single story buildings. Again, this approach is plan specific and slight change in the floor plan leads to modification of the technique. Some works such as proposed by Lu *et al.* [2007] for 3D reconstruction contain a CAD file as input which consists of lines of the architectural design and real non-distorted original polylines. First, T, L and X shapes are identified by extracting parallel overlapped lines. Then, the 3D reconstruction of the plan is retrieved by identifying connections between the lines. After that, segmentation of the graphical symbols such as furniture and staircases is done by deleting the structural lines. This method is primarily based on symbol recognition using geometrical features extracted from the image.

2.2.2 Semantic analysis of floor plans

Cherneff *et al.* [1992] present a knowledge-based interpretation method for architectural drawings. Their aim is to extract the complete structure of the plan, which comprises of the recognition of walls, doors, windows, rooms, and the relations between them. The input is a floor plan vectorized in order to obtain unique symbols for doors. Then, the system is composed of two models: the semantic model and the structural model. The semantic model represents the contextual composition of the architectural elements. The structural model entails the geometrical configuration of the plan, including the 2D spatial indexing of primitives. Walls are defined as parallel segments incidenting to doors or windows at their ending points. This fact places a constraint on the possibilities of interpretation, since it is based on the assumption that walls are always straight, however in real world scenario walls might be curved or even not be modeled using parallel lines. The work presented by Cherneff *et al.* [1992] focuses on room segmentation. A proximity map based semi-automatic approach is used for searching for regions in machine printed floor plan images. This method serves as an extension of the area-filling approach which mainly deals with splitting rooms when there is a lack of physical separation. However, the drawback here is that this method retrieves many false positives in the form of all those objects that are modeled as closed regions, for example, doors, tables, and staircases. [Macé *et al.*, 2010] focus on the extraction of the building structure from scanned images. Similarly to [Dosch *et al.*, 2000], [Dosch and Masini, 1999] they firstly separate textual from graphical information. Then, on the graphical layer, they recognize walls by detecting parallel lines at the contours of the thicker lines. Thick lines are earlier separated from the thin ones morphologically. Finally, rooms are recognized by a recursive decomposition of the images into convex regions. In this method, the wall detector strongly depends on the wall notation, which encompasses a redefinition in order to be able to deal with a variety of floor plan models. Work proposed by Ahmed *et al.* [2012] also starts with the classical text/graphics separation. After that, they morphologically separate the graphical components into thin, medium, and thick lines. Medium and thick lines are considered to be walls whereas small lines model the rest of the architectural elements. Then, doors and windows are detected by means of a patch-based description of the image using SURF [Bay *et al.*, 2006]. Rooms are recognized by finding closed areas enclosed by walls, doors, and windows. At the end, the textual labelling inside the room areas are used to verify and correct the final segmentation. This whole method focuses on a single graphical notation, specially in the process of wall extraction. Also, the work of Zhi *et al.* [2003] takes input in the form of a CAD file. It then automatically extracts the topological and geometrical information from a 2D architectural drawing and builds a simulator that helps in building evacuation in case of crisis situations. Firstly, filtering of redundant information such as dimensions, text, and furniture, is carried out in a semi-automatic fashion and only the remaining necessary objects like doors, windows and walls are kept. Then, the plan is transformed into an attributed graph and loops are searched for in the floor plan. Further, according to their respective attributes, loops are classified into different categories such as rooms and corridors are represented by spatial loops, walls and columns through physical loops, window loops, door loops and unrecognized loops. Finally, the compartments in the floor plans are identified and the system is integrated in a simulation model that aids in difficult evacuation strategies by connecting rooms to their nearest exits/staircases in complex buildings.

2.2.3 Structural Analysis

Works as proposed by Aoki *et al.* [1996] and Lladós *et al.* [1997] analyze hand-sketched floor plans. On the one hand, [Aoki *et al.*, 1996] transforms the sketches into CAD files. In this particular work the lines that model the building structure are extracted, which are basically handwritten on a preprinted paper with a grid of lines. This method again describes line elements in the form of windows and walls, and closed region elements here represent doors. On the other hand, [Lladós *et al.*, 1997] recognizes building elements and their respective topological properties using Hough transform and subgraph isomorphism. Subgraph isomorphism aids in symbol recognition whereas Hough transform helps in detecting walls modeled using hatched patterns. It should be noted that in both, [Aoki *et al.*, 1996] and [Lladós *et al.*, 1997]], the conventions to be used while drawing are prescribed beforehand. Floor plan structural retrieval has gathered the attention of architects recently. The works of [Ahmed *et al.*, 2014], [Weber *et al.*, 2010] and [Wessel *et al.*, 2008] are three examples of this application. In the case of the work proposed by Weber *et al.* [2010], the query is specified in the form of a sketch drawn online by a user. Their system allows the user to sketch an abstract schematic representation of a floor plan and uses this representation to search for structurally similar documents. The sketch is modelled as a graph, which encloses the structure of the plan, and it is compared to the repository representations using subgraph matching algorithms. As proposed by Wessel *et al.* [2008], the specified input is in the form of a polygon representing a 3D floor plan, so they do not need to vectorize the plan. They construct a graph where attributed nodes are rooms and attributed edges are connections between them: doors or windows. Based on this connectivity graph, fast and efficient shape retrieval is achieved.

Table 2.2. : Reviews of approaches for symbol spotting

Paper Title	Technique	References
Statistical Grouping For Segmenting Symbols Parts From Line Drawings, With Application To Symbol Spotting	Groups line drawings into shapes, Identifying ROI and aiding matching process	Nayef and Breuel [2011]
Symbol Spotting in Line Drawings Through Graph Paths Hashing	Hashing the shape descriptors of graph paths, Local sensitivity hashing, Spatial voting scheme	Dutta *et al.* [2011]
Combination of Product Graph and Random Walk Kernel for Symbol Spotting in Graphical Documents	Path similarity using product graph, Random Walk Kernel for calculating distance measures	Dutta *et al.* [2012]
An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings	Integer linear formulation, Substitution-tolerant subgraph isomorphism.	Le Bodic *et al.* [2012]
A symbol spotting approach in graphical documents by hashing serialized graphs	Hashing the shape descriptors of graph paths, Local sensitivity hashing, Spatial voting scheme	Dutta *et al.* [2013]
Object Recognition in Floor Plans by Graphs of White Connected Components	Region Adjacency Graph, White connected components as nodes	Barducci and Marinai [2012]

2.3 STATE OF THE ART IN SYMBOL SPOTTING

Symbol spotting in floor plans deals with given an image of a query symbol, location of the query symbol and associated documents are retrieved. There are two approaches to symbol spotting. The first pertains to a region-based approach, which holds its basis on searching for regions of interest inside drawings, using various descriptors to describe them, and finally using a query symbol to index them. The second approach for spotting deals with isolating the symbols using segmentation to get the symbols segregated from their background, and then performing their recognition. Generally the spotting methods mentioned in the literature can be categorized into the region-based approach. However, this approach has the following inherent problems: (1) It is difficult to locate regions of interest in a scaled down drawing; (2) Rotation invariance in such type of drawings is hard to achieve; (3) As the patterns in line segments are simplistic in nature, special descriptors need to be developed to represent line drawings, instead of using the regular textual based descriptors and, (4) Indexing techniques like hashing fail to scale well when one brings large databases into the picture. Due to these problems, the methods implementing the region based approach result in low recall and precision rates. Inspite of the drawbacks, a primary reason to follow this approach, rather than the segmentation-based approach, is the incapability of the current segmentation techniques to peform while dealing with complicated technical drawings. Table 2.2 lists some important works related to symbol spotting.

2.3.1 Region based symbol spotting

In this spotting approach, various methods for detection and description of regions of interest within drawings are employed by the state-of-the-art methods. Authors in [Rusinol and Lladós, 2007] and [Rusiñol *et al.*, 2010], propose the use of different spotting methods working with detecting closed regions to further identify the regions of interest. In [Rusinol and Lladós, 2007], a bipolar coordinate system is used to depict a region of interest and for indexing purposes a hash table is used. In [Rusiñol *et al.*, 2010], attributed strings are used to represent regions of interest. A polygonal approximation of the contour of the closed regions is computed and association of adjacent segments is done in the form of polylines. These polylines are encoded as attributed strings, where attributes consist of, the length of the segment and the angle between the current and the previous segment. Similar or alike strings are then clustered together in a look-up table and a set of median strings represent the indexing keys to index into the table. String matching is used for comparison of the query to regions in the image. Finally, a Hough voting scheme is used for the verification of the initially found matches. In another work as proposed in [Ah-Soon and Tombre, 2001], overlapping sliding windows are used to detect regions. The regions here are described as feature vectors depicting the relations between the connections of the region segments, and graphs are used to describe the descriptions. The symbol search is then ensured by calculating the similarity of the relationship between the segments in the network.

2.3.2 Segmentation and recognition of isolated symbols for symbol spotting

Graph representations carry their fair share of popularity in isolated symbol recognition methods for symbol spotting. In [Qureshi *et al.*, 2007], authors propose the use of attributed relational graphs (ARG) for the representation of line primitives in the form of vectors, to represent floor plan drawings. The relationship between the identified vectorial primitives paves way for the detection of regions inside the floor plans. Matching of the query to the regions is done using a graph description and matching method proposed by Qureshi *et al.* [2006]. A combination of graph parsing and matching for representing line drawings was proposed by Llados and Sanchez

[2004]. In this paper, the authors present a mechanism for graph indexing to spot known symbols in the drawings. Locteau *et al.* [2007] again used a graph based approach for representation of geometrical shapes which collectively represent geometrical primitives in line drawings. Detection of the regions is done based on open or closed curves. Subsequent query matching to the regions is performed using edit distance metric. Tabbone *et al.* [2003] presented a spotting method that uses the segmentation-based spotting approach. In this particular approach authors propose the need of a segmentation method to segregate text and graphics along with user feedback to further perform symbol segmentation in simplistic line drawings. The descriptor used here is a discrete version of the F-signature which is the histogram of forces. For matching the query to the segmented symbols, the sum of absolute differences is used.

It can be safely stated for the spotting methods following the region-based approach, that if one is able to accurately and precisely identify the regions in the line drawings, then the subsequent steps of matching and description can achieve good results. On the other hand, methods employing isolated symbol spotting approach, rely on efficient prior segmentation, which makes this approach a bit challenging. It is observed that there is an increase in the number of spotting methods which project good resultant recall rates, but usually have a drawback, that they give low precision rates.

The online process for symbol spotting is faster and efficient and usually deals with the following steps: (1) locating regions of interest, (2) clustering (3) creating an efficient indexing data structure from the obtained clusters. During the time of retrieval, the query description is compared with the entries already stored apriori in the data structure. Approaches mentioned in [Nguyen *et al.*, 2009], [Luqman *et al.*, 2010], [Kong *et al.*, 2011] and [Rusiñol *et al.*, 2010] follow this particular technique for symbol spotting. To summarize, the issue with using these methods is that they are lacking in projection of efficient steps for content analysis, which further results in a not-so-good representation of the floor plan drawings, and thus, inadvertently affects the recall and precision rates of retrieval, reducing them to low values.

2.4 STATE OF THE ART IN GENERIC FEATURES FOR RETRIEVAL

So far the existing techniques involved in floor plan analysis and retrieval are discussed. In the latter chapters of the thesis, comparisons with generic retrieval based approaches are done. This section presents a review of features used for retrieval tasks. Several book chapters and papers in the literature present a thorough review of the advances in the feature descriptors [Doermann *et al.*, 2014; Zhang and Lu, 2004]. The primary goals of these description techniques can be listed down as: (1) minimizing the intra-class distances and maximizing the inter-class distances, (2) minimizing the spatial dimensionality to enhance the time complexity and the classification efficiency, and (3) dealing with noise, transformations and distortions in the images. Some popular techniques involve adapting existing feature descriptors from popular research fields like pattern recognition, e.g SIFT [Lowe, 2004], SURF [Bay *et al.*, 2006], and HOG [Dalal and Triggs, 2005]. In HOG [Dalal and Triggs, 2005], the idea is that the shape and appearance of an object within an image can be described by a histogram of intensity gradients. This descriptor is built by dividing the image into small regions, and computing a histogram of the edge orientations for each region. The combination of these histograms form the descriptor for the image. This method has been used for human detection in images. SIFT [Lowe, 2004] has been one of the most widely used feature descriptor in object recognition. Lowe [2004] first proposed this method of representing an image as a collection of local feature vectors which are invariant to translation, scaling, and rotation. These features are initially detected using a Harris-Laplace corner detector in scale space. Once corners are detected, the scale space at which they were detected are stored. Then for each of these keypoints, the dominant gradient is computed over a neighborhood of 16×16. This is also stored and after which, the descriptor for this keypoint is computed. This keypoint descriptor is described by a 128 dimensional vector. This vector is obtained by selecting a 16×16 neighborhood around the keypoint. This is then partitioned into 4×4 regions with each region containing 4×4 pixels. For each region, a histogram with 8 bins is computed from gradient magnitudes in that region. Thus, the 128-dimensional vector is formed. SIFT has also many variants. The first of which is SURF or Speed Up Robust Features [Bay *et al.*, 2006]. The key difference is that this uses integral images to speed up convolutions. Other variants include RGB-SIFT, HSV-SIFT, and Opponent-SIFT. These mainly use SIFT on different color spaces since SIFT was primarily made for grayscale images. RGB-SIFT operates on the RGB channels, HSV-SIFT on HSV channels, and C-SIFT on opponent channels. One final variant is called dense SIFT (or D-SIFT) where descriptors are computed for every pixel. In certain cases, describing the symbol characteristics requires strategies which are highly domain specific, e.g. zoning [Escalera *et al.*, 2009], geometric moments [El Rube *et al.*, 2006], and histogram-oriented [Yang, 2005] techniques. Finally, classification techniques on extracting the features are highly benefited by mathematical techniques such as the computation of sums, products and distances. Thus, several effective algorithms for classification tasks, such as K-nearest neighbour, SVM, boosting and neural network learning have been widely employed in the literature. Another technique used for extracting features is Run Length Histograms (RLH) [de las Heras *et al.*, 2013] approach where, the style of an architectural drawing is characterized by the perception of lines, shapes and texture. Authors propose runlength histograms extracted in vertical, horizontal and diagonal directions as a characterization of line and space properties in floor plans that further estimate a description of walls and room structure. Another efficient technique is an Online Algorithm for Scalable Image Similarity learning (OASIS) [Chechik *et al.*, 2010], which learns a bilinear similarity measure over sparse representations. OASIS is an online dual approach using the passive-aggressive family of learning algorithms that employ an efficient hinge loss cost.

2.5 STATE OF THE ART IN DEEP LEARNING FRAMEWORKS FOR IMAGE AND SKETCH RETRIEVAL

With the extensive research on deep neural networks (DNN) [Krizhevsky and Hinton, 2011; Szegedy *et al.*, 2015; Simonyan and Zisserman, 2014], the success and efficiency of the learning-based features have been observed in multiple areas. High-level abstractions providing an approximation to the human cognition system can be learned [Bengio *et al.*, 2009] using deep architectures. As a result, DNN can be used to extract semantically oriented features using the activation in different layers in the networks. In the work proposed by Hörster and Lienhart [2008], feature extraction is carried out using local patches employing a deep restricted Boltzmann machine. Convolutional neural networks (CNN) form a sub-component of the DNNs. Authors in [Krizhevsky *et al.*, 2012b] have demonstrated a highly superior performance in various tasks on image recognition and retrieval [Razavian *et al.*, 2014]. In the paper proposed by Wan *et al.* [2014], detailed studies are conducted on the capability of learned visual features using deep CNN for a variety of applications which include content based image retrieval as well. Razavian *et al.* [2014] study the VGG-Net and Alex-Net and further try to tap the response of the last convolutional layers along with max pooling for image representation for the final task of image retrieval [Razavian *et al.*, 2016]. In the work of Zheng *et al.* [2015], the sixth layer of the Alex-Net [Krizhevsky *et al.*, 2012b] is activated and the activations are taken out as a DNN feature for each image, which is integrated in the image similarity match score level along side traditional visual features that include HSV histogram, SIFT-based BoW feature, and GIST. The state-of-the-art sketch based deep retrieval models [Sangkloy *et al.*, 2016] are deep models that aim to close the domain gap by learning a joint feature embedding for the two domains. Multibranch deep convolutional neural networks (CNNs) are employed where each branch corresponds to one domain and the final shared layer defines the embedding space which is subject to various discriminative losses such as pairwise contrastive loss or triplet ranking loss. These losses are designed to pull matching pairs of photos and sketches close and push mismatched pairs away. These models thus indirectly align the two domains. However, with limited training data and by focusing only on discriminative losses, these models struggle to capture all the domain-invariant information and thus generalise poorly to test data where the domain discrepancies and misalignments could be different from those in the training data. Alternatively, Generative Adversarial Networks (GANs) have also been employed for generic sketch retrieval [Creswell and Bharath, 2016] and [Chen and Hays, 2018] but have not yet been explored for application in floor plan retrieval. GANs combine the discriminative model with a generative component. One of the typical cases is to use source images, noise vectors or both to generate simulated samples that are similar to the target samples and preserve the annotation information of the source domain. One more approach that is used is the reconstruction-based domain adaptation approach, which assumes that the data reconstruction of the source or target samples can be helpful for improving the performance of domain adaptation. The reconstructor can ensure both specificity of intra-domain representations and indistinguishability of inter-domain representations by using stacked autoencoders (SAEs), that combine the encoder network for representation learning with a decoder network for data reconstruction [Ghifary *et al.*, 2016],[Zhuang *et al.*, 2015].

2.6 FLOOR PLAN DATASETS

Table 2.3. : Details of publicly available existing floor plan datasets

Floor plan Datasets		
Dataset	Count	Remarks
CVC-FP [de las Heras *et al.*, 2015]	122	4 Sub-Categories, varying in wall textures, to study graphical notations in floor plans
FPLAN-POLY [Rusiñol *et al.*, 2010]	42	Used for floor plan analysis and room analysis
SESYD [Delalandre *et al.*, 2007]	1000	100 layouts/ class, differ in arrangement of symbols, used for symbol spotting tasks

Performance evaluation of symbol spotting tasks and layout analysis on floor plans requires datasets that are publicly available to aid the research in this area. Table 2.3 contains the details of some publicly available floor plan datasets. One of the widely popular databases for symbol recognition and spotting tasks is the Systems Evaluation SYnthetic Documents (SESYD) database [Delalandre *et al.*, 2007]. It is a collection of labelled synthetic images. It consists of electrical and architectural drawings, designed specifically for symbol spotting, recognition and retrieval. The whole dataset consists of 5 document collections containing $11,100$ images with $128,700$ symbols. It contains 1000 floor plans divided into 10 broad categories, each containing 100 samples. Inter-class similarity is very low as layouts across each category have a huge difference. Intra-class similarity is very high as floor plans within a class only differ in the furniture/symbol arrangement and orientation in each plan.

Another floor plan dataset FPLAN-POLY proposed in [Rusiñol *et al.*, 2010] contains 42 floor plan vectorial images in dxf format which can be viewed with the common CAD softwares. The main goal of this database also is to provide a framework for the evaluation of different symbol spotting methods in vectorized graphic documents. CVC-FP dataset proposed in [de las Heras *et al.*, 2015] is annotated for the architectural objects and their structural relations. It contains 122 floor plans spread over 4 different layout categories. This dataset aims at aiding in learning and better interpretation about how the elements are structurally arranged in a floor plan.

2.7 SUMMARY

In this Chapter, various research areas such as floor plan analysis, symbol spotting, retrieval, feature extraction and deep learning models for analysis are discussed. Although the literature in the above mentioned areas is quite rich, there does not exist a composite framework in the area of floor plan analysis that takes meaningful information extracted out of floor plans to perform content based retrieval. Moreover, the publicly available datasets in the area of floor plans are mainly suited for symbol detection and structural analysis in floor plans. Neither are the existing datasets diverse enough to offer varied content for retrieval task nor are the samples in them adequate in number to implement deep learning approaches for efficient analysis and retrieval. Thus, in the consequent chapters of this thesis techniques for content based retrieval in floor plans are proposed to aid users/architects to help find their dream property according to their specific requirements.